# *EXC-1553UNET/Px &*
# *ES-1553RUNET/Px*

## Avionics Communication Device


# User's Manual

Avionics Communication Device

EXCALIBUR SYSTEMS

One Step Ahead

311 Meacham Ave ♦ Elmont ♦ NY 11003 ♦ tel. (516) 327-0000 ♦ fax (516) 327-4645
e-mail: excalibur@mil-1553.com website: www.mil-1553.com

# Table of Contents

# Figures

## Tables

# 1 Introduction

Chapter 1 provides an overview of the *UNET*.



*Caution:* On the *EXC-1553UNET/Px*, make sure there is no I/O communication while disconnecting any of the cables. Connecting or disconnecting the cables during communication can seriously damage the *EXC-1553UNET/Px*.

*Caution:* On the *ES-1553RUNET/Px*, make sure the power is OFF while connecting or disconnecting communication cables to or from the device. Connecting or disconnecting cables to or from the *ES-1553RUNET/Px* while the power is ON can seriously damage the *ES-1553RUNET/Px* or a system connected to the *ES-1553RUNET/Px*.

**Note:** In this manual, when the name *UNET* is used, it refers to both the *EXC-1553UNET/Px* and the *ES-1553RUNET/Px*.

## 1.1 Overview

The *EXC-1553UNET/Px* is an intelligent, single- or dual-channel, MIL-STD-1553 interface device. Its small size and ability to interface through USB or Ethernet interfaces make it a complete solution for developing, testing and performing system simulation of the MIL-STD-1553 bus, both in the lab and in the field. The *EXC-1553UNET/PxS* is the single function version of the device with a single RT and no error injection.

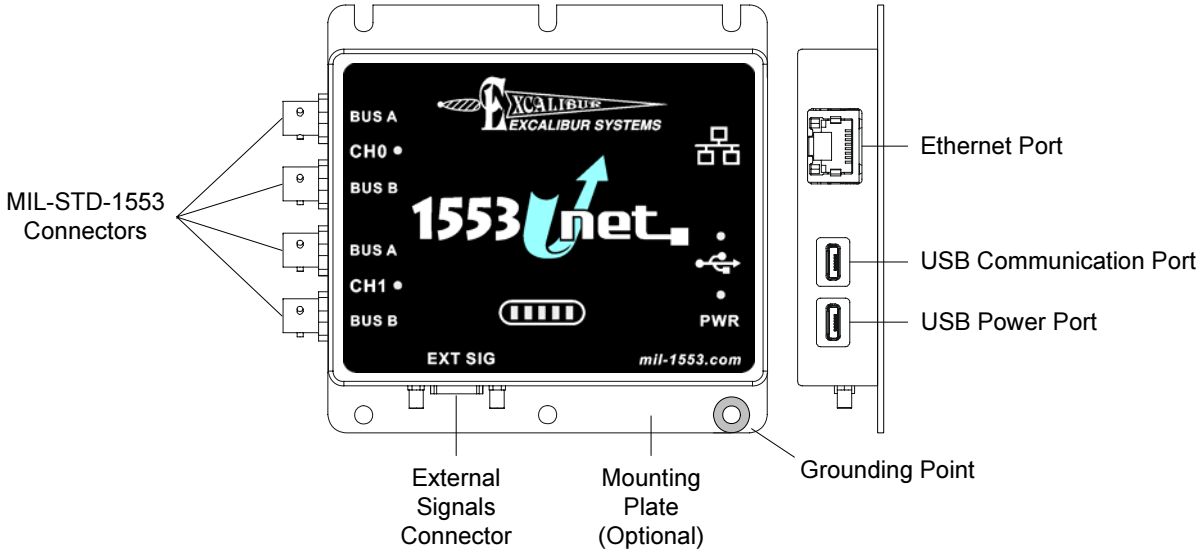



**Figure 1-1** ***EXC-1553UNET/Px***

**Note:**

- For an explanation of the LED indicators, see **4.3 LED Indicators of EXC-1553UNET/Px** on page 4-5.
- For ordering information see **Chapter 12: Ordering Information.**

In addition, this device provides IRIG B input and 8 Discrete I/O signals. The Discrete channel can record changes in the input Discrete with an associated Time Tag via a built-in FIFO. Output Discretes are open collector, capable of handling up to 32V with a maximum sink current of 100 mA each.

You can set for each Discrete:

- Whether it is input or output
- Input voltage level: TTL (0 – 5V) or Avionics (0 – 32V)
- Whether to enable or disable debounce on inputs

Multiple *UNET*s can operate via USB ports on the same computer. In addition, multiple units can operate on the same network, by programming each one with a unique IP address, and can be accessed from any computer on the network.

The *UNET* shares its API with the entire Px family so that applications currently running on our PCIe, PCI, ExpressCard, or PCMCIA cards, will run without change on this device.

### 1.1.1 Battery Option

The *EXC-1553UNET/Px* is available with an internal rechargeable battery. This allows the user to operate the *EXC-1553UNET/Px* without the need for any external power source. The internal battery can also be used as an auxiliary power source that can provide the additional power required for high data transmission conditions where external power sources are limited, for example, when the external power is provided by a host laptop computer. In this case, the internal battery can be used to minimize the amount of power drawn from the laptop's battery, which could be a critical requirement in the field where an external power supply is not readily available. For more information, see **4.7.1 Battery Option Power Information** on page 4-17.

### 1.1.2 Ruggedized Version

The *ES-1553RUNET/Px* is a ruggedized single-channel, MIL-STD-1553 interface device with an Ethernet interface. It complies with MIL-STD-810G, MIL-STD-461E, MIL-STD-704E and MIL-STD-1275B. The *ES-1553RUNET/PxS* is the single function version of the device with a single RT and no error injection.

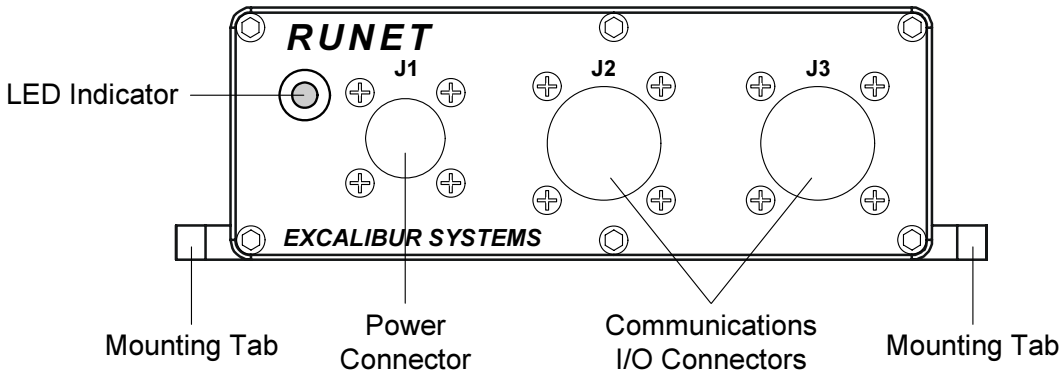



**Figure 1-2** ***ES-1553RUNET/Px***

**Note:**

- For an explanation of the LED indicator, see **4.4 LED Indicators of ES-1553RUNET/Px** on page 4-7.
- For ordering information see **Chapter 12: Ordering Information.**

## 1.2 Block Diagram

Figures 1-3 and 1-4 show block diagrams of the *EXC-1553UNET/Px* and *ES-1553RUNET/Px*. For information on connecting the cables, see **2.2 Connecting the Cables** on page 2-2.

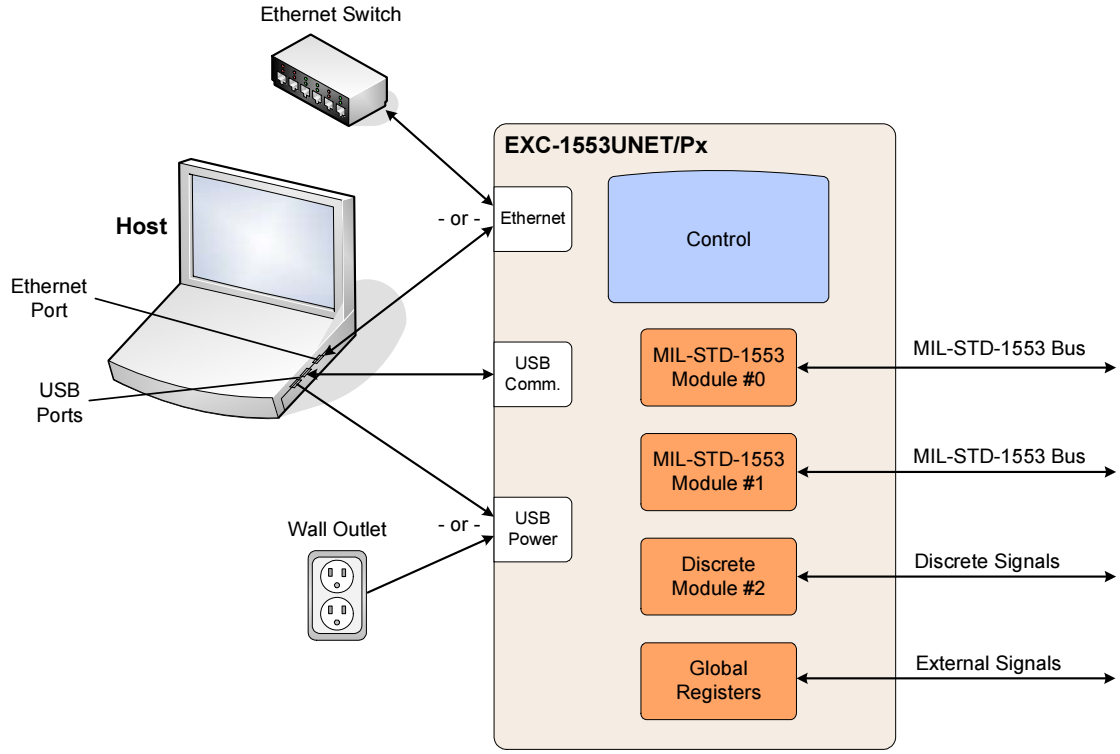



**Figure 1-3** ***EXC-1553UNET/Px*** **Block Diagram**

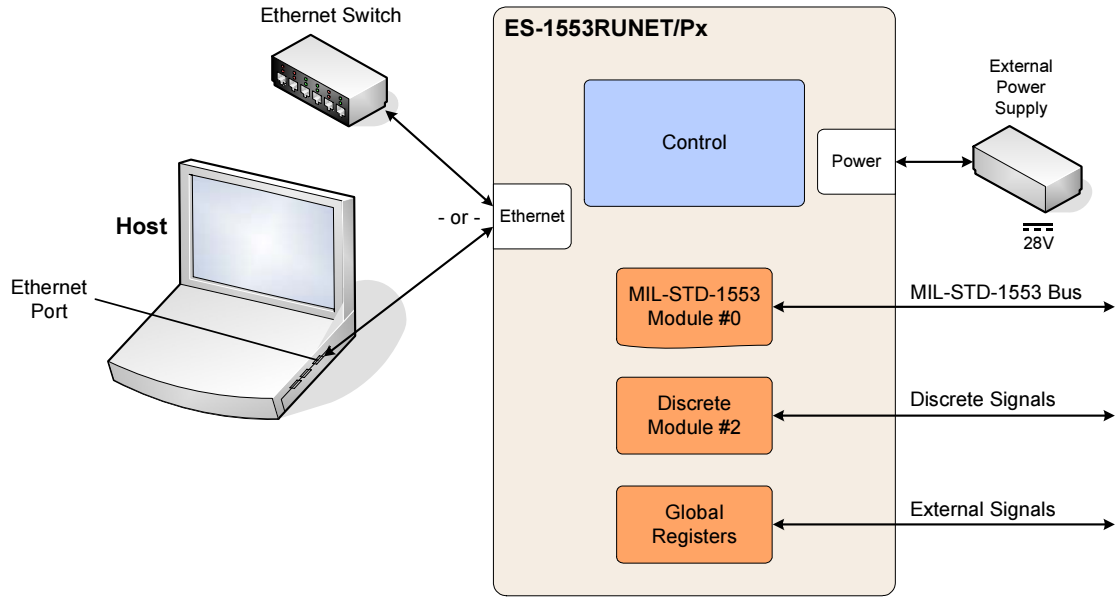



**Figure 1-4** ***ES-1553RUNET/Px*** **Block Diagram**

## 1.3 Product Features

**General Features**

- Up to two independent MIL-STD-1553A/B channels (*EXC-1553UNET/Px* only)
- Transformer or Direct Bus coupling
- Real-time operation
- 32-bit Time Tag
- Programmable Time Tag resolution (RT and Bus Monitor modes)
- Programmable interrupts
- 8 Discrete I/O signals
- IRIG B input (standard IRIG B120 serial time code)
- Smart power management
- *EXC-1553UNET/Px* power source: Computer USB ports, USB power supply
- *ES-1553RUNET/Px* power source: 11-36 VDC power supply
- *EXC-1553UNET/Px* mechanical configuration options:
- Mounting plate
- Hard-wired MIL-STD-1553 I/O cables or panel mounted connectors

**Channel Specifications**

- 64K x 8 dual-port RAM
- Independent MIL-STD-1553 dual-redundant channels
- Multifunction channels operate as multiple RTs, BC/ Concurrent-RT or Triggerable Bus Monitor
- Single function channels operate as a single RT, BC or Triggerable Bus Monitor
- Internal Concurrent Monitor (RT and BC/RT modes)
- Both 1553A and 1553B protocol capability
- Multiple-RT simulation (up to 32 RTs)
- Multibuffering of data (RT mode)
- Major/Minor frames (BC mode)
- Asynchronous frames (BC mode)
- Automatic retry (BC mode)
- Programmable broadcast mode
- Service Request Processing (SRP)
- Loopback mode (Bus A to Bus B) enables complete built-in test; plus cable testing
- Error injection capabilities (multifunction only):
- Word count
- Bit count
- Incorrect sync
- Incorrect RT address
- Incorrect parity
- Non-contiguous data
- Manchester
- MIL-STD-1760 Option:
- Checksum error detection
- Checksum error injection
- Header words

**Physical Characteristics**

For *EXC-1553UNET/Px*:

- Dimensions: 98.5mm x 76mm x 18mm (not including connectors)
- Weight (/P2 basic configuration): 180g

For *ES-1553RUNET/Px*:

- Dimensions: 75.5mm (L) x 141mm (W) x 51mm (H) (not including mounting tabs and connectors)
- Weight: Approximately 700g

**Software Support**

- *Software Tools*: Advanced API with C source code. The *Software Tools* are available for several operating systems. See the Downloads section of our website.
- *MerlinPlus* MIL-STD-1553 Bus Analyzer for Windows
- *Exalt Plus*: Excalibur Analysis Laboratory Tools for Windows (optional)

**Operating Environment**

- Operating Temperature: -40° to + 75°C
- Humidity: 5% - 90% noncondensing

**Host Interface**

For *EXC-1553UNET/Px*:

- Selectable USB 2.0 or 100Mbps Ethernet
- Power (/P2 max.): 5 VDC @ 850 mA

  **Note:** In most cases the power supported through a single USB port will be enough to operate the *EXC-1553UNET/Px*.

For *ES-1553RUNET/Px*:

- 100Mbps Ethernet
- Power (max.): 11-36 VDC, 5 Watts

**Compliance (*ES-1553RUNET/Px* only)**

- MIL-STD-810G Environmental conditions
- MIL-STD-461E Electromagnetic compatibility
- MIL-STD-704E Electric power compatibility for military airborne equipment
- MIL-STD-1275B Electric power compatibility for 28 VDC electrical systems in military vehicles
- See our website for complete compliance testing reports

## 1.4 Supporting Software

The *UNET* has the following supporting software:

- *Excalibur Software Tools* – Advanced API functions written in C language that enable you to write application and diagnostic programs. For more information, see **Chapter 3: Developing Applications**.
- *Merlin+* – A Windows test and simulation program for MIL-STD-1553 bus communications data. The program enables avionics equipment testers to send/receive messages in Bus Controller, Remote Terminal and Sequential Monitor modes by defining raw data. In Sequential Monitor mode the user has the additional option of defining data with engineering units. *Merlin+* can be from our website. Go to www.mil-1553.com, click **Downloads**, and then click **Applications**.
- *Discrete Generator* – A Windows program that enables you to configure the Discrete channels on your Discrete module, run the module and view the status of incoming Discretes. You can also configure and monitor interrupts based on status of each Discrete channel. For more information, see **2.8 Running the Discrete Generator** on page 2-31.
- *Exalt Plus* (***Ex****calibur* ***A****nalysis* ***L****aboratory* ***T****ools*) – An andvanced Windows program that enables you to monitor and analyze bus activity in real-time; to record bus activity that you can replay later (even when no modules are present and the computer is not connected to the bus); and to simulate bus activity by transmitting data over the bus. This program is an optional add-on to your purchase. For more information contact an Excalibur sales representative. See **1.8 Technical Support** on page 1-12.

## 1.5 MIL-STD-1553 Channel Information

The following sections provide information about MIL-STD-1553 and MIL-STD-1760 channels.

### 1.5.1 MIL-STD-1553 A/B Multiprotocol Considerations

The *UNET* offers a wide range of options that can be used for various 1553 applications. The various options are controlled via control registers which are described in the chapters 6 through 10. The *Software Tools* provided by Excalibur access these control registers to control the 1553 module.Alternatively, you can develop your own software tools to access these control registers.

Examples of user selectable parameters are:

- Select whether or not an RT will return a status word in the event a message containing a data word error is received by the RT.
- Selectable broadcast mode
- Variable response time
- Select mode code subaddress (00000, 11111, or both)
- 1553A RT timing
- Define each bit in the 1553 Status word

### 1.5.2 MIL-STD-1553 Bus Connections

The *UNET* can be ordered as Transformer Coupled or Direct Coupled. When using the more standard Transformer Coupling mode, use stub coupler devices, which are available from Excalibur Systems. Two terminators are required for each coupler, which services a single bus, e.g., Bus A. See **Figure 1-5**. For more information, see **Chapter 12: Ordering Information.**

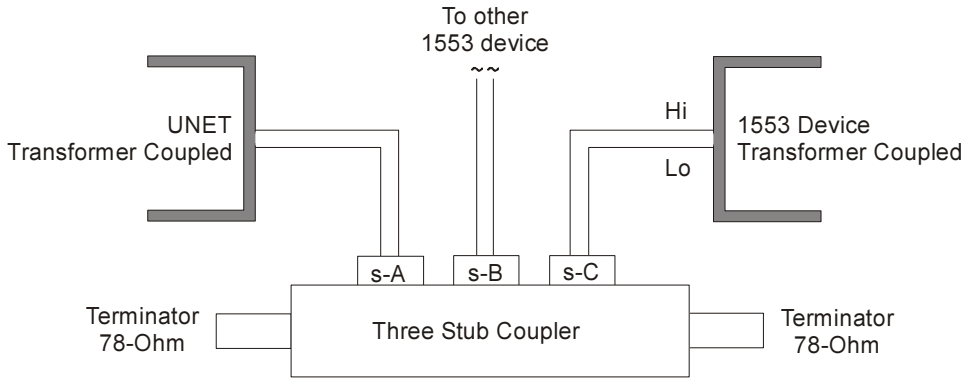



**Figure 1-5 Transformer Coupled Connection (One Bus Shown)**

For short distances, the *UNET* may be ordered as Direct Coupled to be coupled directly to another 1553 device. (See **Chapter 12: Ordering Information**.) To ensure data integrity, make certain that the cable connecting the two devices is properly terminated with 78-Ohm resistors. See **Figure 1-6**.

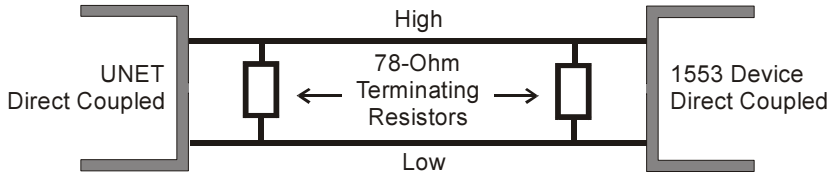



**Figure 1-6 Direct Coupled Connection (One Bus Shown)**

The *EXC-1553UNET/Px* can be ordered with hard-wired MIL-STD-1553 cables or panel mounted connectors.

The *EXC-1553UNET/Px-C* has hard-wired MIL-STD-1553 cables. Each cable ends with a female Trompeter CJ70 connector (or equivalent). The cables are marked Bus A and Bus B (in addition to the channel number). The twinax cables mate, for example, with Trompeter PL75 male connectors. These connectors are not supplied by Excalibur.

The *EXC-1553UNET/Px-M* has female Trompeter BJ157 connectors (or equivalent) mounted to the front panel of the unit. These connectors mate, for example, with Trompeter PL155 male connectors. For this configuration, Excalibur supplies adapter cables that have Trompeter PL155 (or equivalent) connectors on one side and Trompeter CJ70 (or equivalent) on the other side.

**Example of MIL-STD-1553 Bus Connection**

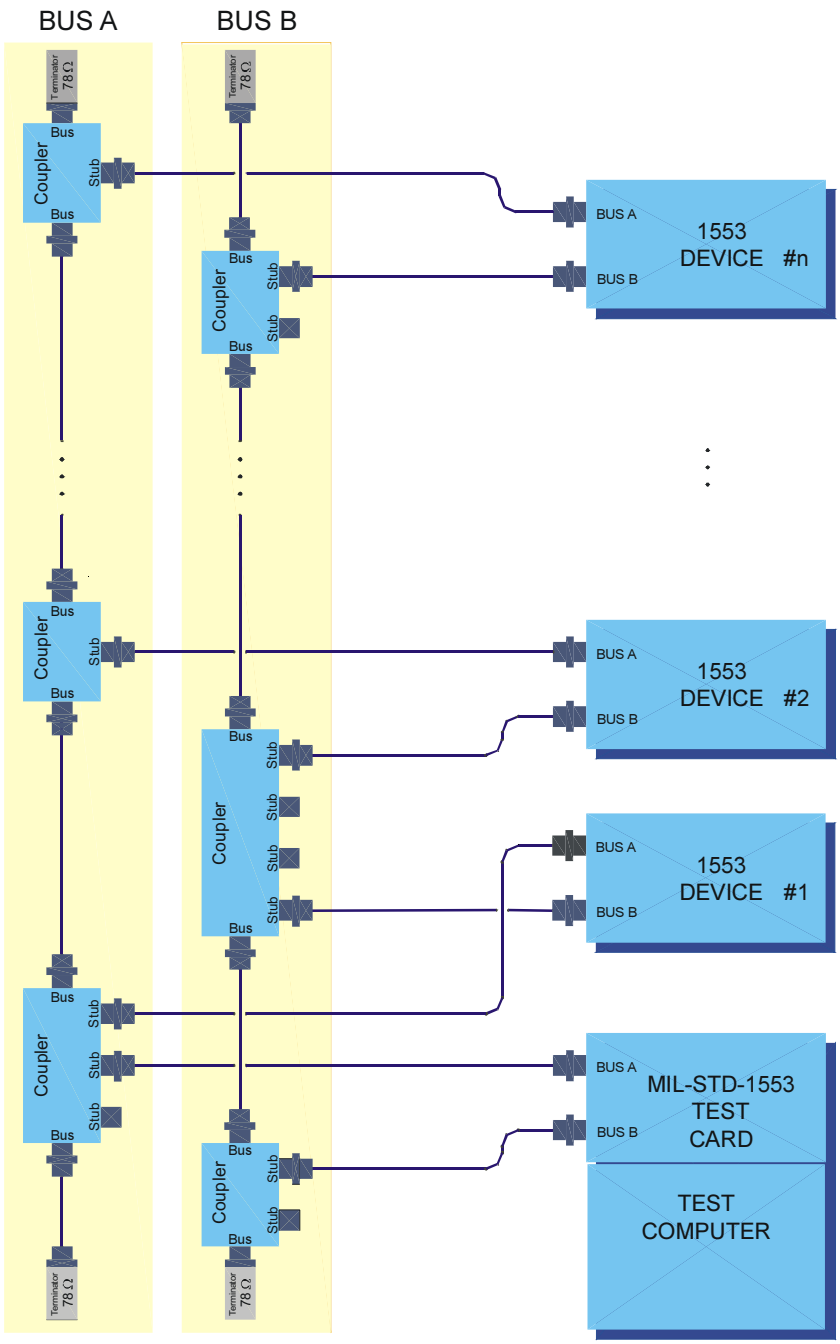



**Figure 1-7 MIL-STD-1553 Bus Connection**

### 1.5.3 1760 Option

**1760 Option only**

MIL-STD-1760 implements an enhanced MIL-STD-1553 digital interface for the transfer of digital messages to the remote terminal. The enhancements include additional error detection in the form of checksum. Checksum is mandated on critical control messages and provisional on the remainder of the messages. Implementing this level of error detection ensures a higher degree of error free data integrity requirements than only odd parity provides.

The 1760 option implements:

**Checksum generation and checksum error detection capabilities.** Checksums are calculated as each Data Word is sent or received. If the checksum flag is set on an outgoing message, the checksum will be sent in place of the last Data Word. On an incoming message, the calculated checksum is checked against the last Data Word received. If it does not match, the Checksum Error bit is set in the Message Status Word.

**Checksum error injection in BC/Concurrent-RT mode.** The user can set the checksum to intentionally send an error, giving the additional capability to test for checksum errors on the receiving RTs.

**Header Words for message identification.** The first Data Word of a message may be a Header Word. The Header Word is associated with a specific RT subaddress.

For more details see the sections on **1760 options** in **Chapter 6: MIL-STD-1553 Channel Remote Terminal Operation (Advanced)** and **Chapter 7: MIL-STD-1553 Channel BC/Concurrent-RT Operation (Advanced)**.

To order the *UNET* with the 1760 options, see **Chapter 12: Ordering Information**.

### 1.5.4 MIL-STD-1553 Single Function Option

The single function *UNET* operates as a single Remote Terminal (RT), a Bus Controller (BC) or Triggerable Bus Monitor, with an Internal Concurrent monitor for RT and BC operation.

The differences between the standard *UNET* and the single function *UNET* are as follows:

- The single function *UNET* simulates only one RT at a time.
- It has no error injection capability.
- The single RT address is provided via the RT Number register (see **6.10.1 RT Number Register (PxS Only)** on page 6-17).
- In several of the RT mode tables, only the first word (or byte) is used; the rest is reserved (see **6.2 RT Memory Map** on page 6-4).
- Illegalization is done based on the SAid, not the RTid, using the Broadcast SAid Control Table (see **6.10.6 Broadcast SAid Control Table (PxS Only)** on page 6-19).

## 1.6 MIL-STD-1553 Module General Memory Map

The MIL-STD-1553 Module General Memory Map is a summary of the major memory areas in the MIL-STD-1553 . Detailed memory maps for each of the module's modes are provided later in this manual.

The MIL-STD-1553 module occupies 64K bytes of the module's 128K-memory space. These 64K bytes are shared between the Control registers and the Data Block.

| | |
|---|---|
| *Mode Specific Registers* | 7010 – FFFF H |
| Hardware Revision Register | 700E – 700F H |
| *Mode Specific Registers* | 700C – 700D H |
| Time Tag (Hi) | 700A – 700B H |
| Time Tag (Lo) | 7008 – 7009 H |
| Time Tag Reset Register | 7007 H |
| *Mode Specific Registers* | 7001 – 7006 H |
| Reset Register | 7000 H |
| *Mode Specific Registers* | 4000 – 6FFF H |
| Configuration Register | 3FFF H |
| ID Register | 3FFE H |
| Status Register | 3FFD H |
| Start Register | 3 FFC H |
| *Mode Specific Registers* | 3E86 – 3FFB H |
| Options Register | 3E84 – 3E85 H |
| *Reserved* | 3E81 – 3E83 H |
| Firmware Revision Register | 3E80 |
| *Mode Specific Registers* | 0000 – 3E7F H |

**Figure 1-8 *M4K1553Px* General Memory Map**

## 1.7 Discrete Channel Information

The following sections describe the Discrete voltage/current levels and configuration.

### Discrete Channel Configuration

The input voltage level and debounce setting for each input Discrete are set in the Discretes Configuration Registers, see **Discretes Configuration Registers** on page 11-12, or via the *M4KDiscrete Software Tools*, see the *M4KDiscrete Software Tools Programmer's Reference*.

The following sections provide more information about input and output Discretes.

### Input Discrete Voltage Levels

Input Discretes can operate at either TTL or Avionics voltage levels, as defined in the following table.

| Type | Voltage | | Value |
|---|---|---|---|
| TTL | 0V – 0.8V | = | Logic 0 |
| | 2V – 5V | = | Logic 1 |
| Avionics | 0V – 3V | = | Logic 0 |
| | 7.5V – 32V | = | Logic 1 |

**Table 1-1 Input Discrete Voltages**

In the standard *UNET*, the voltage level must be set externally by pulling up the Discrete I/O line to the desired voltage level, 5V for TTL or 7.5V – 32V for Avionics. The recommended value for the pull up resistors is 10K Ohms. When the line is not grounded, the recorded Discrete value will be 1 and when it is grounded the value will be 0.

A custom *UNET* can be ordered with the voltage pulled up internally to the desired voltage level. For more information, contact our Sales Department. See **1.8 Technical Support** on page 1-12.

**Note:** The voltage level of each Discrete must be set in the Discretes Configuration Registers, see **Discretes Configuration Registers** on page 11-12, or via the *M4KDiscrete Software Tools*, see the *M4KDiscrete Software Tools Programmer's Reference*. If a Discrete is set to Avionics voltage level and the voltage is (incorrectly) pulled up to 5V, the Discrete will always record a value of 0.

### Input Discrete Debounce Setting

Each input Discrete can be configured with or without debounce. When debounce is activated, any change on the input Discrete will only be registered after 60 msec.

### Output Discretes

All output Discretes are open collector. Each output Discrete must be pulled up to the required voltage (maximum of 32V) externally to the *UNET*. A maximum

of 100 mA can be sunk on any individual output Discrete. The external pull up resistor should be chosen carefully to ensure that the maximum sink current is not exceeded. The recommended value for the pull up resistors is 10K Ohms.

When an output Discrete is set to 0, the output Discrete will be shorted to ground. When an output Discrete is set to 1, the output Discrete will be left floating and thus pulled up to the voltage used for that output Discrete via the external pull up resistor.

## 1.8 Technical Support

Excalibur Systems is ready to assist you with any technical questions you may have. For technical support, click the **Technical Support** link on the **Support** page of our website: www.mil-1553.com. You can also contact us by phone. To find the location nearest you, refer to the **Contact Us** page of our website. For corrections to this manual, email our Technical Writing Department directly at: tw@mil-1553.com. Please include the name of the manual, the manual revision number (located on the last page of the manual), and the location within the manual.

# 2 Installation and Setup

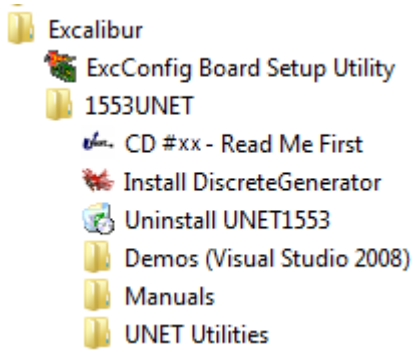
Chapter 2 provides installation and setup information for the *UNET*.

## 2.1 Installing the Software on the Host Computer

This section describes how to install the *UNET* software on the host computer that will be used to control the *UNET* via USB or Ethernet.

**To install the *UNET* software:**

1. Insert the *Excalibur Installation CD* into your computer's CD drive.
   The InstallShield Wizard is displayed.
2. Follow the on-screen instructions.
   The software is installed, and the following items are added to the Start menu:




**Note:** In the above figure, "**xx**" is replaced by the CD number of the current *UNET* installation.

## 2.2 Connecting the Cables

This section describes how to connect the *UNET*'s cables. The *EXC-1553UNET/Px* can communicate with host via USB or Ethernet. For cabling instructions for both configurations, see **2.2.1 Connecting the EXC-1553UNET/Px Cables** on page 2-3.

The *ES-1553RUNET/Px* can interface with host only through Ethernet. For cabling instructions, **2.2.2 Connecting the ES-1553RUNET/Px Cables** on page 2-5.

**Note:** When setting up the *UNET*, it is recommended to check the *UNET*'s status via its LED states. See **4.3 LED Indicators of EXC-1553UNET/Px** on page 4-5 or **4.4 LED Indicators of ES-1553RUNET/Px** on page 4-7.

### 2.2.1 Connecting the *EXC-1553UNET/Px* Cables

**Cabling for USB Communication with Host**

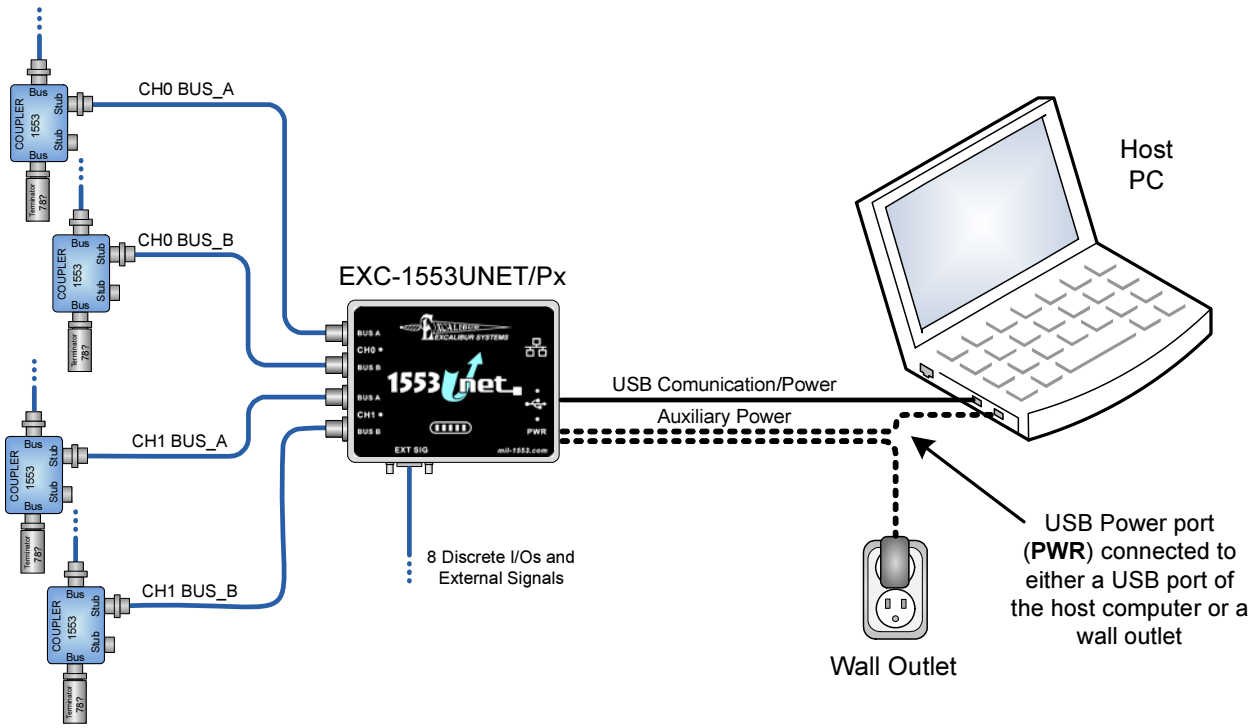



**Figure 2-1** USB Communication with Host

**To connect the cables:**

1. Connect the MIL-STD-1533 I/O connectors of the *EXC-1553UNET/Px* to the MIL-STD-1533 bus.
2. Connect the USB Communication port ⇻ of the *EXC-1553UNET/Px* to the USB port of the host computer using the Micro-B USB to Standard-A USB adapter cable provided by Excalibur.
3. In most cases (depending on the load on MIL-STD-1533 bus and capabilities of the host computer), the power supported through the USB Communication port ⇻ will be enough to operate the *EXC-1553UNET/Px*. If you require more power, the USB Power port (**PWR**) of the *EXC-1553UNET/Px* should be connected to another USB port of the host computer or to a wall outlet using the 5V USB Power Supply provided by Excalibur Systems. For power information when using an internal battery, see **4.7.1 Battery Option Power Information** on page 4-17.

*Caution:* Use only the USB Power Supply provided by Excalibur Systems. Excalibur Systems is not responsible for any damage or lack of functionality when a different power supply is used.

**Cabling for Ethernet Communication with Host**

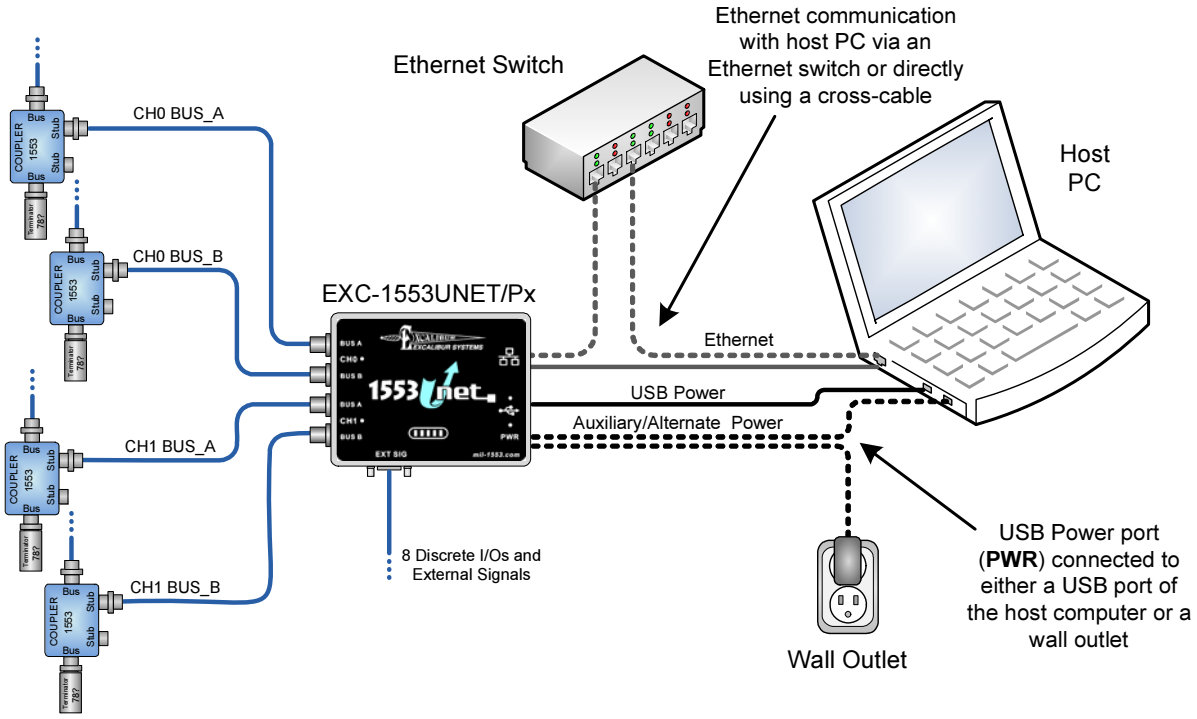



**Figure 2-2** Ethernet Communication with Host

**To connect the cables:**

1. Connect the RJ45 port of the *EXC-1553UNET/Px* to an Ethernet switch or to the host computer via an Ethernet cable (not included).

**Note:** When connecting the *EXC-1553UNET/Px* directly to the host computer you will need a cross-cable, unless the computer's network card can automatically switch the Ethernet signals.

2. Connect the MIL-STD-1533 I/O connector of the *EXC-1553UNET/Px* to the MIL-STD-1533 bus.
3. To supply power to *EXC-1553UNET/Px*, connect the USB Communication port to the USB port of any computer. In most cases (depending on the load on MIL-STD-1533 bus and the capabilities of the computer), the power supplied through the USB Communication port will be enough to operate the *EXC-1553UNET/Px*. If you require more power, the USB Power port (**PWR**) should be connected to another USB port of the computer or to a wall outlet using the 5V USB Power Supply provided by Excalibur Systems. Each of these three power sources (USB Communication port to PC USB, USB Power port (**PWR**) to PC USB or USB Power port (**PWR**) to wall outlet) can be used separately as the prime power source. For power information when using an internal battery, see **4.7.1 Battery Option Power Information** on page 4-17.

*Caution:* Use only the USB Power Supply provided by Excalibur Systems. Excalibur Systems is not responsible for any damage or lack of functionality when a different power supply is used.

### 2.2.2 Connecting the *ES-1553RUNET/Px* Cables

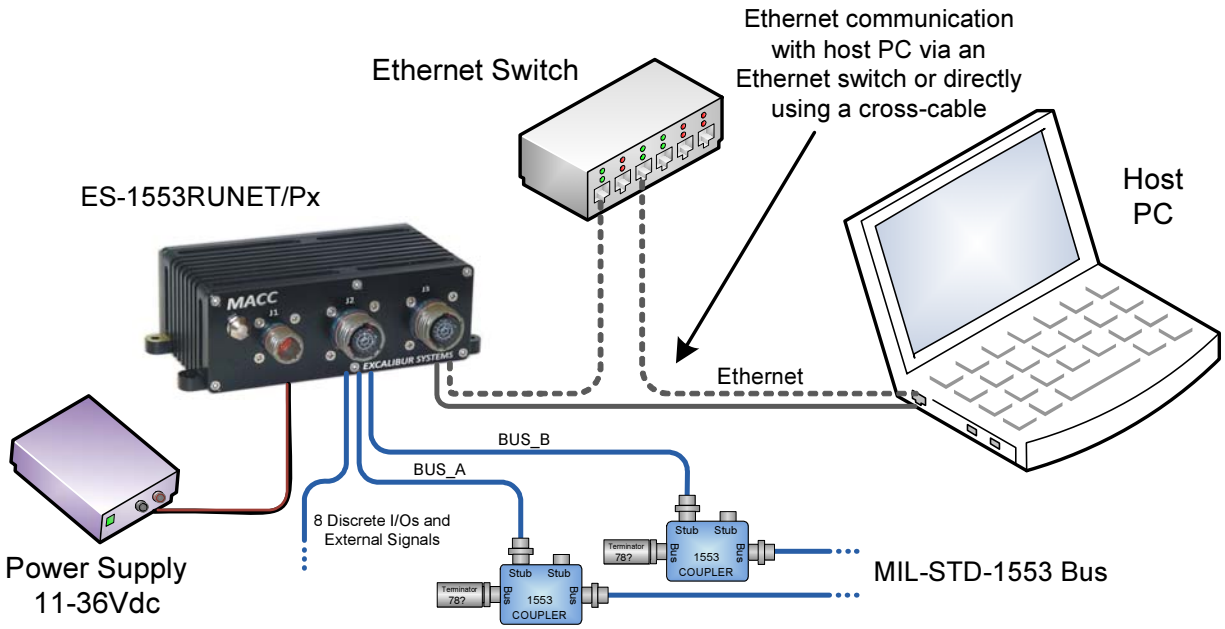



**Figure 2-3 *ES-1553RUNET/Px* Communicating with the Host via Ethernet**

**To connect the cables:**

1. Connect the I/O signals of connector J2 to the MIL-STD-1533 bus and (if required) to the Discrete connections.
2. Connect the I/O signals of connector J3 to the Ethernet network and (if required) to the IRIG B source and/or the 1553 RT address source (for a single function configuration).

**Note:** When connecting the *ES-1553RUNET/Px* directly to the host computer you will need a cross-cable, unless the computer's network card can automatically switch the Ethernet signals.

3. Connect the DC power source (11-36V, 5W) to connector J1.

*Caution:* On the ES-1553RUNET/Px, make sure the power is OFF while connecting or disconnecting communication cables to or from the device. Connecting or disconnecting cables to or from the ES-1553RUNET/Px while the power is ON can seriously damage the ES-1553RUNET/Px or a system connected to the ES-1553RUNET/Px.

## 2.3 Installing the USB Driver on the Host Computer

**Note:** USB driver installation is only required when using the USB interface for communication. When using the USB port for power only, USB driver installation is not required.

When you connect the *UNET* to a computer connected to the Internet, Windows automatically finds the USB driver. In the event that the computer is not connected to the Internet, install the device driver using the Device Manager.

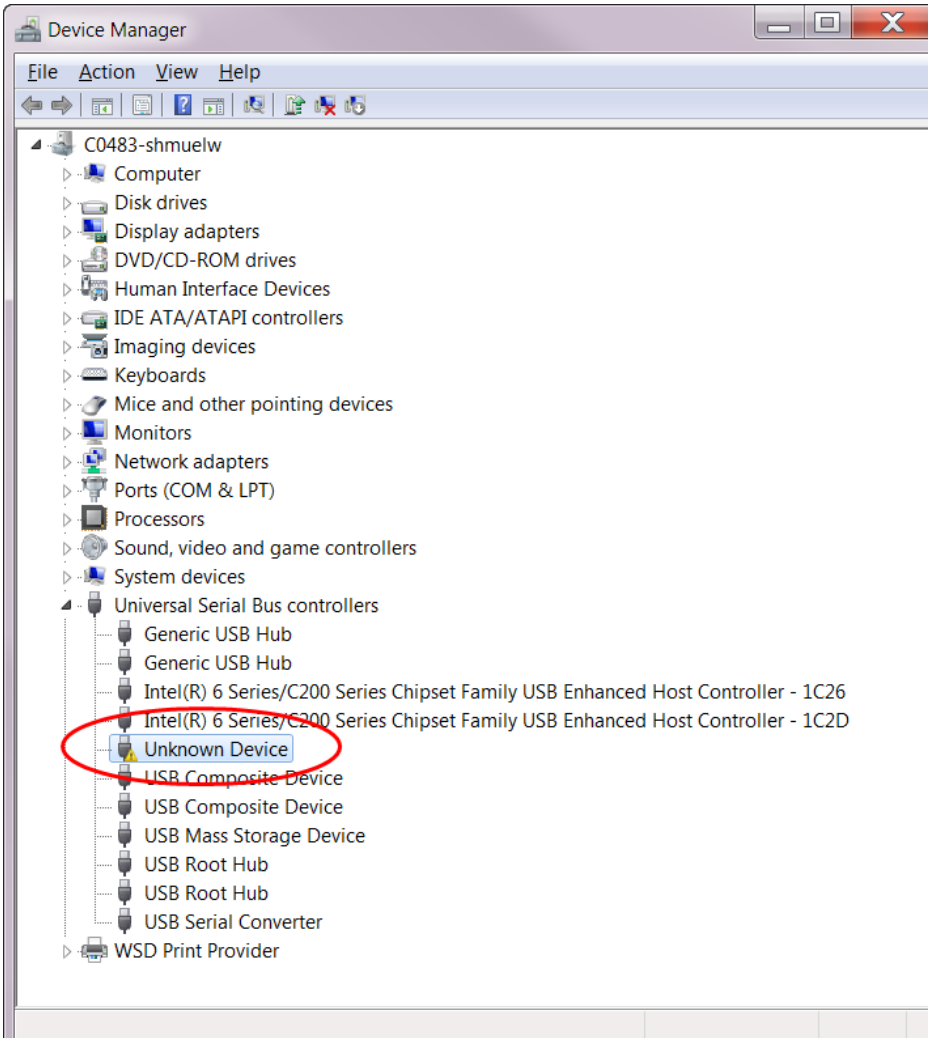
**To install the USB driver manually:**

1. Connect one of the USB cables to the host computer and to the USB Communication port ⇻ of the *EXC-1553UNET/Px*.
2. Right-click on **My Computer**, and select **Properties** from the context menu.
3. Do one of the following:

   For Windows 7: Click **Device Manager**.

   For Windows XP: Click the **Hardware** tab, then click **Device Manager**.

   The Device Manager is displayed.
4. Expand the Universal Serial Bus controllers group of devices.



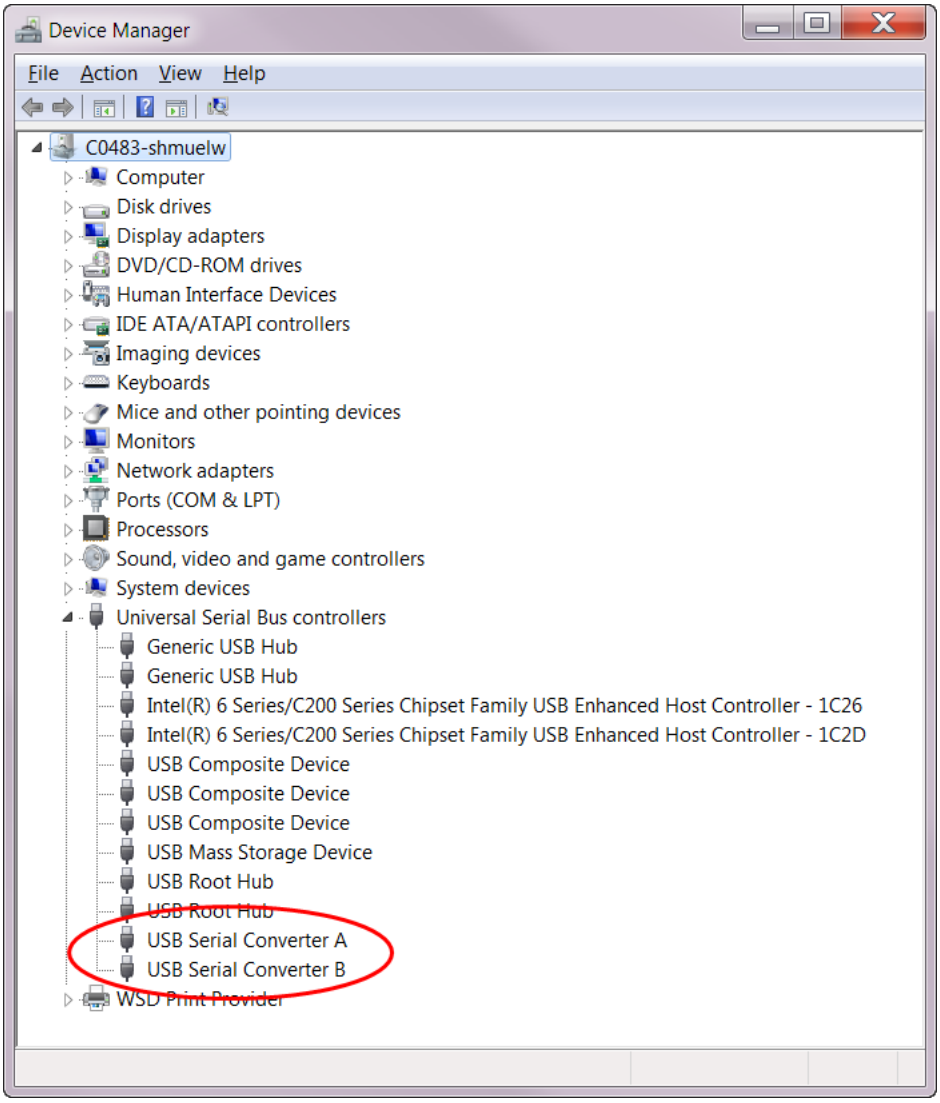
5. Right-click the unknown USB device, and select **Update Driver Software** from the context menu.
6. Choose the option to browse your computer for the driver.
7. Select the folder: **C:\Excalibur\1553UNET\USB Driver**. (This path assumes the default software installation location.)
8. Click **Next**.

   The driver is installed.

**Note:** In the event that the installation is not successful, try connecting the USB cables to the front USB ports or to other USB ports, and reinstalling the driver. Also, do not use an extension cable during the driver installation.

9. Disconnect the USB cable(s), then reconnect the cables.
10. Open the Device Manager and verify that the USB driver was installed correctly.

    If the driver is installed correctly, USB Serial Converter A and USB Serial Converter B appear in the Universal Serial Bus controllers group.



## 2.4 Assigning a Device Number on the Host Computer

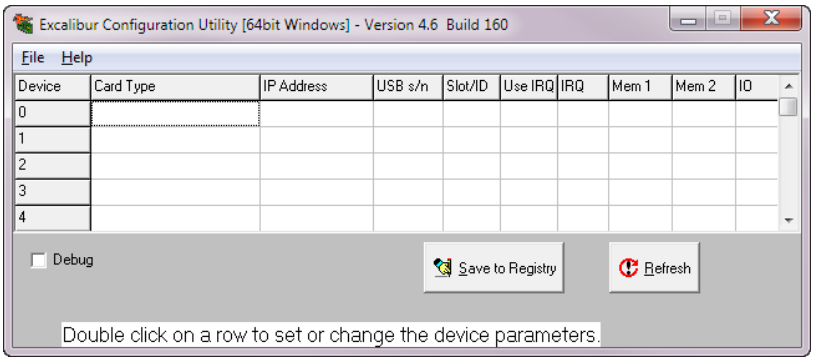
This section describes how to use the ExcConfig utility to assign a device number to the *UNET* and save the device number in the Windows System Registry. The device number is required as a parameter of several of the *Software Tools* functions.

**Note:** When assigning a device number to the *UNET*, the instructions differ for communicating with the *UNET* via USB or Ethernet.

**To assign a device number to the *UNET*:**

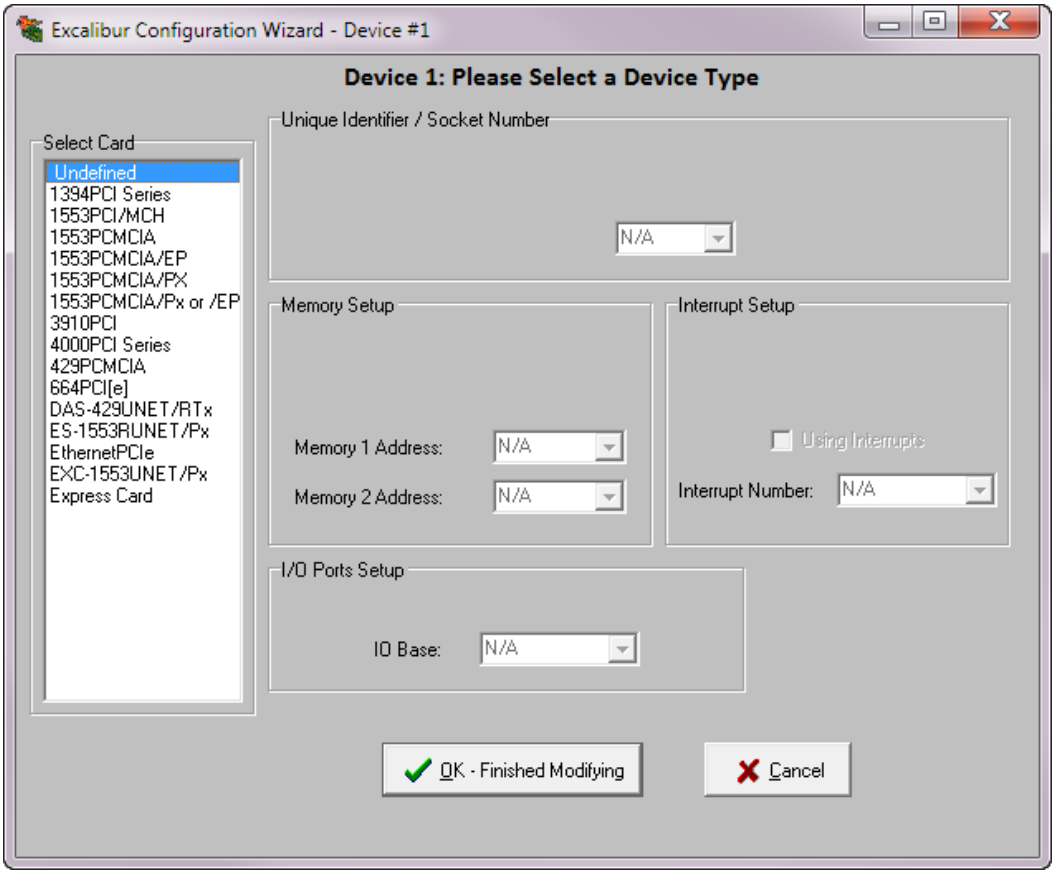
1. Run the ExcConfig: Click **Start | All Programs | Excalibur**, right-click **ExcConfig Board Setup Utility**, then select **Run as administrator**. (When using Windows XP, running as administrator is not required.)

   The ExcConfig main screen is displayed, showing all Excalibur devices saved in the Windows System Registry of this computer. Each row on the ExcConfig main screen is associated with a device number (0–15).




2. Double-click one of the cells in an empty row, for example, Device #1.

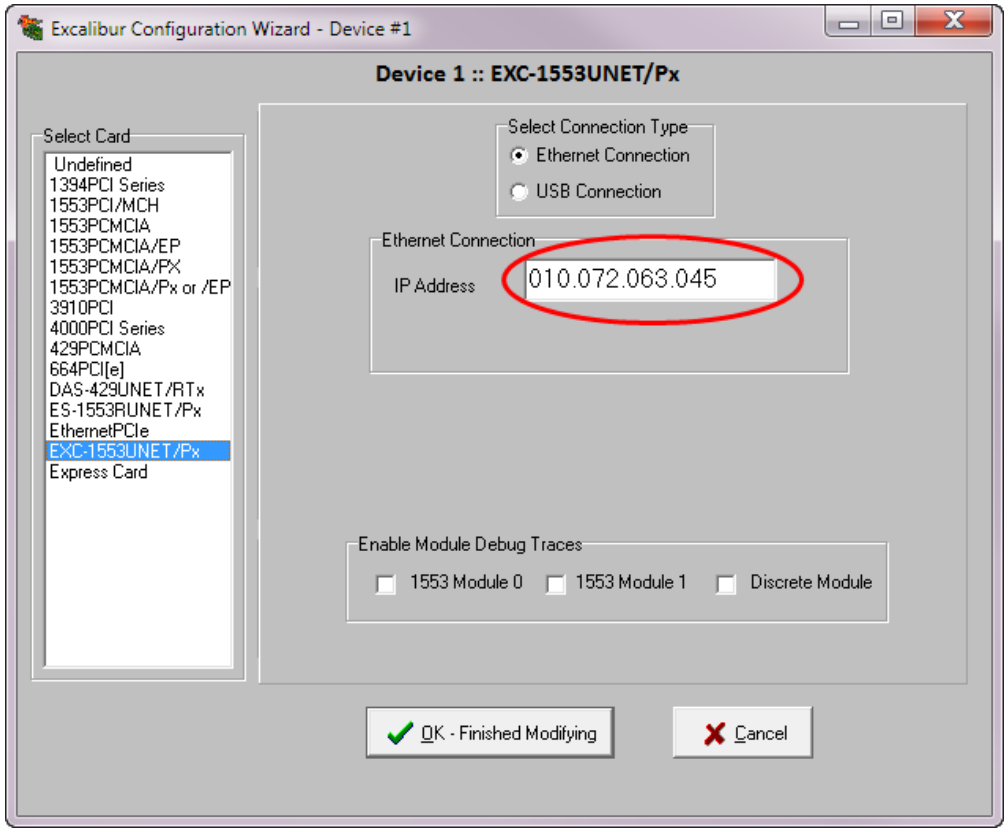
The Excalibur Configuration Wizard is displayed.




3. In the Select Card section, select **EXC-1553UNET/Px** or **ES-1553RUNET/Px**.
4. Do one of the following:
   - When using an *EXC-1553UNET/Px* and communicating with the host via Ethernet, continue with Step 5.
   - When using an *EXC-1553UNET/Px* and communicating with the host via USB, continue with Step 10.
   - When using an *ES-1553RUNET/Px*, continue with Step 15.

5. In the Select Connection Type section, click the **Ethernet Connection** option button.

   The IP Address field is displayed with the default IP address of the *EXC-1553UNET/Px*.




**Note:** The following step is only required when there will be two or more *UNET*'s on the same network. In this case, each *UNET* must have a unique IP address. Otherwise, you can leave the default IP address.

6. When using two or more *UNET*'s on the same network, do the following: In the Ethernet Connection section, in the IP Address field, type the IP address that you want to use for the *UNET*.

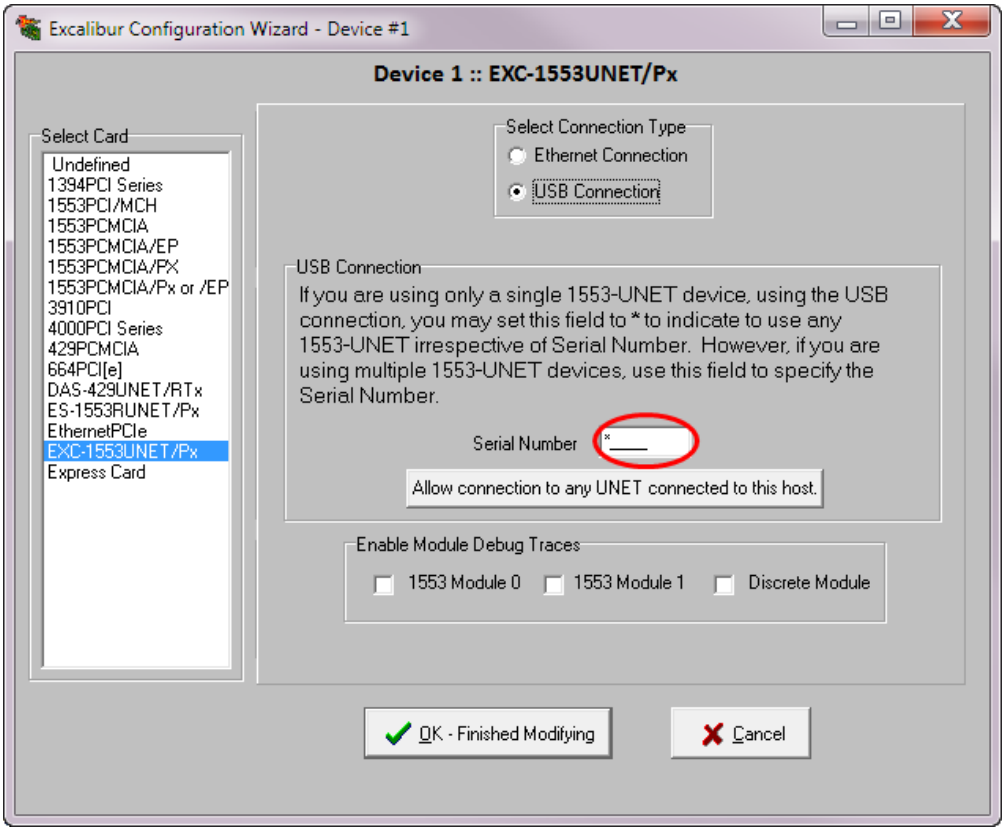
**Note:** ExcConfig saves these values in the Windows registry, but does not download the new IP address to the *UNET*. Downloading the IP address to the *UNET* will be done later. (See **2.5 Setting the UNET's Ethernet Settings** on page 2-13.)

7. Only when instructed by Excalibur Support, in the Enable Module Debug Traces section, select one more of the modules: **1553 Module 0**, **1553 Module 1, Discrete Module**. This creates a log file that traces the communication between the host and the channels (modules) on the *UNET*. Note that when this debug functionality is not required, it is recommended to clear these checkboxes.
8. Click **OK - Finished Modifying**.

   The configuration details of the device are displayed on the ExcConfig main screen.
9. Continue with Step 18 on page 2-12.

10. (This step is a continuation from Step 4 when using an *EXC-1553UNET/Px* and communicating with the host via USB.) In the Select Connection Type section, click the **USB Connection** option button.

    The Serial Number field is displayed.




**Note:** The following step is only required when there will be two or more *UNET*'s of the same type (1553 or 429) connecting to the computer via USB. Otherwise, you can skip the next step and leave the asterisk (*). This will allow this device number to communicate with any *UNET* connected to the computer via USB.

11. Type the serial number located at the back of the *UNET* in the Serial Number field. This will assign the selected device number to this *UNET* only.

**Note:** You can click the **Allow connection to any UNET connected to this host** button to remove the serial number in the Serial Number field and replace it with an asterisk.

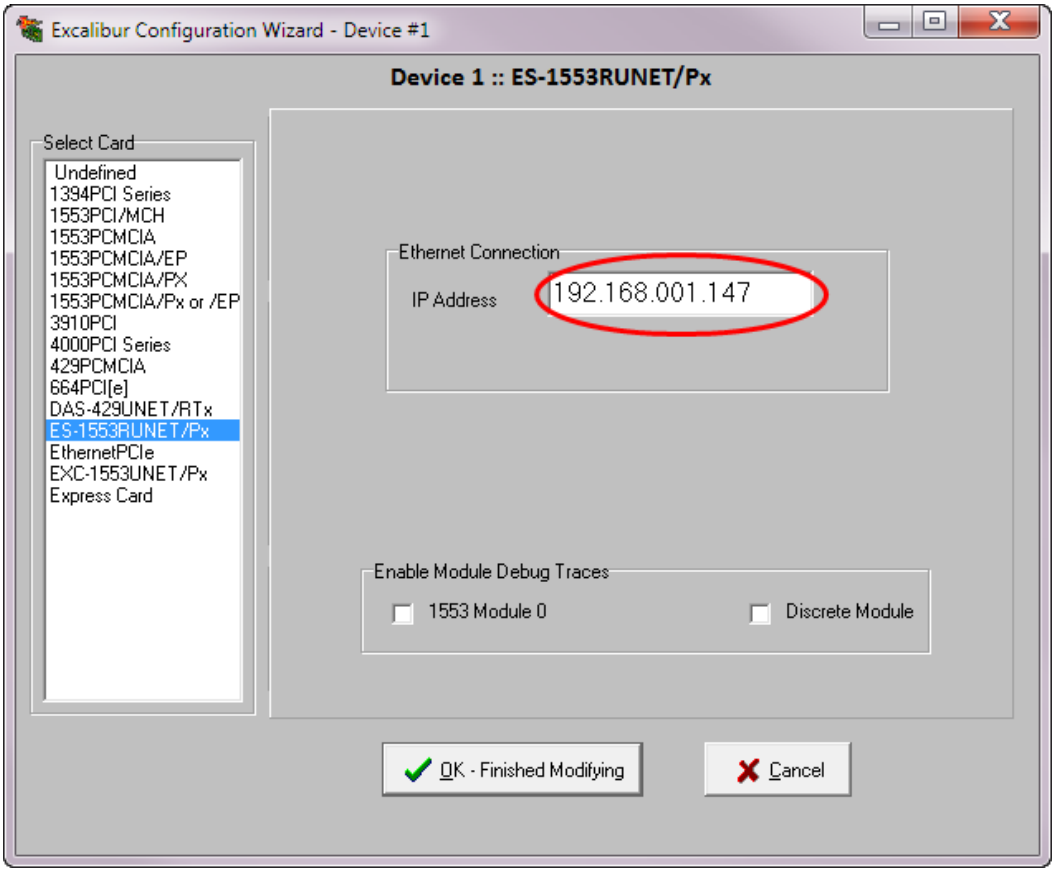
12. Only when instructed by Excalibur Support, in the Enable Module Debug Traces section, select one more of the modules: **1553 Module 0**, **1553 Module 1**, **Discrete Module**. This creates a log file that traces the communication between the host and channels (modules) on the *UNET*. Note that when this debug functionality is not required, it is recommended to clear these checkboxes.

13. Click **OK - Finished Modifying**.

    The configuration details of the device are displayed on the ExcConfig main screen.

14. Continue with Step 18 on page 2-12.

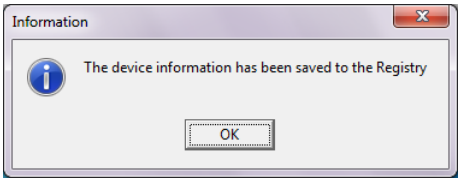
15. (This step is a continuation from Step 4 when using an *ES-1553RUNET/Px*.) In the Ethernet Connection section, in the IP Address field, type the user-defined IP address of the *ES-1553RUNET/Px*.




16. Only when instructed by Excalibur Support, in the Enable Module Debug Traces section, select **1553 Module 0** and /or **Discrete Module**. This creates a log file that traces the communication between the host and the channels (modules) on the *ES-1553RUNET/Px*. Note that when this debug functionality is not required, it is recommended to clear these checkboxes.
17. Click **OK - Finished Modifying**.

    The configuration details of the device are displayed on the ExcConfig main screen.
18. To save the configuration, click **Save to Registry** on the ExcConfig main screen.

    The following confirmation dialog box is displayed.




19. Click **OK**.
20. Click the **X** at the top-right of the ExcConfig main screen to exit the ExcConfig utility.

## 2.5 Setting the *UNET*'s Ethernet Settings

**Note:** This section is only relevant when using Ethernet for communication between the host and the *UNET*.

The following sections describe how to set the user-defined IP address and MAC address of the *UNET*. When there are two or more *UNET*'s on the same network, it is important to change the *UNET*'s IP address and MAC address, and not to continue using the default IP address and MAC address.

### 2.5.1 Overview of Setting the *UNET*'s Ethernet Settings

On the *EXC-1553UNET/Px*, there are two ways to set the user-defined IP address and MAC address: via the *UNET*'s USB Communication port or via Ethernet. On the *ES-1553RUNET/Px* you must use Ethernet. When the USB option is available, it is recommended to set the user-defined IP Address and MAC address via USB.

The following sections provide detailed instructions for setting the user-defined IP address and MAC address via USB and via Ethernet.

### 2.5.2 Setting the Ethernet Settings via USB

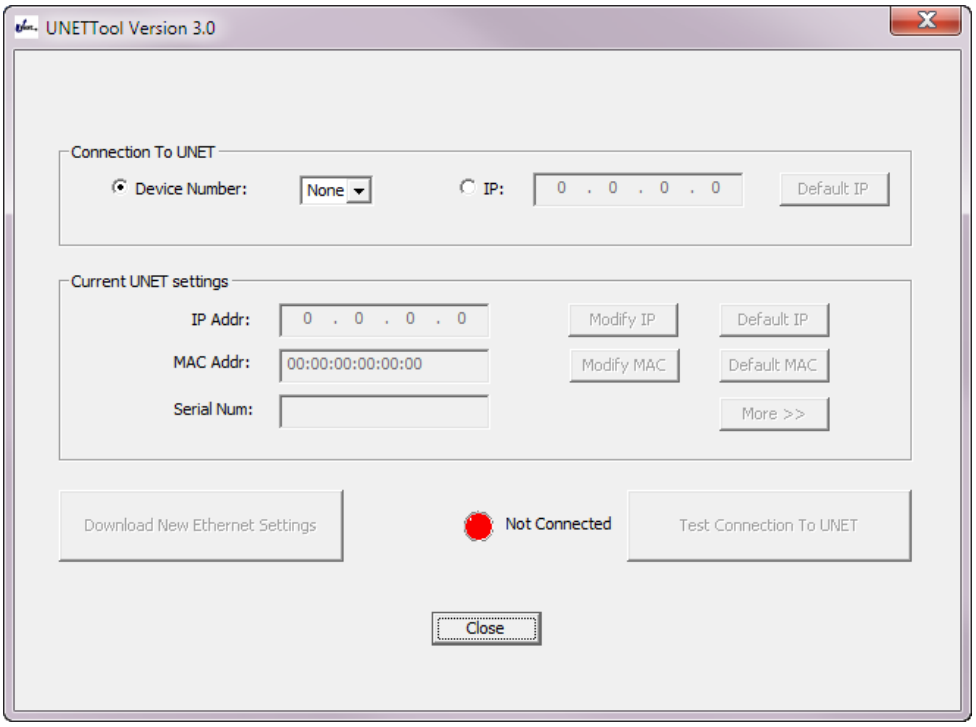
**To set the Ethernet Settings via USB:**

1. Connect one side of one of the supplied USB cables to one of the computer's available USB ports and the other side to the USB Communication port of the *EXC-1553UNET/Px*, and make sure the LED is lit.
2. Install the USB driver on the host computer (if it was not already installed). See **2.3 Installing the USB Driver on the Host Computer** on page 2-6.
3. Assign a device number selecting USB as the connection type and leave the *UNET*'s default serial number (*). See **2.4 Assigning a Device Number on the Host Computer** on page 2-8.

**Note:** You can have two device numbers assigned to the same *UNET*, one for communicating via USB and one for Ethernet.

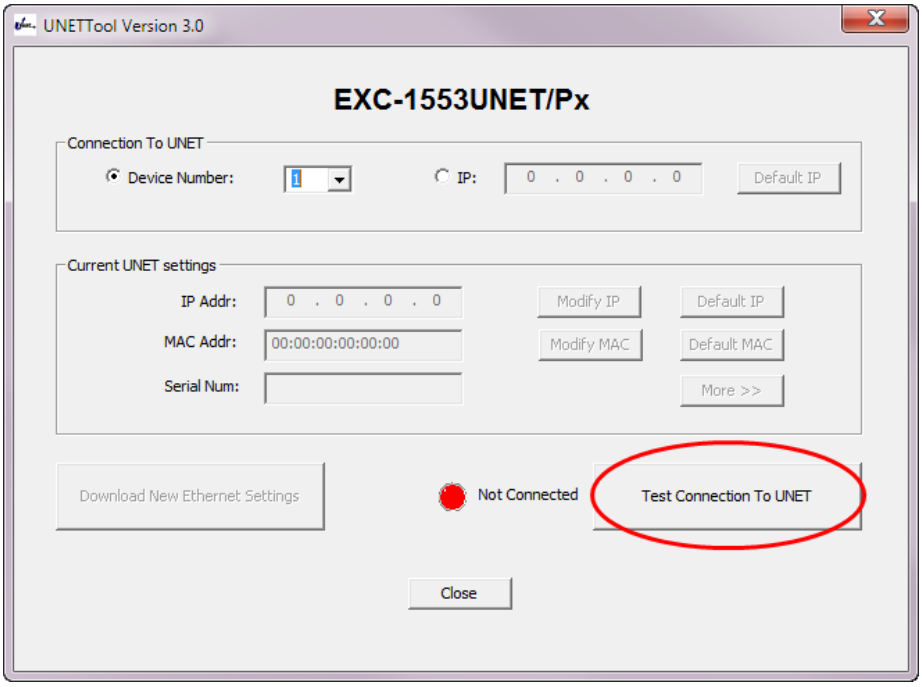
4. Run the UNETTool: Click **Start | All Programs | Excalibur | 1553UNET | UNET Utilities | UNET GUI Tool**.

   The UNETTool main screen is displayed.



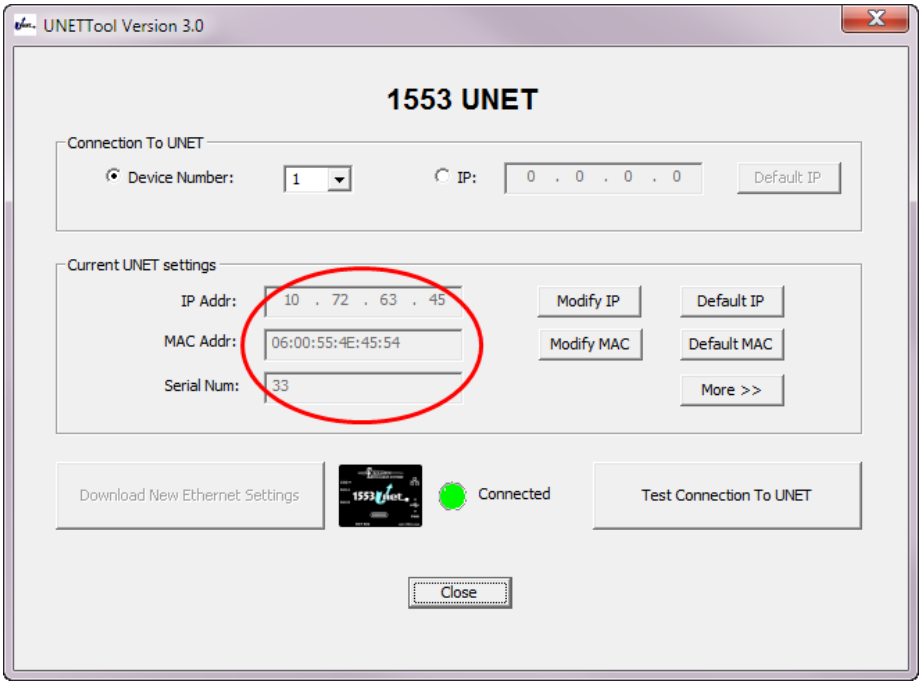

5. In the Connection to UNET section of the screen, select the device number that was assigned to the *EXC-1553UNET/Px* in Step 3, for example, Device #1.

The Test Connection To UNET button becomes active.

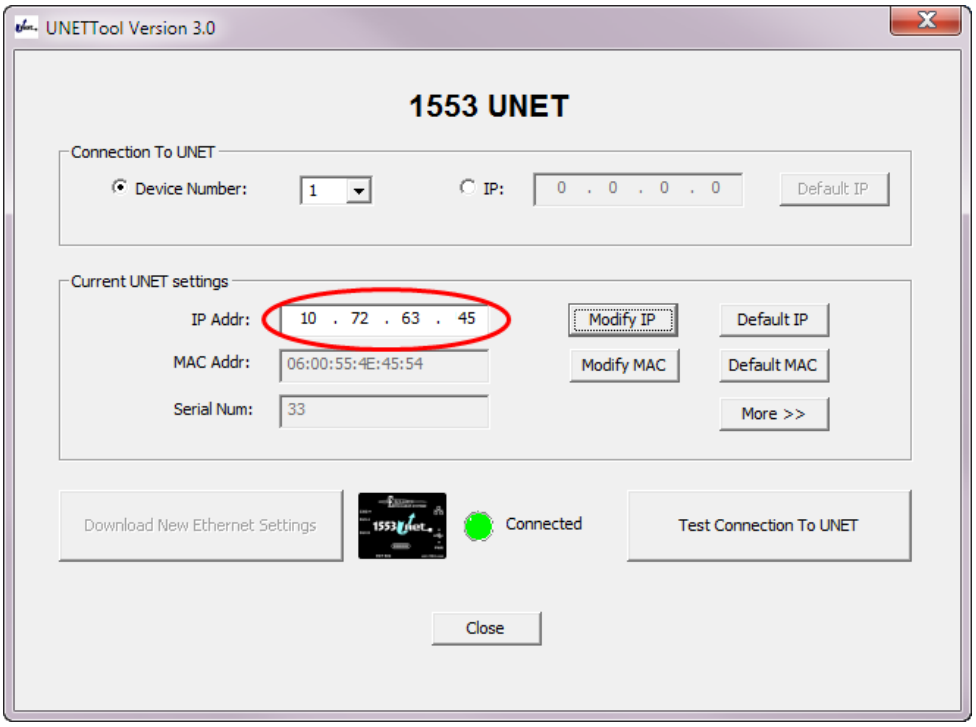



6. Click **Test Connection To UNET**.

   The UNETTool attempts to connect to the *UNET* using the specified device number. When the UNETTool succeeds in connecting to the *UNET*, the LED on the UNETTool turns green, and the *UNET*'s IP address, MAC address and serial number are displayed in the Current UNET settings section of the UNETTool main screen.

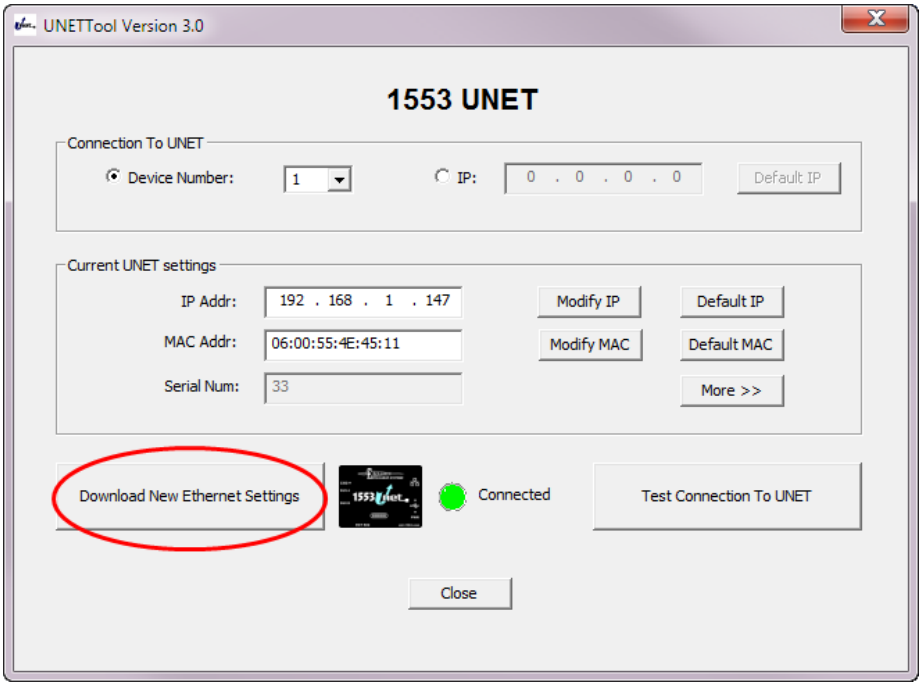


7. Click **Modify IP**.
   The IP Addr field becomes read/write.



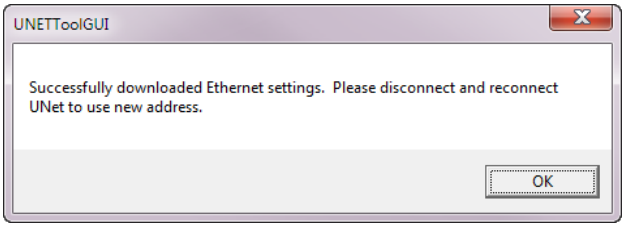

8. In the IP Addr field, type the desired user-defined IP address, for example, 192.168.1.147. If you are not sure of what IP address to use, ask your system administrator for a valid IP address on your network.
9. (Optional) To revert to the default IP address, click **Default IP**.
10. Click **Modify MAC**.
    The MAC Addr field becomes read/write.
11. In the MAC Addr field, type the desired user-defined MAC address, for example, 06:00:55:4E:45:11. If you are not sure of what MAC address to use, ask your system administrator for a valid MAC address.
12. (Optional) To revert to the default MAC address, click **Default MAC**.

The Download New Ethernet Settings button is enabled.



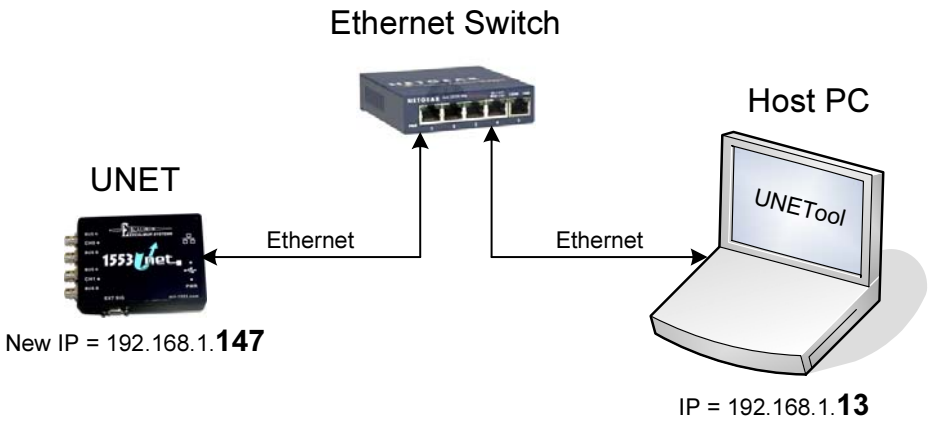

13. Click **Download New Ethernet Settings**.

    The following dialog box is displayed.



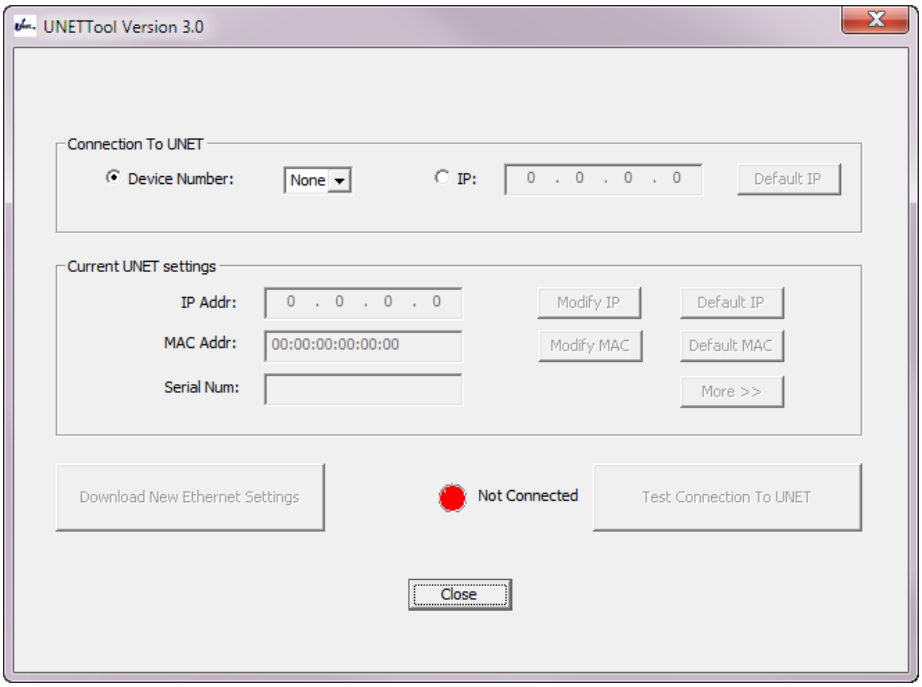

14. Click **OK**.
15. In order for the changes to take effect, restart the *UNET* as follows: Disconnect all USB cables from the *UNET* for several seconds, then reconnect them.
16. Click **Close** to exit the UNETTool.

17. Connect the *UNET* to the host PC via Ethernet. The following figure is an example.



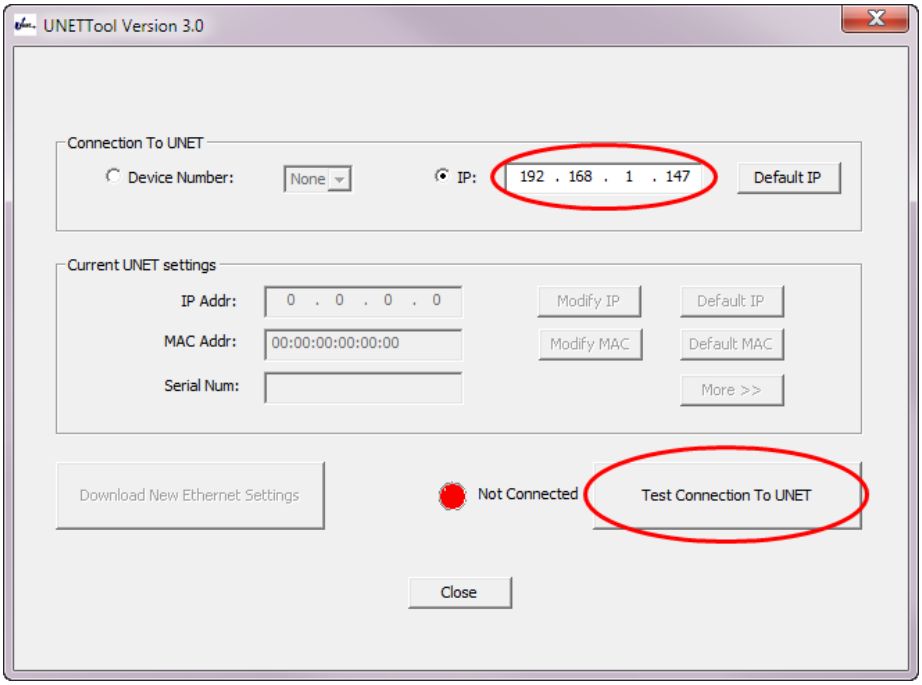

18. Turn off your computer's firewall. When using Windows 7 or later, make sure the firewall of both public and private networks is turned off.
19. Reopen the UNETTool: Click **Start | All Programs | Excalibur | 1553UNET | UNET Utilities | UNET GUI Tool**.

    The UNETTool main screen is displayed.



20. In the Connection to UNET section, click the **IP** option button, then type the new IP address of the *UNET*.

    The Test Connection to UNET button is enabled.




21. Click **Test Connection To UNET**.

    The UNETTool attempts to connect to the *UNET* via Ethernet using the new IP address. When the UNETTool succeeds in connecting to the *UNET*, the LED on the UNETTool turns green, and the new IP address and MAC address are displayed in the Current UNET settings area of the UNETTool. This confirms that user-defined IP address and MAC address was modified successfully.

22. Turn the firewall back on.
23. Add the *UNET*'s user-defined IP address as an exception to the firewall.
24. Test the connection to the *UNET* again (steps 20 and 21).
25. Click **Close**.

### 2.5.3 Setting the Ethernet Settings via Ethernet

When connecting via Ethernet to the *UNET* for the first time, the **default IP** (10.72.63.45) must be used since the *UNET* does not yet have a user-defined IP address set. When connecting to the *UNET* using the default IP, make sure there is only one *UNET* connected to the network, since any *UNET* on the network will respond to commands sent to the default IP address.

In order to connect to the *UNET* with the default IP address you must temporarily change the Local Area Connection settings on the host computer to configure the host computer to be on the same network as the *UNET*, then set the *UNET*'s user-defined IP address and MAC address.

**Note:** When there are two or more *UNET*'s on the same network, it is important to change the *UNET*'s IP address and MAC address, and not to continue using the default IP address and MAC address.

After changing the *UNET*'s IP address, change the Local Area Connection settings on the host computer back to its original setting.

**To change the Local Area Connection settings on the host computer:**

1. On the Windows 7 Taskbar, click the **Start**, type **ncpa.cpl**, then press **Enter**. (On Windows XP, click **Start | Run**, type **ncpa.cpl**, then click **OK**.)
2. Right-click **Local Area Connection**, then select **Properties**.

   The Local Area Connection Properties dialog box is displayed.

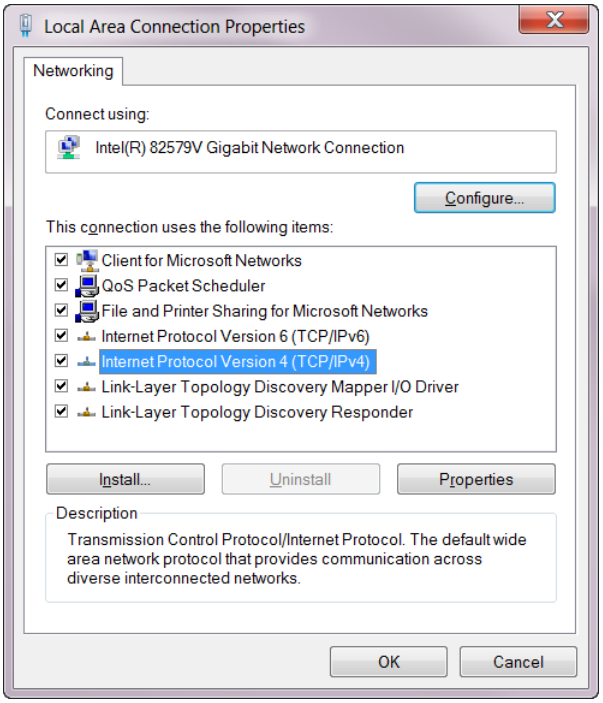



**Figure 2-4 Local Area Connection Properties Dialog Box**

3. Double-click **Internet Protocol Version 4 (TCP/IPv4)**.

The Internet Protocol Version 4 (TCP/IPv4) Properties dialog box is displayed.

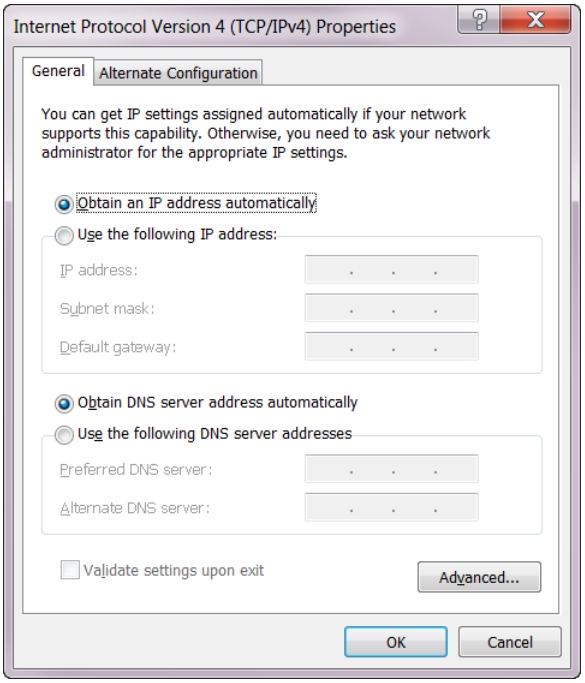



**Figure 2-5 Local Area Connection Properties Dialog Box**

**Note:** Make a note of the current settings in the Internet Protocol Version 4 (TCP/IPv4) Properties dialog box, so that you can return to these settings after connecting to the *UNET* and configuring its user-defined IP address.

4. Select **Use the following IP address**.
5. In the IP address field, type **10.72.63.x**, where **x** is any number from 1 to 255 except for 45, which is the IP address of the *UNET*.
6. In the Subnet mask field, type **255.255.255.0**.
7. Click **OK** in the Internet Protocol Version 4 (TCP/IPv4) Properties dialog box.
8. Click **OK** in the Local Area Connection Properties dialog box.

**Note:** There may be a short delay after changing the Local Area Connection's IP address, before you can connect with the *UNET*.

**To set the *UNET*'s Ethernet settings:**

1. Use Ethernet cables to connect the *UNET* to the host computer via an Ethernet switch as shown in **Figure 2-6**.

   In **Figure 2-6** the host computer is configured to be on the same network as the *UNET* (10.72.63.*), as described in previous procedure. (See page 2-20.) The last segment of the IP address of the host computer can be any value from 1 to 255 except for 45, which is the IP address of the *UNET*.

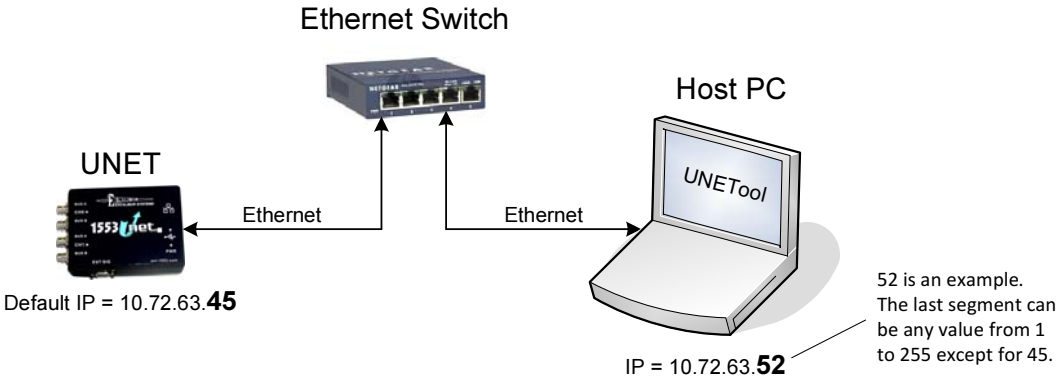
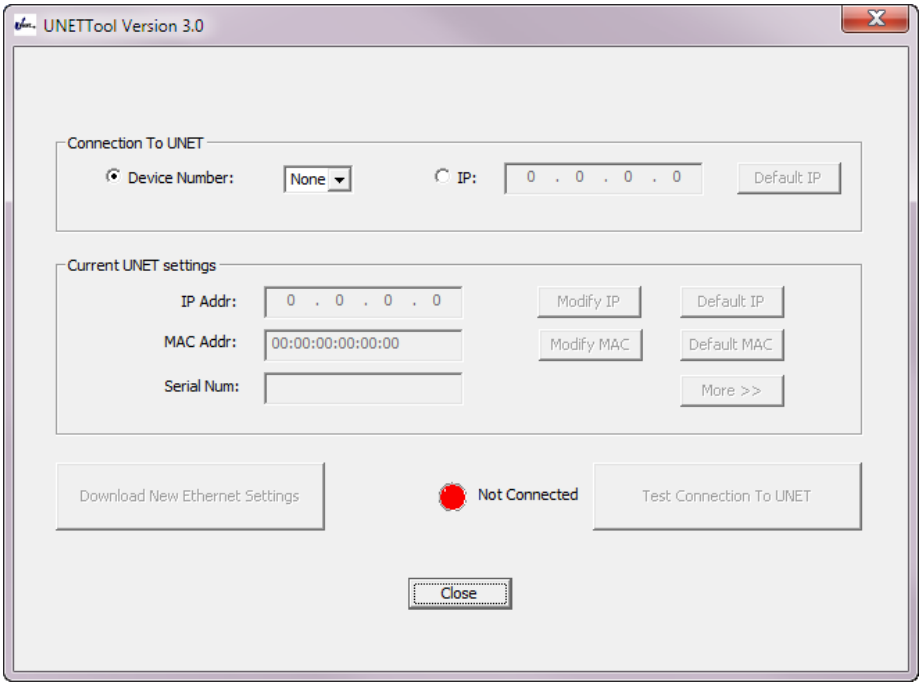



**Figure 2-6 Network Setup for Connecting with the *UNET*'s Default IP**

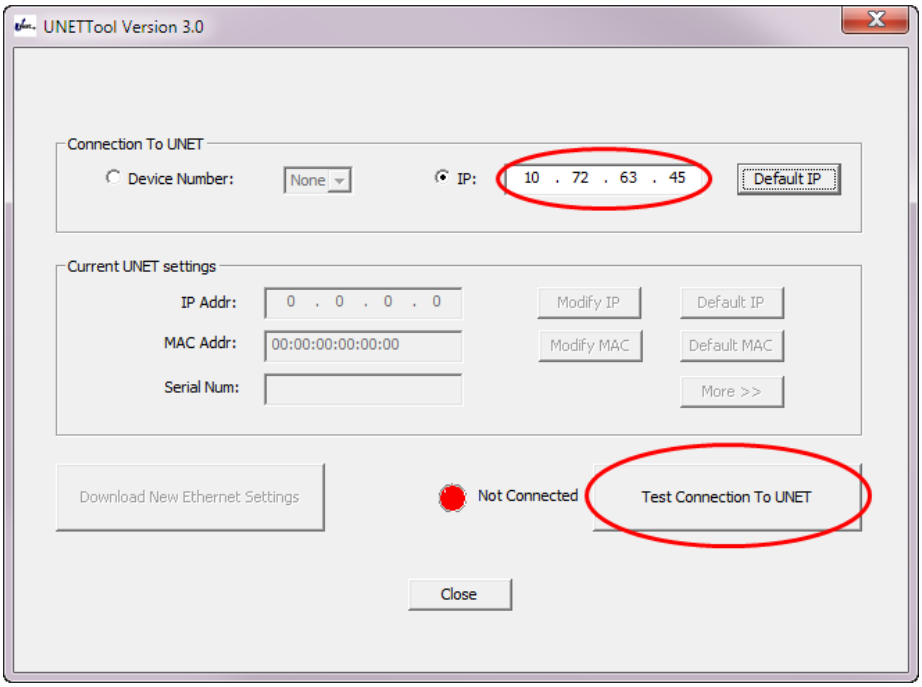
2. For power, do one of the following:
   - Connect the supplied power adapter to a wall outlet and its USB connector to the USB Power port (**PWR**) of the *UNET*.
   - Connect one side of the supplied USB cable to one of the computer's available USB ports and the other side to either the USB Power port (**PWR**) or the USB Communication port ⇽ of the *UNET*.
3. Turn off your computer's firewall. When using Windows 7 or later, make sure the firewall of both public and private networks is turned off.
4. Run the UNETTool: Click **Start | All Programs | Excalibur | 1553UNET | UNET Utilities | UNET GUI Tool**.

The UNETTool main screen is displayed.




5. In the Connection to UNET section of the screen, click the **IP** option button.
6. Click **Default IP**.

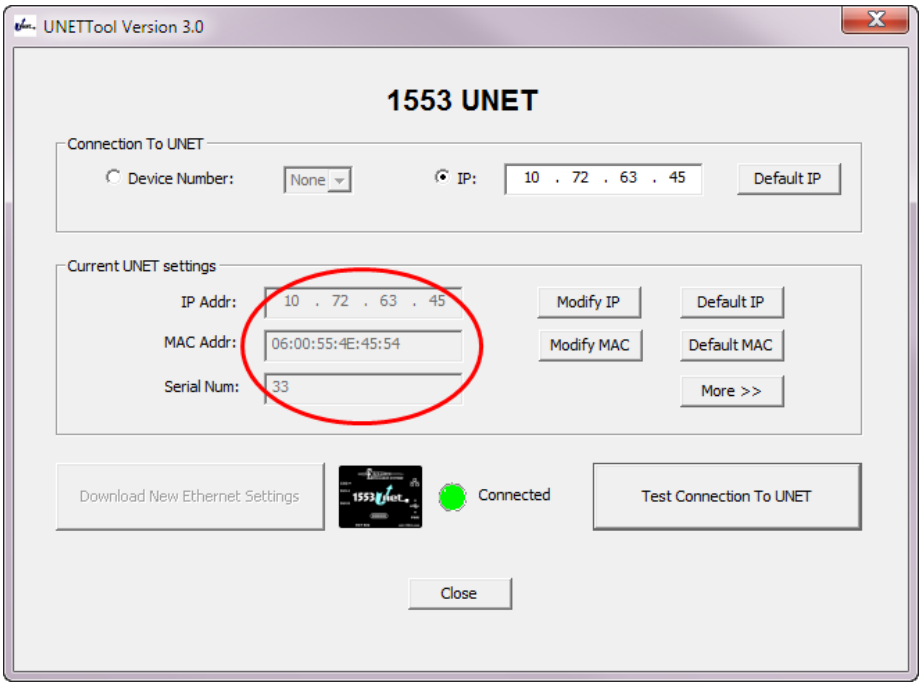
   The default IP of the *UNET* is displayed and the Test Connection to UNET button is enabled.




7. Click **Test Connection To UNET**.

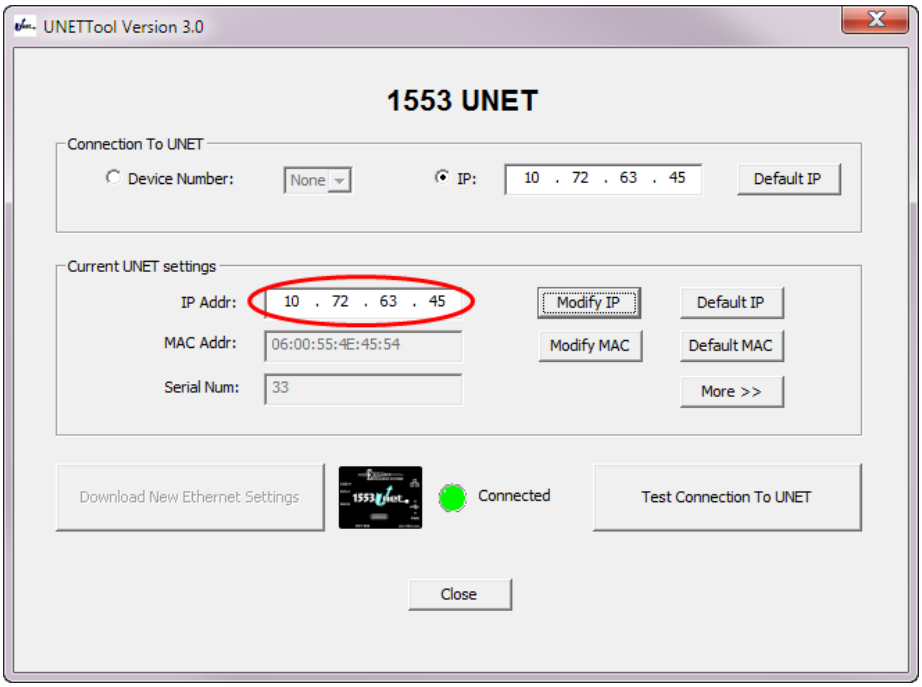
   The UNETTool attempts to connect to the *UNET* via Ethernet using the default IP address. When the UNETTool succeeds in connecting to the

*UNET*, the LED on the UNETTool turns green, and the *UNET*'s IP address, MAC address and serial number are displayed in the Current UNET settings section of the UNETTool main screen.



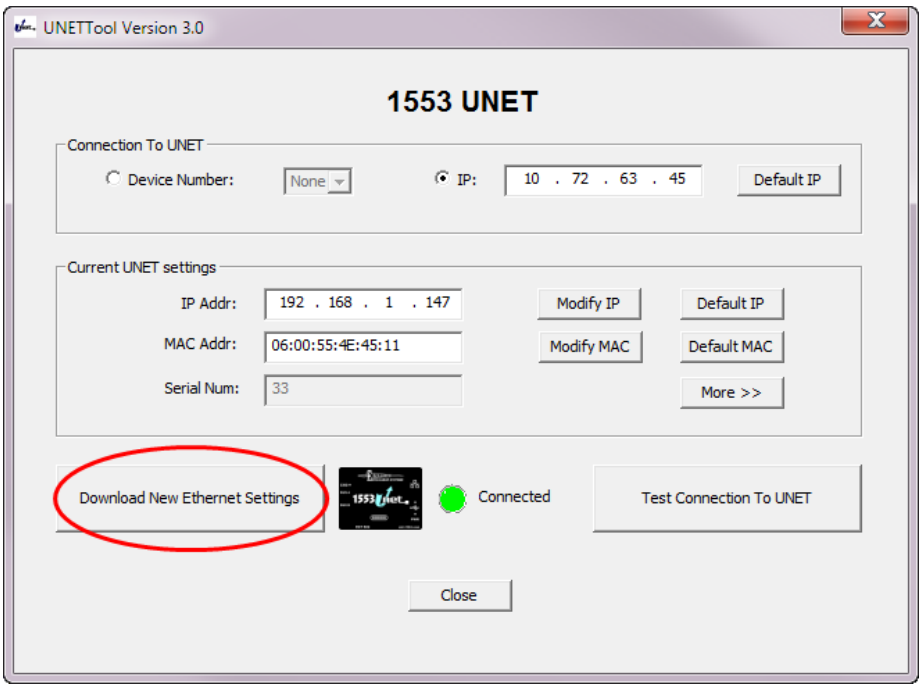

8. Click **Modify IP**.

   The IP Addr field becomes read/write.

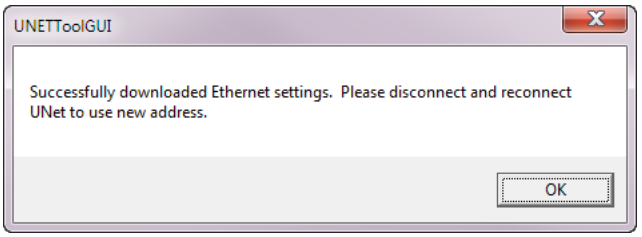



9. In the IP Addr field, type the desired user-defined IP address, for example, 192.168.1.147.
10. (Optional) To revert to the default IP address, click **Default IP**.

11. Click **Modify MAC**.
    The MAC Addr field becomes read/write.
12. In the MAC Addr field, type the desired user-defined MAC address.
    The Download New Ethernet Settings button is enabled.



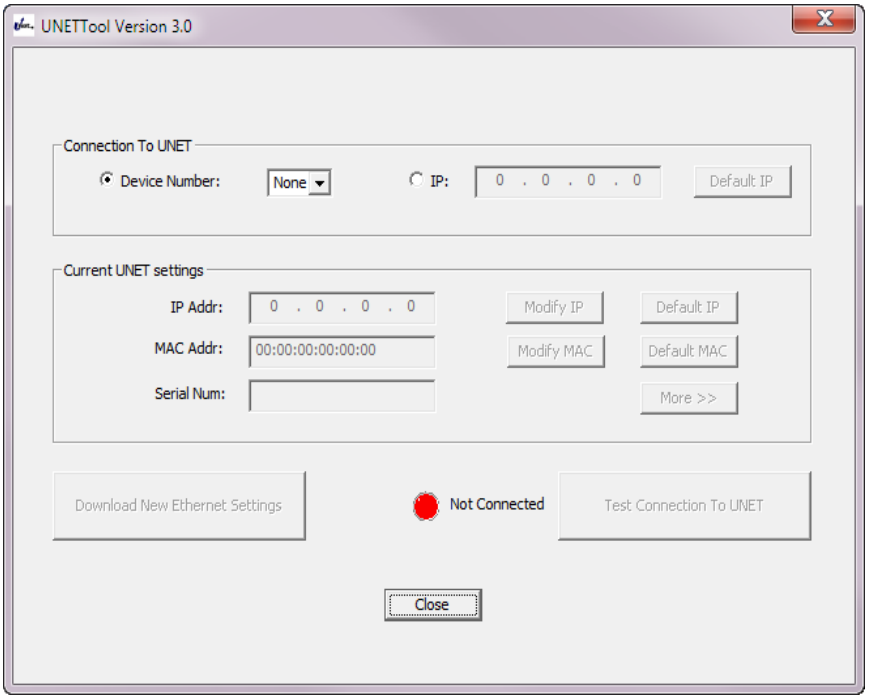

13. Click **Download New Ethernet Settings**.
    The following dialog box is displayed.



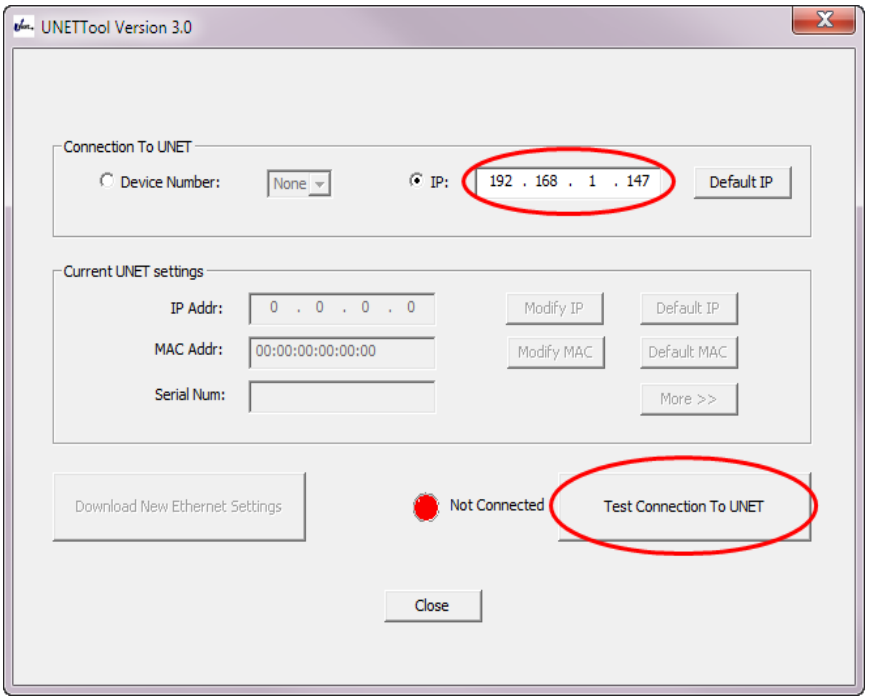

14. Click **OK**.
15. In order for the changes to take effect, restart the *UNET* as follows: Disconnect all USB cables from the *UNET* for several seconds, then reconnect them.
16. Click **Close** to exit the UNETTool.
17. Configure the host computer to have the IP address of the local network by undoing the steps performed in steps 4 and 6 of the previous procedure. See page 2-21.

18. Reopen the UNETTool: Click **Start | All Programs | Excalibur | 1553UNET | UNET Utilities | UNET GUI Tool**.

    The UNETTool main screen is displayed.




19. In the Connection to UNET section, click the **IP** option button, then type the new IP address of the *UNET*.




20. Click **Test Connection To UNET**.

    The UNETTool attempts to connect to the *UNET* using the new IP address. When the UNETTool succeeds in connecting to the *UNET*, the LED on the

UNETTool turns green, and the new IP address and MAC address are displayed in the Current UNET settings area of the UNETTool. This confirms that user-defined IP address was modified successfully.

21. Turn the firewall back on.
22. Add the *UNET*'s user-defined IP address as an exception to the firewall.
23. Test the connection to the *UNET* again (steps 19 and 20).
24. Click **Close**.

## 2.6 Testing the Connection Using the Device Number

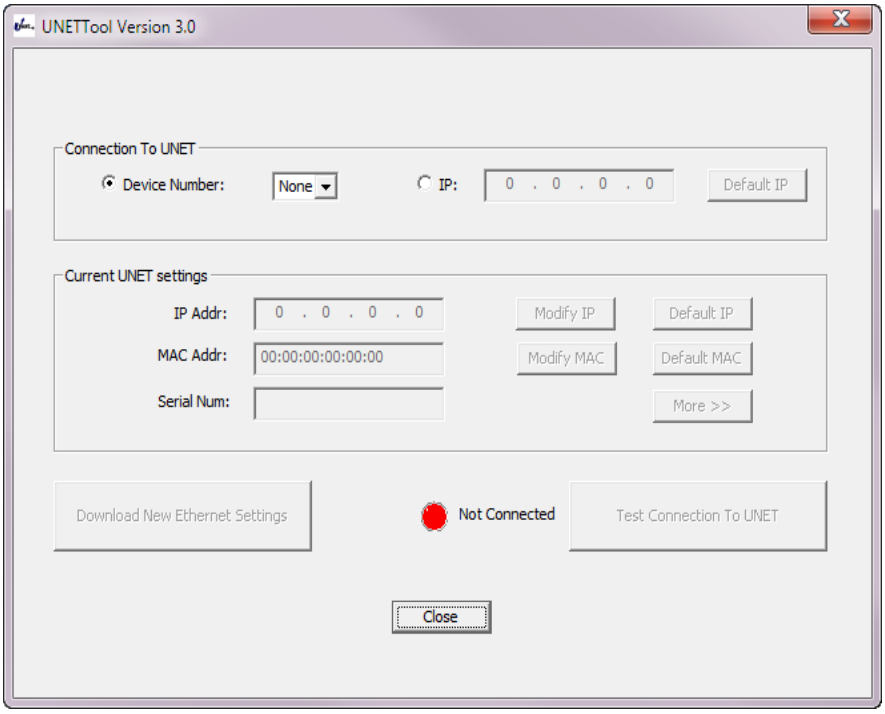
This section describes how to test the connection between the host and the *UNET* using the assigned device number that was assigned using the ExcConfig utility. (See **2.4 Assigning a Device Number on the Host Computer** on page 2-8.)

When testing the connection using the device number, the communication method used to connect to the *UNET* matches the communication method specified when defining the device number. If Device #1 was defined as using USB communication, and Device #2 was defined as using Ethernet communication, then when you select Device #1 the UNETTool will perform the test using USB communication, and when you select Device #2 the UNETTool will perform the test using Ethernet communication.

**To test the connection to the *UNET* using the device number:**

1. Make sure that the cables are connected as described in **2.2 Connecting the Cables** on page 2-2 and that a device number was assigned as described in **2.4 Assigning a Device Number on the Host Computer** on page 2-8.
2. Run the UNETTool: Click **Start | All Programs | Excalibur | 1553UNET | UNET Utilities | UNET GUI Tool**.

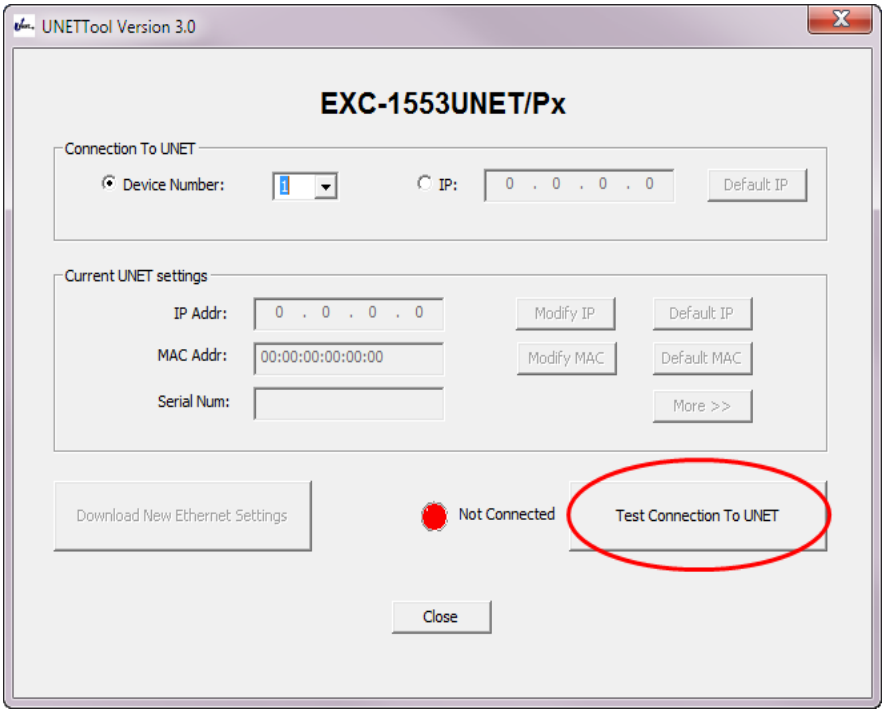
   The UNETTool main screen is displayed.




3. In Connection to UNET section, click the **Device Number** option button, then select the device number of the *UNET*, for example, 1.

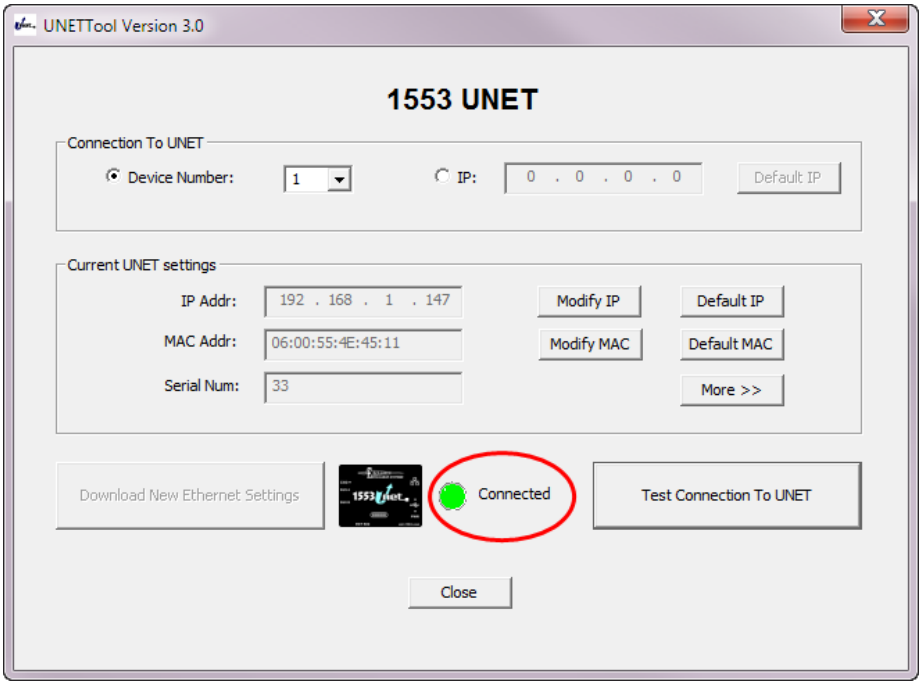
   The Test Connection To UNET button becomes available.

**Note:** The device number must match the device number that was assigned using the ExcConfig utility. See **2.4 Assigning a Device Number on the Host Computer** on page 2-8.




4. Click **Test Connection To UNET**.

   The UNETTool attempts to connect to the *UNET* using the selected device number. When the UNETTool succeeds in connecting to the *UNET*, the LED on the UNETTool turns green and the status changes to **Connected**.




5. Click **Close**.

## 2.7 Running a Test Program

*Excalibur Software Tools* include demo programs to verify that the *UNET* is operating properly.

*Excalibur Software Tools* are advanced API functions written in C language that enable you to write application and diagnostic programs. The functions are divided into two parts: module-level functions and board-level functions. The demo programs are also divided into module level and carrier board level demo programs.

The source code for the demo programs is also provided as a guide to develop your own applications using the *Software Tools*. The source code is located at:
**C:\Excalibur\1553UNET\demos_1553**
**C:\Excalibur\1553UNET\demos_4000**
**C:\Excalibur\1553UNET\demos_discrete**
(The above paths assume the default software installation location.)

Some of the demo programs are available on the Start menu at:
**Start | All Programs | Excalibur | 1553UNET | Demos (Visual Studio 2008)**

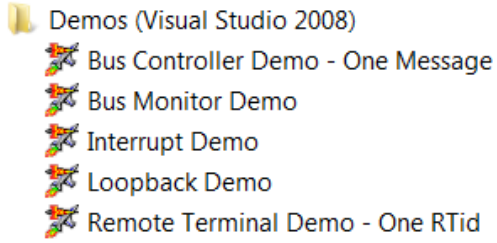



**Figure 2-7 Demo Programs on the Start Menu**

For more information on the functions used in the demo programs, see:

- *1553Px Family Software Tools Programmer's Reference*
- *M4KDiscrete Software Tools Programmer's Reference*
- *4000 Family Carrier Board Software Tools Programmer's Reference*

These manuals can be found at: **Start | All Programs | Excalibur | 1553UNET | Manuals.**

**Note:** These demo programs are console applications. To test the Discrete I/Os using a GUI application, use the Discrete Generator. See **2.8 Running the Discrete Generator** on page 2-31.

**To run a test program:**

1. Make sure that the cables are connected as described in **2.2 Connecting the Cables** on page 2-2.
2. Double-click (for example) **demo_int.exe** located in: **C:\Excalibur\1553UNET\demos_1553\bin.** (This path assumes the default software installation location.)

   *– or –*

   Click **Start | All Programs | Excalibur | 1553UNET | Demos (Visual Studio 2008) | Interrupt Demo.**

3. When prompted for a device number, use the device number defined using ExcConfig. See **2.4 Assigning a Device Number on the Host Computer** on page 2-8. This device number should also be used when developing your own applications.
4. When prompted for a module number, use 0 for a single channel *UNET*, or 0 or 1 for a two-channel *UNET*.

   When the system is set up properly, **demo_int.exe** shows the number of interrupts received by the host computer. **demo_int.exe** runs the *UNET*'s MIL-STD-1533 channels in BC mode, transmits messages, sends interrupts, and displays the number of interrupts received by the host computer.

**Note:** When prompted for a module number when running a demo program in **demos_4000**, use 4. When prompted for a module number when running a demo program in **demos_discrete**, use 2.

## 2.8 Running the Discrete Generator

The *Discrete Generator* is a Windows GUI program that enables you to configure the Discrete channels on your Discrete module, run the module and view the status of incoming Discretes. You can also configure and monitor interrupts based on status of each Discrete channel.

**To install the *Discrete Generator*:**

1. Click **Start | All Programs | Excalibur | 1553UNET | Install DiscreteGenerator**.

   The InstallShield Wizard is displayed.
2. Follow the on-screen instructions.

**To run the *Discrete Generator*:**

- Click **Start | All Programs | Excalibur | Discrete Generator | Discrete Generator Tool**.

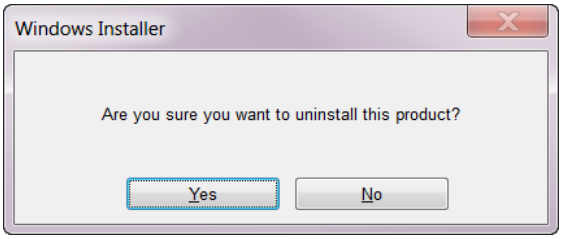
  The *Discrete Generator* main screen is displayed.

  For information on using the *Discrete Generator*, see the *Discrete Generator User's Manual* located at:
  **Start | All Programs | Excalibur | Discrete Generator | Discrete Generator Manual**.

## 2.9 Uninstalling the Software

This section describes how to uninstall the *UNET* software. This is required when upgrading to a newer software revision.

1. Click **Start | All Programs | Excalibur | 1553UNET | Uninstall UNET1553**.
   The following confirmation dialog box is displayed.




2. Click **Yes**.
   The *UNET* software is uninstalled.

# 3 Developing Applications

Chapter 3 describes how to port an application developed for other Excalibur MIL-STD-1553 and Discrete modules/channels to use with the *UNET*, and brief instructions for developing new applications for the *UNET* using *Excalibur Software Tools*.

*Excalibur Software Tools* are advanced API functions written in C language that enable you to write application and diagnostic programs.

These functions are divided into two parts: module-level functions for the MIL-STD-1553 and Discrete modules; and board-level functions. The *Software Tools* are fully described in the following manuals:

- *1553Px Family Software Tools Programmer's Reference*
- *M4KDiscrete Software Tools Programmer's Reference*
- *4000 Family Carrier Board Software Tools Programmer's Reference*

These manuals can be found at: **Start | All Programs | Excalibur | 1553UNET | Manuals**.

## 3.1 Porting an Existing Application to Work with the *UNET*

This section describes how to port an existing application to the *UNET* that was developed for an MIL-STD-1553 and Discrete module/channels on a different Excalibur board.

**To port your application to the *UNET*:**

1. In your original application environment, locate the currently used Excalibur DLLs of the *M4K1553Px* module/channel and the 4000 carrier board. If you are porting applications for Discrete, locate the DLL currently used for the Discrete module.
2. Rename the DLLS by adding a suffix such as **_ORIGINAL.dll**.
3. Run your application.

   You should receive an error that the Excalibur DLLs are missing.
4. If you receive an error that the Excalibur DLLs are missing, continue with Step 7.

   If you do not receive an error that the Excalibur DLLs are missing, and your application is running, than you have renamed DLLs (in Step 2) that are not being used by your application. In this case, close your application, look for the DLLs in **C:\Windows\System32**, rename them as in Step 2, and run your application.
5. If you receive an error that the Excalibur DLLs are missing, continue with Step 7.

   If you do not receive an error that the Excalibur DLLs are missing, and your application is running, use the Process Explorer utility to locate the DLLs used by your application, as follows:

   a. Double-click **procexp.exe** located in **C:\Excalibur\1553UNET\UNET utilities**.

      The Process Explorer License Agreement screen appears.

b. Click **Agree**.

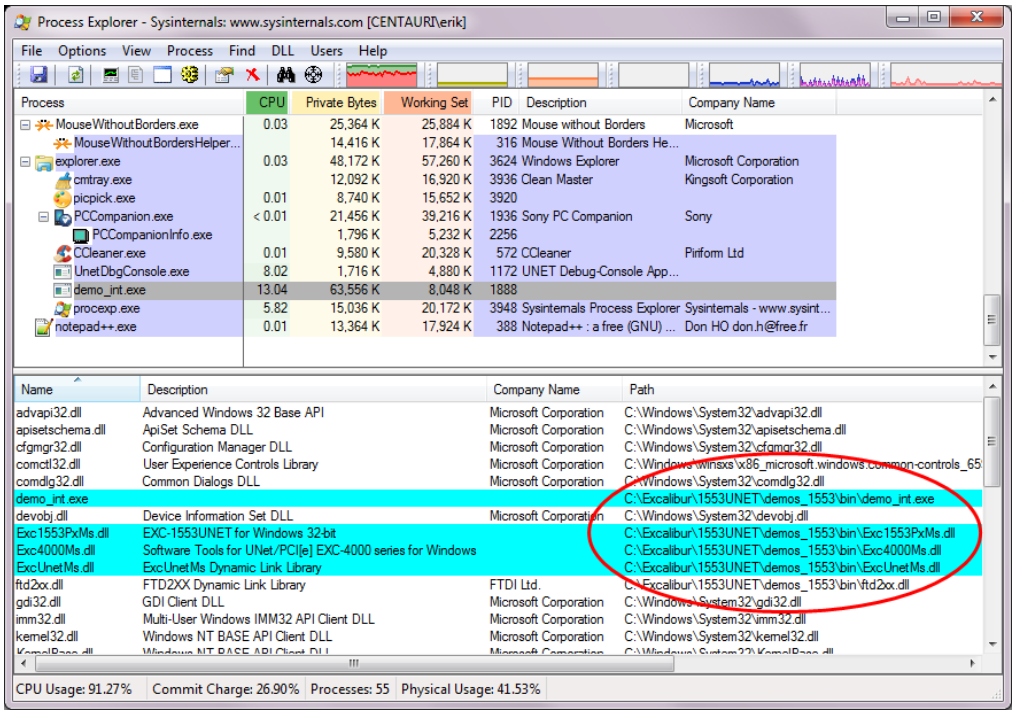
The Process Explorer main screen is displayed.

c. Make sure your application is still running.

d. Select your application in the list of processes.

e. Click the View DLLs icon on the toolbar.

The DLLs used by your application are displayed in the lower half of the window.




**Note:** If the View DLLs icon does not appear on the toolbar, and the DLLs are not displayed in the lower half of the screen, click the View Handles icon, then click the View DLLs icon.

6. Close your application, rename the DLLs as in Step 2, and run your application.

   You will receive an error that the Excalibur DLLs are missing.

7. Copy the relevant DLLs (for your modules and/or carrier board) from the appropriate subfolder in **C:\Excalibur\1553UNET\lib**, to the folder where your original DLL was located.

**Note:** For older applications, make sure the DLL names match the ones originally used by your application. If not, rename them to match the DLL names that were used in your application.

8. In addition to the DLLs for the modules and/or carrier board, some additional DLLs are required:
   - For Microsoft Visual Studio, copy **ExcUnetMs.dll** and **ftd2xx.dll** to the folder where your application is located.
   - For Borland, copy all the additional DLLs to the folder where your application is located.
9. Run your application.

   There should not be a missing DLL error, but there should be an error regarding initializing the card, such as **Init Failed**.
10. Run ExcConfig and set the device number of the *UNET* to match the device number used in your application. See **2.4 Assigning a Device Number on the Host Computer** on page 2-8.
11. Make sure you perform all other relevant installation steps in **Chapter 2: Installation and Setup** before running your application.
12. Run your application.

    It should run without any errors with the *UNET*.
13. If the application still does not run properly, copy the LIB files from the appropriate subfolder in **C:\Excalibur\1553UNET\lib** and recompile your application.

## 3.2 Developing New Applications for the *UNET*

**To develop new applications for the *UNET*:**

1. Add the **include** folder from **C:\Excalibur\1553UNET\include** to your project settings in your development environment.
2. Add the relevant **lib** folder for Microsoft Visual Studio (Debug or Release) or Borland, from **C:\Excalibur\1553UNET\lib** to your project settings in your development environment.
3. Add the relevant LIB file names (for your modules and/or carrier board) to your project.
4. Create your application.
5. Make sure you perform all relevant installation steps in **Chapter 2: Installation and Setup** before running your application.
6. Run your application.

**Note:** If you are using Microsoft Visual Studio, you can use the source code of a demo as programming examples. The demos are located in:
**C:\Excalibur\1553UNET\demos_1553**
**C:\Excalibur\1553UNET\demos_discrete**
**C:\Excalibur\1553UNET\demos_4000**

# 4 Mechanical and Electrical Specifications

Chapter 4 describes the mechanical and electrical specifications of the *UNET*.

## 4.1 Mechanical Outline of *EXC-1553UNET/Px*

The following are sample configurations. For all possible configurations, see **Chapter 12: Ordering Information.**

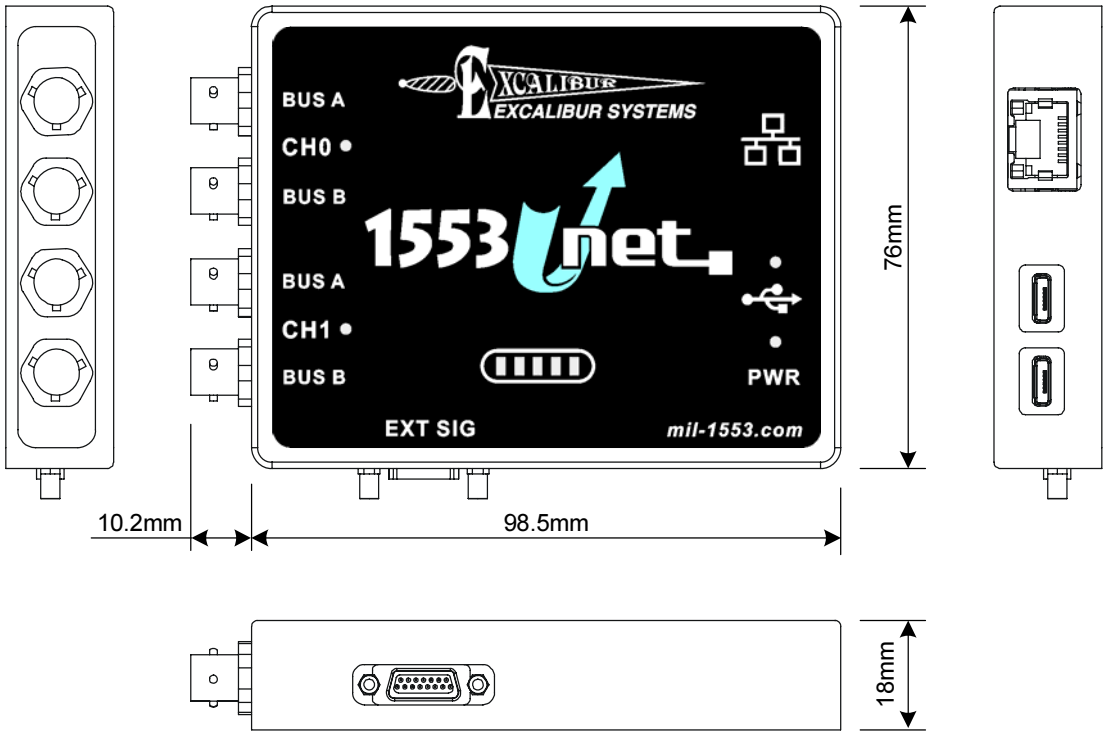



**Figure 4-1** ***EXC-1553UNET/P2-M*** **with Two MIL-STD-1553 Channels**

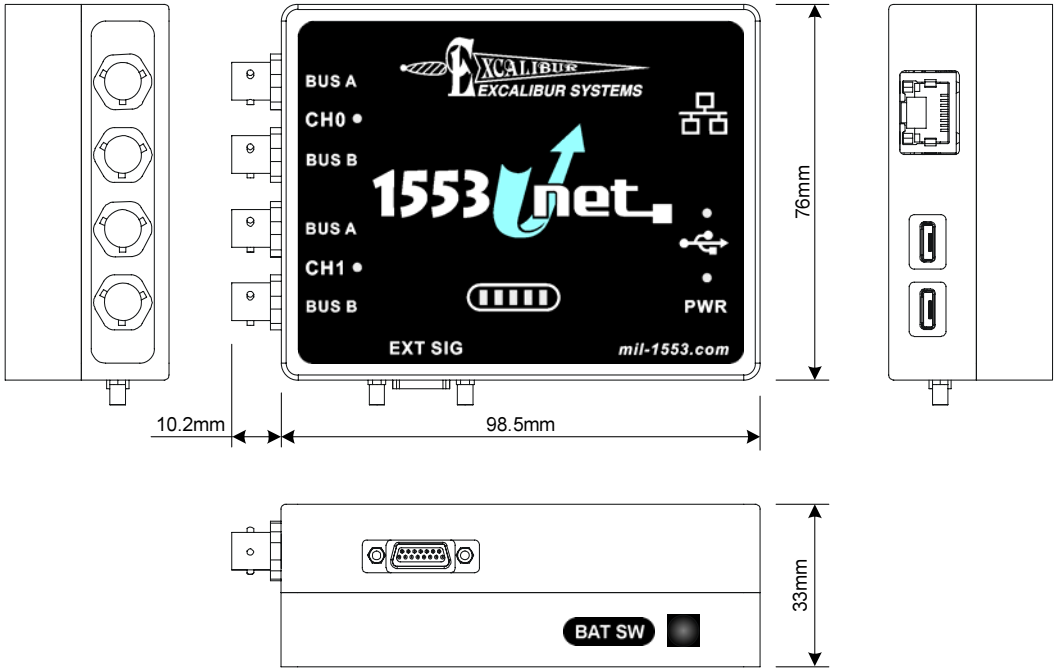



**Figure 4-2** ***EXC-1553UNET/P2-M-B*** **with Internal Battery**

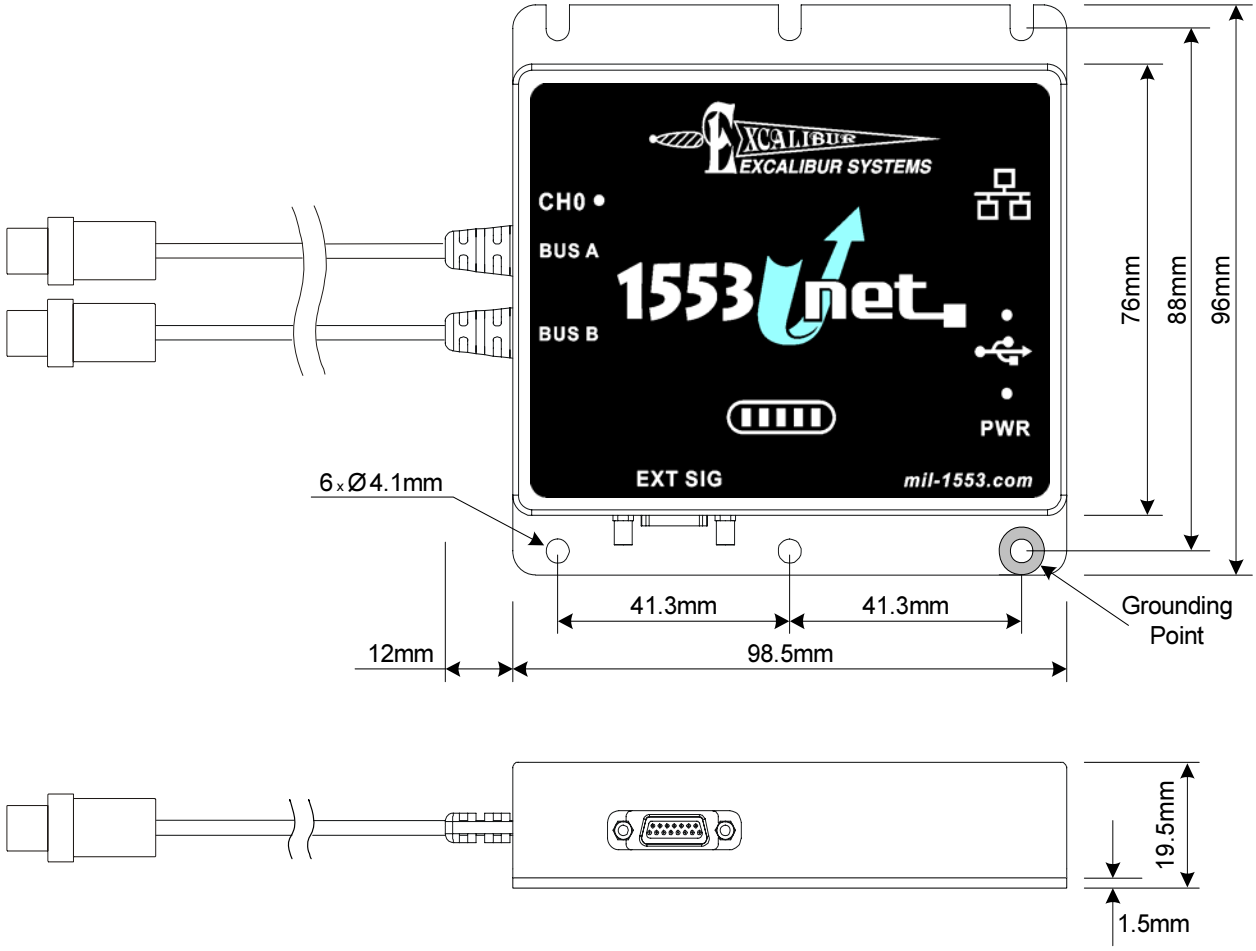



**Figure 4-3** ***EXC-1553UNET/P1-C-P*** **with One MIL-STD-1553 Channel and Hard-wired Cable and Mounting Plate**

## 4.2 Mechanical Outline of *ES-1553RUNET/Px*

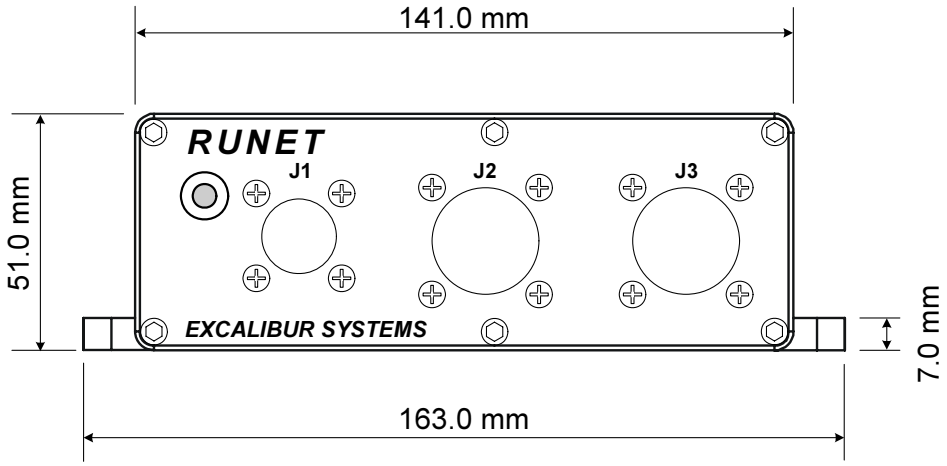



**Figure 4-4** ***ES-1553RUNET/Px*** **Front View**

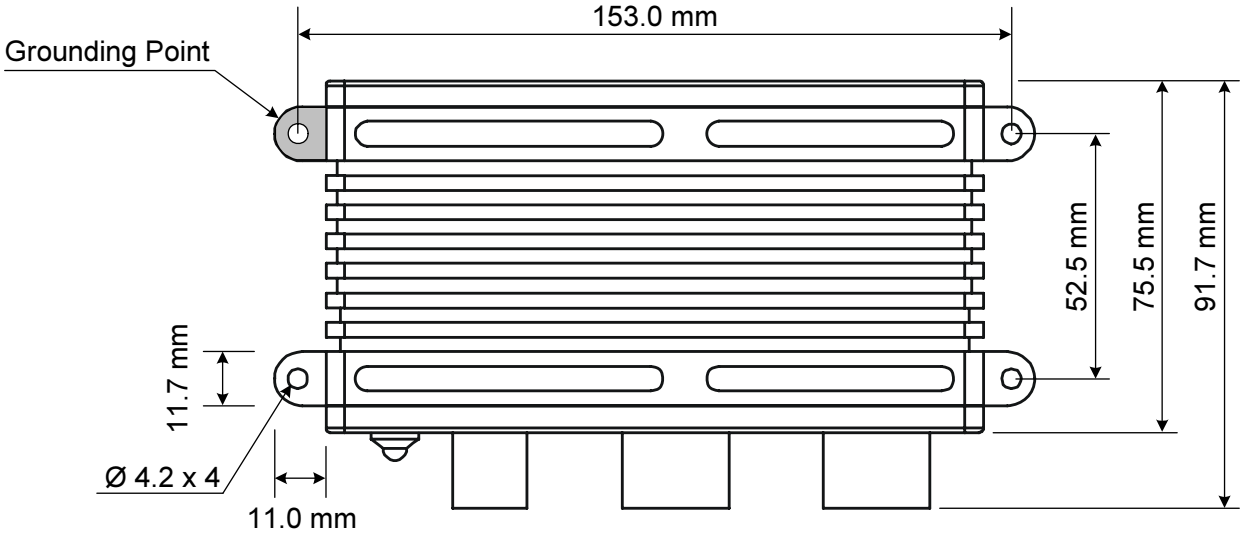



**Figure 4-5** ***ES-1553RUNET/Px*** **Top View**

## 4.3 LED Indicators of *EXC-1553UNET/Px*

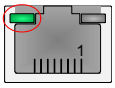
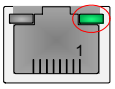
| LED | Color | State | Indication |
|---|---|---|---|
| **CH0** | Blue | Not lit | Power has not been supplied<br>- or -<br>MIL-STD-1553 channel 0 has failed to initialize |
| | | Solid | MIL-STD-1553 channel 0 is powered up and ready |
| | | Flashing | Activity on bus A or B of MIL-STD-1553 channel 0 |
| **CH1** | Blue | Not lit | Power has not been supplied<br>- or -<br>MIL-STD-1553 channel 1 has failed to initialize |
| | | Solid | MIL-STD-1553 channel 1 is powered up and ready |
| | | Flashing | Activity on bus A or B of MIL-STD-1553 channel 1 |
|  | Green | Not lit | No USB communication cable is connected<br>- or -<br>The USB communication cable is connected, but the host has not yet establish a connection with the *EXC-1553UNET/Px* (USB enumeration failed) |
| | | Solid | USB communication cable is connected, the *EXC-1553UNET/Px* has established a connection with the host, and the bus is active |
| **PWR** | Green | Not lit | No USB power cable is connected |
| | | Solid | USB power cable is connected and receiving power |
|  | Green | Not lit | No Ethernet cable connected |
| | | Solid | Ethernet cable connected, but no communication on bus |
| | | Flashing | Activity on Ethernet bus |
|  | Green | Not lit | No Ethernet cable connected |
| | | Solid | Ethernet cable connected and Ethernet bus is set to 100 Mbps |

**Table 4-1 LED Indicators of *EXC-1553UNET/Px***

Table 4-2 describes the functionality of the Battery LEDs. These LEDs only exist when the *EXC-1553UNET/Px* has an internal battery (-B configuration).

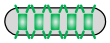
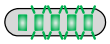
| LED | Color | State | Indication |
|---|---|---|---|
| | | **Battery LED States when Charging** | |
|  | Green | None lit | 0% charge in the internal battery |
|  | Green | All Battery LEDs flashing in sequence | Charging; 0% – 19% charge |
|  | Green | 1 LED solid; the rest of the LEDs flashing in sequence | Charging; 20% – 39% charge |
|  | Green | 2 LEDs solid; the rest of the LEDs flashing in sequence | Charging; 40% – 59% charge |
|  | Green | 3 LEDs solid; the rest of the LEDs flashing in sequence | Charging; 60% – 79% charge |
|  | Green | 4 LEDs solid; one flashing | Charging; 80% – 99% charge |
|  | Green | All LEDs solid | 100% charge |
| | Green | **Battery LED States when Discharging** | |
|  | Green | All LEDs solid | Discharging; 100% – 91% charge in the internal battery |
|  | Green | 4 LEDs solid; 1 LED flashing | Discharging; 90% – 81% charge |
|  | Green | 4 LEDs solid | Discharging; 80% – 71% charge |
|  | Green | 3 LEDs solid; 1 LED flashing | Discharging; 70% – 61% charge |
|  | Green | 3 LEDs solid | Discharging; 60% – 51% charge |
|  | Green | 2 LEDs solid; 1 LED flashing | Discharging; 50% – 41% charge |
|  | Green | 2 LEDs solid | Discharging; 40% – 31% charge |
|  | Green | 1 LED solid; 1 LED flashing | Discharging; 30% – 21% charge |
|  | Green | 1 LED solid | Discharging; 20% – 11% charge |
|  | Green | 1 LED flashing | Discharging; 10% – 5% charge |
|  | Green | 3 LEDs flashing in an alternating pattern | Discharging; 4% – 0% charge |
|  | Green | No LEDs lit | 0% charge |

**Table 4-2 Battery LED Indicators of *EXC-1553UNET/Px***

## 4.4 LED Indicators of *ES-1553RUNET/Px*

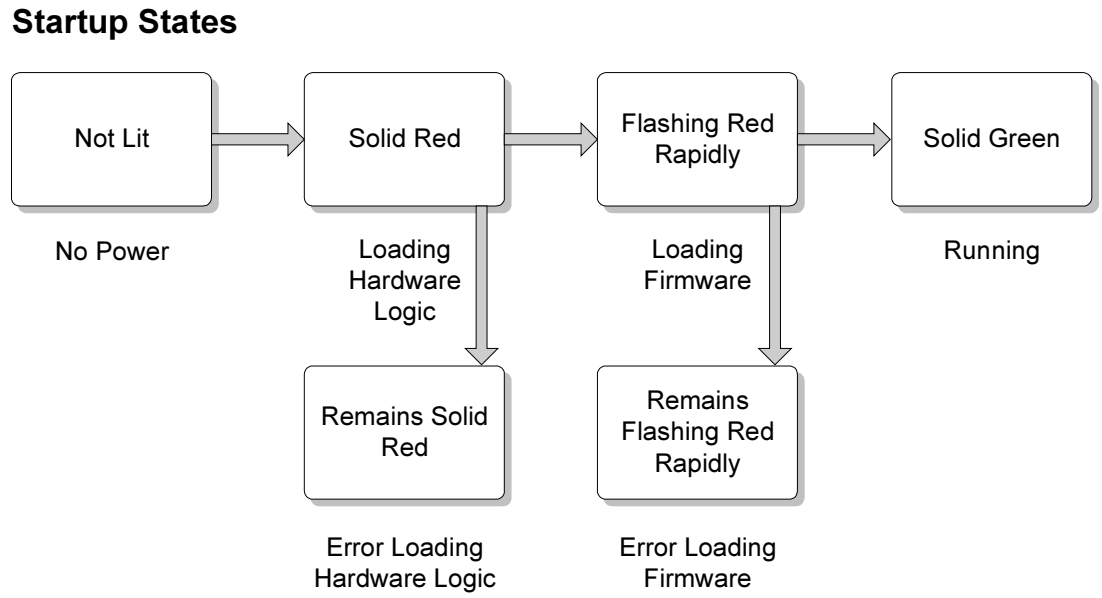



**Figure 4-6** ***ES-1553RUNET/Px*** **Startup States**

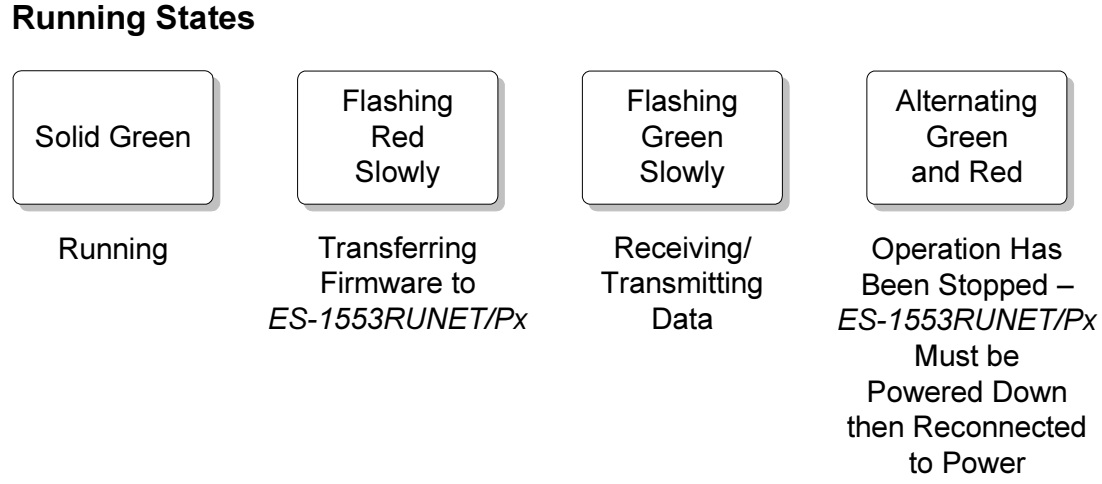



**Figure 4-7** ***ES-1553RUNET/Px*** **Running States**

## 4.5 Connectors of *EXC-1553UNET/Px*

The following sections describe the connectors of the *EXC-1553UNET/Px*. For information on connecting the cables, see **2.2 Connecting the Cables** on page 2-2.

***Caution:*** **On the EXC-1553UNET/Px, make sure there is no I/O communication while disconnecting any of the cables. Connecting or disconnecting the cables during communication can seriously damage the EXC-1553UNET/Px.**

***Caution:*** **On the ES-1553RUNET/Px, make sure the power is OFF while connecting or disconnecting communication cables to or from the device. Connecting or disconnecting cables to or from the ES-1553RUNET/Px while the power is ON can seriously damage the ES-1553RUNET/Px or a system connected to the ES-1553RUNET/Px.**

### 4.5.1 MIL-STD-1553 Twinax Connectors

The *EXC-1553UNET/Px-C* has hard-wired MIL-STD-1553 cables. Each cable ends with a female Trompeter CJ70 connector (or equivalent). The cables are marked Bus A and Bus B (in addition to the channel number). The twinax cables mate, for example, with Trompeter PL75 male connectors. These connectors are not supplied by Excalibur.

The *EXC-1553UNET/Px-M* has female Trompeter BJ157 connectors (or equivalent) mounted to the front of the unit. These connectors mate, for example, with Trompeter PL155 male connectors. For this configuration, Excalibur supplies 30 cm (11.8 in.) adapter cables that have Trompeter PL155 (or equivalent) connectors on one side and Trompeter CJ70 (or equivalent) on the other side.

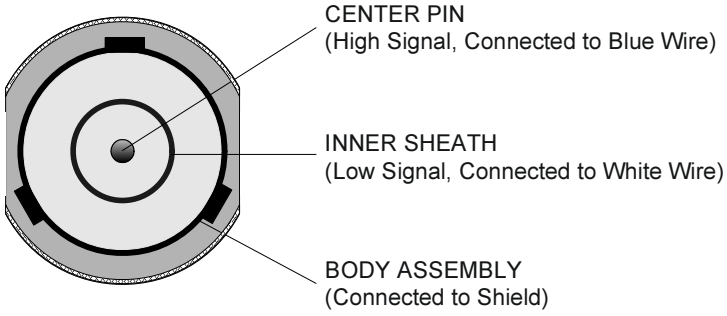



**Figure 4-8 Twinax Connector on *EXC-1553UNET/Px-C* – Front View**

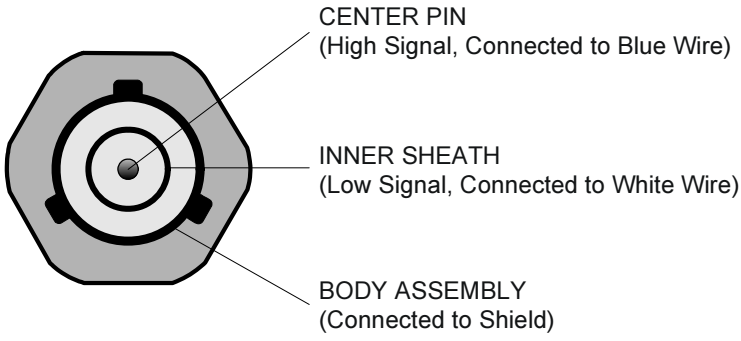



**Figure 4-9 Twinax Connector on *EXC-1553UNET/Px-M* – Front View**

### 4.5.2 USB Communication Connector

The USB Communication Connector is a standard 5-pin, Micro-B USB connector. A 1-meter Micro-B to Standard-A cables is supplied.

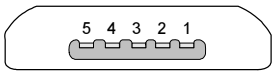



**Figure 4-10 USB Communication Connector – Front View**

| Pin # | Signal Name | Description |
|---|---|---|
| **1** | VBUS | +5V power supply from USB bus |
| **2** | DATA- | USB DATA- signal |
| **3** | DATA+ | USB DATA+ signal |
| **4** | N\C | Not connected |
| **5** | GND | Ground |

**Table 4-3 USB Communication Connector Pinouts**

### 4.5.3 USB Power Connector

The USB Power Connector is a standard 5-pin, Micro-B USB connector. A 1-meter Micro-B to Standard-A cables is supplied.

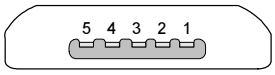



**Figure 4-11 USB Power Connector – Front View**

| Pin # | Signal Name | Description |
|---|---|---|
| **1** | VBUS | +5V power supply from USB bus |
| **2** | PwCon- | USB power configuration signal |
| **3** | PwCon+ | USB power configuration signal |
| **4** | N\C | Not connected |
| **5** | GND | Ground |

**Table 4-4 USB Communication Connector Pinouts**

### 4.5.4 Ethernet Connector

The Ethernet Connector is a standard RJ45, 8-pin, Ethernet connector.

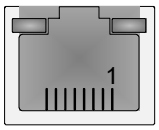



**Figure 4-12 Ethernet Connector – Front View**

| Pin # | Signal Name | Description |
|---|---|---|
| **1** | TX+ | Ethernet Transmit Data+ |
| **2** | TX- | Ethernet Transmit Data- |
| **3** | RX+ | Ethernet Receive Data+ |
| **4** | N/C | Not Connected |
| **5** | N/C | Not Connected |
| **6** | RX- | Ethernet Receive Data- |
| **7** | N/C | Not Connected |
| **8** | N/C | Not Connected |

**Table 4-5 Ethernet Connector Pinouts**

### 4.5.5 External Signals Connector Pinouts

The External Signals Connector is a 1.27mm pitch micro D, 15-pin connector, P/N: Molex 83612-9020.

The mating connector, P/N: Molex 83422-9014, and a socket crimp terminal, P/N: Molex 83000-0083, are supplied by Excalibur.

A custom adapter cable for the External Signals Connector can be ordered from Excalibur. Contact Excalibur Sales. See **1.8 Technical Support** on page 1-12.

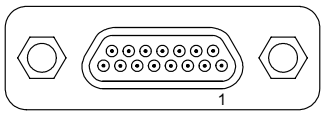



**Figure 4-13 External Signals Connector – Front View**

| Pin # | Signal Name | Description |
|---|---|---|
| **1** | DIO0 | Discrete Channel 0 connection[1] |
| **2** | DIO2 | Discrete Channel 2 connection[1] |
| **3** | GND | Ground connection for Discrete channels |
| **4** | DIO5 | Discrete Channel 5 connection[1] |
| **5** | DIO7 | Discrete Channel 7 connection[1] |
| **6** | EXTTCLKI | External Time Tag Clock Input (nominal value: 1MHz). This signal supplies an external global clock for the Time Tags of all the channels. Use the signal to synchronize the Time Tags that are implemented on the channels[2] to other boards or systems.[3] See **Time Tag Clock Select Register** on page 5-4. |
| **7** | EXTTRSTn | External Time Tag Reset TTL Input. Use this low active pulsed signal (minimum 100 nsec.wide) to simultaneously reset the Time Tags of all the channels from an external source. Use the signal to synchronize these Time Tags to other boards or systems.[3] |
| **8** | GND | Provides ground reference for the digital signal connections. |
| **9** | DIO1 | Discrete Channel 1 connection[1] |
| **10** | DIO3 | Discrete Channel 3 connection[1] |
| **11** | DIO4 | Discrete Channel 4 connection[1] |
| **12** | DIO6 | Discrete Channel 6 connection[1] |
| **13** | EXSTARTn | External Start LVTTL Input. Provides an option to start the MIL-STD-1553 channel externally by applying a negative pulse with respect to the GND pin, with a minimum width of 100 nsec. Before applying the pulse, the channel should be fully set up in the required mode, except for Bit 0 of the Start register, which should be left at 0. To stop the selected operation, follow the normal procedure described under the Start register. See **Software Reset Register** on page 5-3 |
| **14** | IRIG B | IRIG B120 Input. This should be a 1KHz sine wave, amplitude modulated, IRIG B signal with a 3:1 modulation ratio at 3V typical. |
| **15** | EXTTRSOn | Global Time Tag Reset TTL Output. This low active signal is activated each time a Global Time Tag Reset is applied. Use the signal to synchronize other boards or systems to the Time Tags that are implemented on the channels.[2] This signal is activated by either an internal Global Time Tag Reset (see **Software Reset Register** on page 5-3) or by an External Time Tag Reset signal (EXTTRSTn).[3] |

**Table 4-6 External Signals Connector Pinouts**

1. See **1.7 Discrete Channel Information,** on page 1-11.
2. See >> and **Time Tag Hi & Lo** on page 11-14 for a description of how the Time Tag clock is implemented for each channel.
3. See **4.5.5.1 Synchronizing with an External Source** on page 4-12 and **4.5.5.2 Synchronizing Between EXC-1553UNET/Px Devices** on page 4-13.

#### 4.5.5.1 Synchronizing with an External Source

To synchronize a single *EXC-1553UNET/Px* to an external system, the external clock source and the external reset must be connected to the EXTTCLKI and the EXTTRSTn signals respectively.

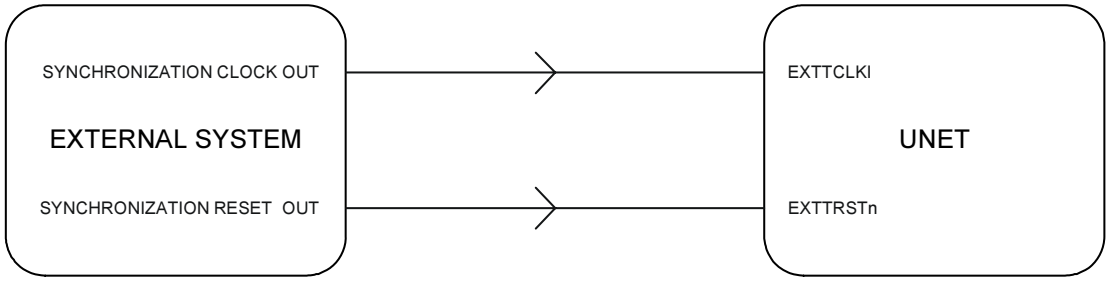



**Figure 4-14 Synchronization of a Single *EXC-1553UNET/Px* Board to an External System**

To synchronize an external system to a single *EXC-1553UNET/Px*, the EXTTRSOn signal must be connected to the external reset of the external system.

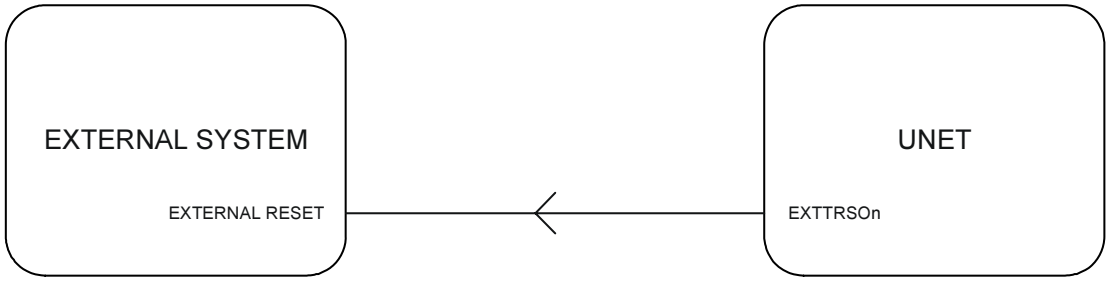



**Figure 4-15 Synchronization of an External System to a Single *EXC-1553UNET/Px* Board**

**Note:** The synchronization clock and reset signals may be connected to multiple targets to achieve system wide synchronization.

#### 4.5.5.2 Synchronizing Between *EXC-1553UNET/Px* Devices

To synchronize multiple *EXC-1553UNET/Px* devices the **EXTTRSOn** signal of one board must be connected to all the **EXTTRSTn** signals of the remaining devices.

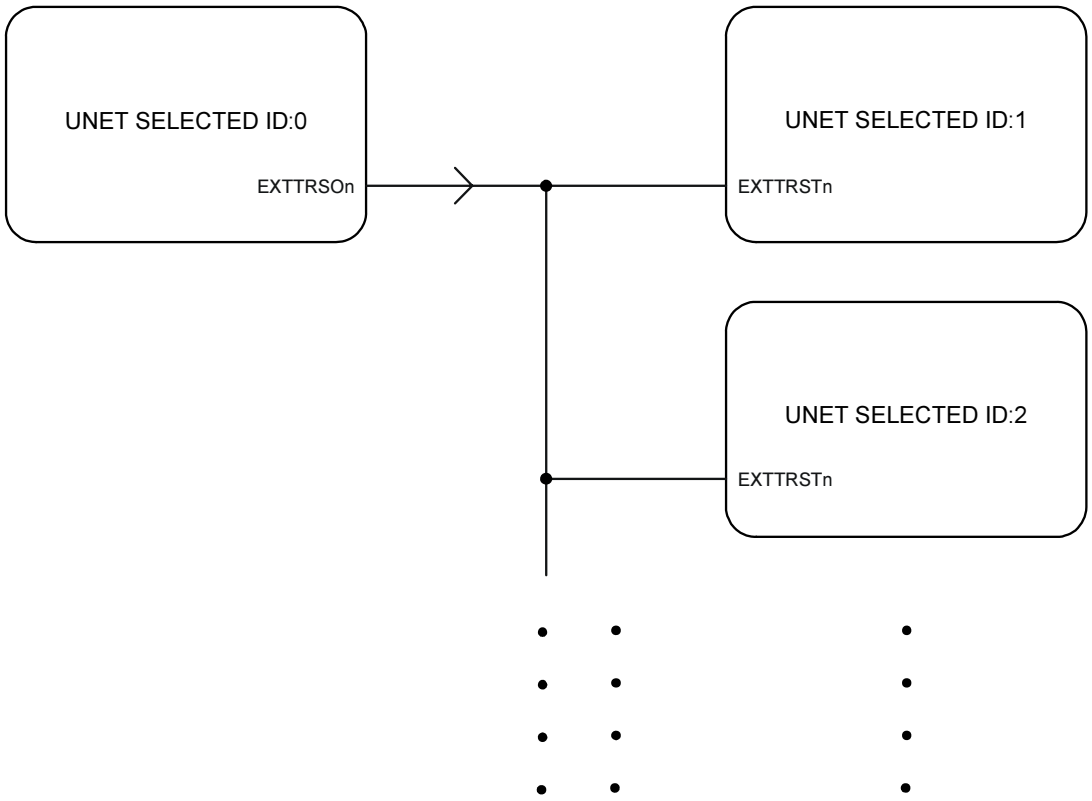



**Figure 4-16 Synchronization Between *EXC-1553UNET/Px* Devices**

## 4.6 Connectors of *ES-1553RUNET/Px*

### 4.6.1 Power Connector [J1]

Connector J1 is a 3-pin, male, MIL-DTL-38999, Series III connector, P/N: Amphenol 88-628751-98P.

The mating connector is a 3-pin, straight plug female, MIL-DTL-38999, Series III connector, P/N: Amphenol TV06RW-9-98S.

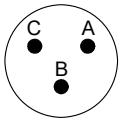



**Figure 4-17 Power Connector [J1] – Front View**

| J1 Pin # | Description |
|---|---|
| **A** | Power input (11-36 VDC) |
| **B** | Reserved |
| **C** | Power return |

**Table 4-7 Connector J1 Pinouts**

### 4.6.2 Communications I/O Connector [J2]

Connector J2 is a 22-pin, female, MIL-DTL-38999, Series III connector, P/N: Amphenol 88-628753-35S.

The mating connector is a 22-pin, straight plug male, MIL-DTL-38999, Series III connector, P/N: Amphenol TV06RW-13-35P.

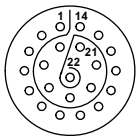



**Figure 4-18 Communications I/O Connector [J2] – Front View**

Connector J2 provides the following functionality:

- 1 MIL-STD-1553 channel
- 8 Discrete channels

| J2 Pin # | Signal Name | Description |
|---|---|---|
| **1** | BUSAL | MIL-STD-1553 Bus A low line connection |
| **2** | BUSAH | MIL-STD-1553 Bus A high line connection |
| **3** | N/C | Not Connected |
| **4** | N/C | Not Connected |
| **5** | N/C | Not Connected |
| **6** | N/C | Not Connected |
| **7** | N/C | Not Connected |
| **8** | DSIO4 | Discrete Channel 4 connection[1] |
| **9** | DSIO6 | Discrete Channel 6 connection[1] |
| **10** | DSIO7 | Discrete Channel 7 connection[1] |
| **11** | DSIO5 | Discrete Channel 5 connection[1] |
| **12** | BUSBL | MIL-STD-1553 Bus B low line connection |
| **13** | BUSBH | MIL-STD-1553 Bus B high line connection |
| **14** | SHIELD | Shield connection provided for MIL-STD-1553 cables |
| **15** | N/C | Not Connected |
| **16** | N/C | Not Connected |
| **17** | DSIO0 | Discrete Channel 0 connection[1] |
| **18** | DSIO2 | Discrete Channel 2 connection[1] |
| **19** | GND | Ground connection for Discrete channels 4–7 |
| **20** | DSIO3 | Discrete Channel 3 connection[1] |
| **21** | DSIO1 | Discrete Channel 1 connection[1] |
| **22** | GND | Ground connection for Discrete channels 0–3 |

**Table 4-8 Connector J2 Pinouts**

1. See **1.7 Discrete Channel Information,** on page 1-11.

### 4.6.3 Communications I/O Connector [J3]

Connector J3 is a 22-pin, female, MIL-DTL-38999, Series III connector, P/N: Amphenol 88-628753-35H.

The mating connector is a 22-pin, straight plug male, MIL-DTL-38999, Series III connector, P/N: Amphenol TV06RW-13-35PA.

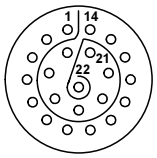



**Figure 4-19 Communications I/O Connector [J3] – Front View**

Connector J3 provides the following functionality:

- MIL-STD-1553 RT address
- IRIG B
- 1 Ethernet channel

| J3 Pin # | Signal Name | Description |
|---|---|---|
| **1** | N/C | Not Connected |
| **2** | ETHRx_N | Ethernet receive low line connection |
| **3** | ETHRx_P | Ethernet receive high line connection |
| **4** | ETHTx_N | Ethernet transmit low line connection |
| **5** | ETHTx_P | Ethernet transmit high line connection |
| **6** | RTA0 | MIL-STD-1553 RT address bit position 0 input[1,3] |
| **7** | RTA2 | MIL-STD-1553 RT address bit position 2 input[1,3] |
| **8** | RTA3 | MIL-STD-1553 RT address bit position 3 input[1,3] |
| **9** | RTAP | MIL-STD-1553 RT address odd parity bit input[1,3] |
| **10** | RTALn | MIL-STD-1553 RT address number lock input[2,3] |
| **11** | N/C | Not Connected |
| **12** | N/C | Not Connected |
| **13** | N/C | Not Connected |
| **14** | N/C | Not Connected |
| **15** | IRIGB | IRIG B input[4] |
| **16** | SHIELD | Shield connection provided for Ethernet cables |
| **17** | RTA1 | MIL-STD-1553 RT address bit position 1 input[1,3] |
| **18** | GND | Provided for MIL-STD-1553 RT address pins that need to be set to ‘0’ |
| **19** | N/C | Not Connected |
| **20** | N/C | Not Connected |
| **21** | N/C | Not Connected |
| **22** | RTA4 | MIL-STD-1553 RT address bit position 4 input[1,3] |

**Table 4-9 Connector J3 Pinouts**

1. Shorted to ground = logic 0
   Open = logic 1
2. Shorted to ground = MIL-STD-1553 RT address locked
   Open = MIL-STD-1553 RT address unlocked
3. This bit is only for a single channel *EXC-1553UNET/P1S.* When the MIL-STD-1553 RT address number is locked, the RT address number is set to the value represented by the RTA0 – RTA4 and RTAP pins. Ensure that the chosen RT address number is the only RT address number used in the conversion tables. When the RT address number is unlocked, the RT address can be set by writing the RT address number in the conversion tables.
4. IRIG B120 Input. This should be a 1KHz sine wave, amplitude modulated, IRIG B signal with a 3:1 modulation ratio at 3V typical.

## 4.7 Power Requirements of *EXC-1553UNET/Px*

The maximum power requirements of the *EXC-1553UNET/Px* @5V are:

| Number of Channels | 0% Duty Cycle (Standby) | 100% Duty Cycle |
|---|---|---|
| **P1 (one channel)** | 450 mA | 650 mA |
| **P2 (two channels)** | 450 mA | 850 mA |

**Table 4-10 *EXC-1553UNET/Px* Power Requirements**

**Note:**

- Use only the USB Power Supply provided by Excalibur Systems. Excalibur Systems is not responsible for any damage or lack of functionality when a different power supply is used.
- In most cases (depending on the load on MIL-STD-1533 bus and the capabilities of the computer), the power supplied through the USB Communication port will be enough to operate the *EXC-1553UNET/Px*. If you require more power, the USB Power port (**PWR**) should be connected to another USB port of the computer or to a wall outlet using the 5V USB Power Supply provided by Excalibur Systems. Each of these three power sources (USB Communication port to PC USB, USB Power port (**PWR**) to PC USB or USB Power port (**PWR**) to wall outlet) can be used separately as the prime power source.

### 4.7.1 Battery Option Power Information

The *EXC-1553UNET/Px*'s internal rechargeable battery can be recharged via one of the *EXC-1553UNET/Px*'s USB ports or both. While charging the internal battery of the *EXC-1553UNET/Px*, the *EXC-1553UNET/Px* is fully operational. The time to fully charge the internal battery is dependent on the amount of power required to operate the *EXC-1553UNET/Px*. The charging time lengthens with the increase of data being transmitted.

When the *EXC-1553UNET/Px* is configured with an internal battery, there is a switch located on the side of the battery casing that allows you to save battery power when the *EXC-1553UNET/Px* is not in use. When any form of external power source is connected, the *EXC-1553UNET/Px* operates and charges the internal battery as normal, and this switch has no effect. When the external

power supply is removed, and the *EXC-1553UNET/Px* operates solely from the internal battery, the switch is used to turn the *EXC-1553UNET/Px* ON or OFF. A short press on the battery switch turns ON the *EXC-1553UNET/Px*. When the *EXC-1553UNET/Px* operates solely from internal battery, pressing the battery switch for approximately five seconds turns the *EXC-1553UNET/Px* off, saving the battery charge.

When operating solely from the battery, the *EXC-1553UNET/Px* can:

- Transmit a single channel with a 100% duty cycle on the bus for approximately 6 hours.
- Transmit a two channels with a 100% duty cycle on the bus for approximately three hours.

The amount of power and transmission time that the *EXC-1553UNET/Px* can provide when running solely from the internal battery is influenced by the environmental temperature, the age of the battery and the number of charging cycles. Like all rechargeable batteries, the amount of power it can provide is reduced over time, and in lower temperature environments.

## 4.8 Power Requirements of *ES-1553RUNET/Px*

The *ES-1553RUNET/Px* draws a maximum of 5 Watts of power at 11-36 VDC.

***Caution:*** **On the ES-1553RUNET/Px, make sure the power is OFF while connecting or disconnecting communication cables to or from the device. Connecting or disconnecting cables to or from the ES-1553RUNET/Px while the power is ON can seriously damage the ES-1553RUNET/Px or a system connected to the ES-1553RUNET/Px.**

# 5 Host Interface Global Registers (Advanced)

Chapter 5 describes the *UNET*'s Host Interface Global registers. This chapter is one of several chapters that describe how to operate the *UNET* via the memory registers, which enables you to develop your own software tools. When using Excalibur's software tools, this chapter is not required.

The following topics are covered:

## 5.1 Host Interface Global Registers Map

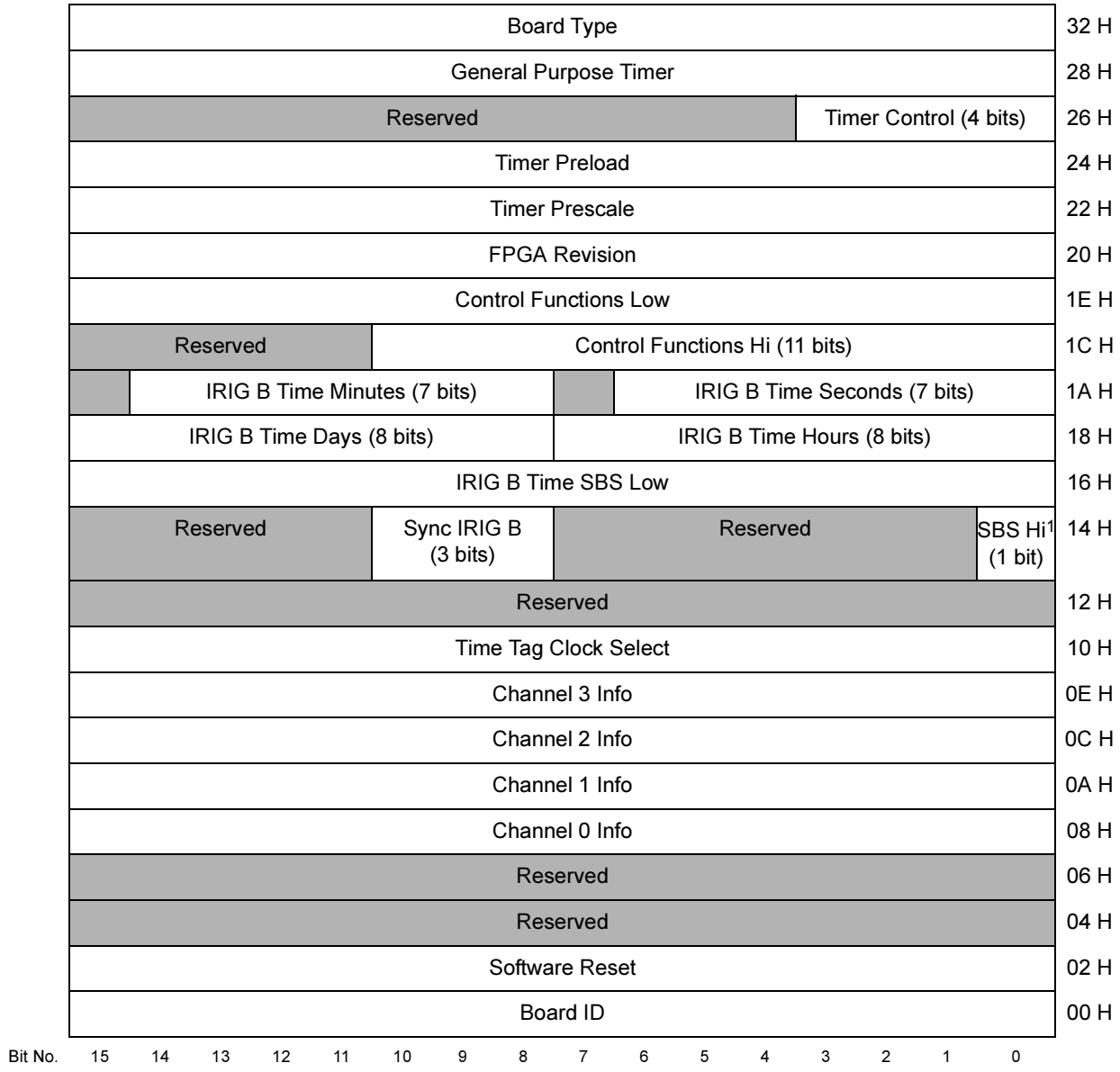



**Figure 5-1 Host Interface Global and IRIG B Registers Map**

1. IRIG B Time SBS Hi Register

### 5.1.1 Board Identification Register

**Address: 00 (H)**
**Length 16 bits**

**Read only** The Board Identification register comprises the following identification items.

| Bit | Description |
|---|---|
| **00-15** | Hard coded to the value 4000 H |

**Table 5-1 Board Identification Register**

### 5.1.2 Software Reset Register

**Address: 02 (H)**
**Length 16 bits**

**Read/Write** The Software Reset register performs reset operations of the channels. Individual channels may be reset.

Bit 04, the Global Time Tag reset bit, resets all the channel's Time Tag counters.

| Bit | Description | |
|---|---|---|
| **05-15** | Reserved – set to 0 | |
| **04** | Global time tag reset | 1 = reset all time tag counters<br>0 = no effect |
| **03** | Reserved – set to 0 | |
| **02** | Channel 2 reset | 1 = reset channel<br>0 = no effect |
| **01** | Channel 1 reset | 1 = reset channel<br>0 = no effect |
| **00** | Channel 0 reset | 1 = reset channel<br>0 = no effect |

**Table 5-2 Software Reset Register**

### 5.1.3 Channel Info Registers

**Address: 08, 0A, 0C, 0E (H)**
**Length 16 bits each**

**Read only** The Channel Info Registers provide identification information for each of the channels. On the *UNET*, Channel 0 is always *M4K1553Px*, Channel 3 is always *M4KDiscrete* and Channel 1 can be either *M4K1553Px* or not installed.

| Bit | Description | |
|---|---|---|
| **12-15** | Channel ID | 00 H = Channel 0 Info register<br>01 H = Channel 1 Info register<br>02 H = Channel 2 Info register<br>03 H = Channel 3 Info register |
| **05-11** | Reserved – set to 0 | |
| **00-04** | Channel type | 05 H = M4K1553Px channel<br>0D H = M4KDiscrete channel<br>1F H = no channel installed |

**Table 5-3 Channel Info Registers**

### 5.1.4 Time Tag Clock Select Register

**Address: 10 (H)**
**Length 16 bits**

**Read/Write** The Time Tag Clock Select Register is used to set either an internal (1 MHz) or external source for the board's Global Time Tag Clock. See section **4.5.5 External Signals Connector Pinouts** on page 4-10, for details of the External Time Tag Clock.

| Bit | Description | |
|---|---|---|
| **01-15** | Reserved – set to 0 | |
| **00** | Time Tag Clock Select | 1 = External Source<br>0 = Internal Source [**Default**] |

**Table 5-4 Time Tag Clock Select Register**

### 5.1.5 FPGA Revision Register

**Address: 20 (H)**
**Length 16 bits**

**Read only** The FPGA Revision register contains the FPGA revision of the board.

## 5.2 IRIG B Host Interface Global Registers

The *UNET* is able to receive and decode standard serial IRIG B120 time code format signals (1 KHz carrier wave, sine wave – amplitude modulated, 100 peaks per second). The *EXC-1553UNET/Px* receives IRIG B signals via its External Signals Connector. See **4.5.5 External Signals Connector Pinouts** on page 4-10. The *ES-1553RUNET/Px* receives IRIG B signals via connector J3. See **4.6.3 Communications I/O Connector [J3]** on page 4-16.

The IRIG B signal, which contains 3 types of words within each Time Code Frame, can be used to synchronize the Time Tags of the channels on the *UNET*.

| | |
|---|---|
| 1st Word | Time-of-year in binary coded decimal (BCD) notation in hours, minutes and seconds. |
| 2nd Word | Set of bits reserved for decoding various control, identification and other special purpose functions. |
| 3rd Word | Seconds-of-day weighted in straight binary seconds (SBS) notation |

These three words can be stored and displayed in the IRIG B Host Interface Global registers 14 - 1E (H).

See **Figure 5-1 Host Interface Global and IRIG B Registers Map** on page 5-2 for the location of the registers on the memory map.

**Note:** The synchronization of IRIG B time can take up to two seconds. IRIG B functions are meant to be used on an occasional basis, not on a constant basis.

### 5.2.1 Sync IRIG B Register

**Address: 14 (H)**
**Bits 08 – 10**

**Read/Write** The 3-bit Sync IRIG B register controls the synchronization of a channel's Time Tags relative to the IRIG B input signal and the display of the IRIG B time within the IRIG B time registers.

| Bit | | Description |
|---|---|---|
| **10** | 1 | Set by board to indicate that the current IRIG B time has been stored in the IRIG B registers |
| | 0 | No IRIG B time has been stored in the IRIG B registers.<br>This bit must be reset by the user after the board has written a '1'. |
| **09** | 1 | Stores and displays the IRIG B time and control functions into the 6 IRIG B registers (14-1E [H]) corresponding to the previous valid IRIG B message. If bit 08 is set, then the IRIG B time will be stored at the same time that the Time tags are reset. To calculate the realtime to which the Time tags are synchronized the user will need to **add** '1' to the value of the IRIG B time stored into these registers. |
| | 0 | The previous valid IRIG B message should not be stored in the IRIG B registers.<br>This bit will be automatically reset by the board after the storage of the IRIG B time. |
| **08** | 1 | Resets and synchronizes Time Tags of all the channels to the next rising edge of the on-time Reference Point **Pr** of the IRIG B signal. Also sets Bit 09 to a value of '1' in order to store and display the IRIG B time and control functions into the 6 IRIG B registers. |
| | 0 | No reset/synchronization of Time tags relative to the **Pr** of the IRIG B signal.<br>This bit will be automatically reset by board after reset of time tags |

**Table 5-5 Sync IRIGB Register**

**Note:** All bits are read and write.

### 5.2.2 IRIG B Time SBS High Register

**Address: 14 (H)**
**Bit 0**

**Read only** The IRIG B Time SBS High register contains the MSB of the 17 bit straight binary representation of the seconds-of-day code word within the IRIG B message.

### 5.2.3 IRIG B Time SBS Low Register

**Address: 16 (H)**
**Bits 15 – 0**

**Read only** The IRIG B Time SBS Low register contains the lower 16 bits of the 17 bit straight binary representation of the seconds-of-day code word within the IRIG B message.

### 5.2.4 IRIG B Time Days Register

**Address: 18 (H)**
**Bits 15 – 6**

**Read only** The IRIG B Time Days register contains the days value of the BCD time-of-year subword within the IRIG B coded message.

### 5.2.5 IRIG B Time Hours Register

**Address: 18 (H)**
**Bits 5 – 0**

**Read only** The IRIG B Time Hours register contains the hours value of the BCD time-of-year subword within the IRIG B coded message.

### 5.2.6 IRIG B Time Minutes Register

**Address: 1A (H)**
**Bits 14 – 8**

**Read only** The IRIG B Time Minutes register contains the minutes value of the BCD time-of-year subword within the IRIG B coded message.

### 5.2.7 IRIG B Time Seconds Register

**Address: 1A (H)**
**Bits 6 – 0**

**Read only** The IRIG B Time Seconds register contains the seconds value of the BCD time-of-year subword within the IRIG B coded message.

### 5.2.8 Control Functions Registers

**Hi Register Address: 1C (H) / Bits 10 – 0**
**Low Register Address: 1E (H) / Bits 15 – 0**

**Read only** The IRIG B time code formats reserve 27 bits known as Control Functions. The Control Functions are for user-defined encoding of various control, identification or other special purpose functions. No standard coding system exists. The control bits may be programmed in any predetermined coding system.

### 5.2.9 FPGA Revision Register

**Address: 20 (H)**
**Bits 15 – 0**

**Read only** The FPGA Revision register contains the FPGA revision of the board.

## 5.3 Global Timer Registers

See **Figure 5-1** on page 5-1 for location of the registers on the memory map.

### 5.3.1 Timer Prescale Register

**Address: 22 (H)**
**Bits 15 – 0**

**Read/Write** The Timer Prescale Register defines the resolution of the General Purpose Timer. It is based on the Global Time Tag Clock (nominally 1 MHz) and thus will give the General Purpose Timer resolution as follows:

| Timer Prescale Register Value (DEC) | General Purpose Time Resolution (μsec) |
|---|---|
| 0 or 1 | 1 (default) |
| 2 | 2 |
| 3 | 3 |
| • | • |
| • | • |
| • | • |
| 10 | 10 |
| • | • |
| • | • |
| • | • |
| 65535 | 65535 |

**Table 5-6 Timer Prescale/General Purpose Timer Resolution**

**Note:** The Timer Prescale register can only be changed when the timer has been stopped.

### 5.3.2 Timer Preload Register

**Address: 24 (H)**
**Bits 15 – 0**

**Read/Write** The value stored in the Timer Preload Register sets the starting count value for the General Purpose Timer from which it will start to count down. The Timer Preload Register can only be changed while the timer is stopped and has a maximum count value of 65535.

**Note:** The General Purpose Timer will not start counting if a value of zero is stored into the Timer Preload Register.

Default value: 00 00

### 5.3.3 Timer Control Register

**Address: 26 (H)**
**Bits 3 – 0**

**Read/Write** The Timer Control Register is used to control the General Purpose Timer register. The value stored in bits 01 to 03 take effect when the General Purpose timer reaches a value of zero. Bit 00 is used to start and stop the General Purpose Timer. The values of bits 01 – 03 can only be changed when the General Purpose Timer register is stopped.

Default value: 00 00

| Bit | Description | | |
|---|---|---|---|
| **04-15** | Reserved - set to 0 | | |
| **03** | Global reset on count completed | 1 | Causes global reset of all installed channels |
| | | 0 | No effect |
| **02** | Reserved | | |
| **01** | Reload mode | 1 | Reload mode |
| | | 0 | Non-reload/One-shot mode |
| **00** | Start/Stop | 1 | Start |
| | | 0 | Stop |

**Table 5-7 Timer Control Register**

### 5.3.4 General Purpose Timer Register

**Address: 28 (H)**
**Bits 15 – 0**

**Read Only** The General Purpose Timer Register stores the current count value of the General Purpose Timer. The General Purpose Timer is controlled by the Timer Control Register. When the General Purpose Timer is started it will count down to zero, at which point either an interrupt can be generated and or all installed channels can be reset.

If the General Purpose Timer is in reload mode then the current value in Timer Preload Register will be stored into the General Purpose Timer and the timer will start to count down from this value.

If the General Purpose Timer is in non-reload / one shot mode, when it reaches zero it will stop and a value of zero will be displayed in the General Purpose Timer Register. In this case bit 00 (Start/Stop bit) of the Timer Control Register will automatically be set to zero in this case. If the General purpose Timer Register is then started it will start to count from the current Timer Preload Register value automatically (without the need to do a write to the Timer Preload Register).

At any point in time, the General Purpose Timer can be stopped at the current count value. When a start is then issued, the General purpose Timer will start to count down from this current count value. If the user wishes to stop the counter and start from the original preload value or from a new preload value, this value will need to be rewritten into the Timer Preload register prior to the restarting of the General Purpose Timer register.

**Note:** The maximum clock period of the General Purpose Timer is 4295 seconds (1 hour, 11min & 35 Seconds).

### 5.3.5 Board Type Register

**Address: 32 (H)**
**Length 16 bits**

**Read only** The Board Type register comprises the following items.

| Bit | Description |
|---|---|
| **00-15** | Hard coded to the value 5502 H |

**Board Type Register**

# 6 MIL-STD-1553 Channel Remote Terminal Operation (Advanced)

Chapter 6 describes how to operate the MIL-STD-1553 channel in Remote Terminal (RT) mode via the memory registers, which enables you to develop your own software tools. When using Excalibur's software tools, this chapter is not required.

The following topics are covered:

## 6.1 RT Mode Overview

Each multifunction channel can be configured to simulate up to 32 remote terminals. The user selects which terminal or terminals are active and can inject errors into message responses. The single function module (*PxS*) simulates only one RT and does not support error injection. For more information on the single function module, see **1.5.4 MIL-STD-1553 Single Function Option** on page 1-9.

After receiving the Start command, the channel handles the transfer of all messages. See **6.10 Control Register Definitions** on page 6-17. Data associated with a particular RT subaddress combination is transferred via a 2K × 8 Look-up Table, which points to one of 256 Data Blocks (512 when using Expanded Block mode). The user loads Data Blocks associated with transmit commands with 1553 data to transmit, and reads received 1553 data from Data Blocks associated with receive commands.

The channel will respond properly to messages received with an intermessage gap time of 4 μsec.

The user can choose whether the remote terminal should transmit its 1553 Status Word at the end of a message, even if the message contains *invalid* Data Words (invalid as specified by MIL-STD-1553). Use the RT Protocol Options register, to activate or disable this feature. See **6.10.18 RT Protocol Options Register** on page 6-27.

The remote terminal transmits its 1553 Status Word in approximately 4 μsec. Use the RT Response Time register to increase the time it takes the remote terminal to transmit the 1553 Status Word.

Since most 1553 parameters, such as response time, Status Word content, etc. are user programmable, the channel can operate in various 1553 environments. The channel also allows you to enable or disable the 1553 Broadcast function. If broadcasting is enabled, RT address 31 (11111) is reserved; if broadcasting is disabled, all 32 RT addresses are available. See **6.10.20 Broadcast Control Register** on page 6-28.

The 1553 Mode Code subaddress identifier can be programmed (see **6.10.41 Mode Code Control Register** on page 6-36) so that either 31, 0, or both are used to indicate that the 1553 Command Word is a Mode Code.

**To determine whether the** channel **is installed and ready to operate:**

Perform the following procedure after a power-up or a software reset.

1. Check the Channel ID register (test for value = 45 H)
2. Check the Channel Status register (test for Channel Ready bit = 1)

The channel is installed and ready when both registers contain the correct values, as written above. For software reset operations, set these values to 0 immediately prior to writing to the channel Software Reset register.

**Note:** Throughout this manual, writing a '1' to the Start register is referred to as "issuing a Start command".

## 6.2 RT Memory Map

When Using the Expanded Data Block Option (512 Blocks Total)

| | |
|---|---|
| Expanded Data Block Area (256 Additional Blocks) | C000 – FFFF H |
| Internal Concurrent Monitor Message Block Area[1] | 8000 – BFFF H |

When Using the Default Data Block Option

| | | | |
|---|---|---|---|
| Internal Concurrent Monitor Message Block Area[1] | 8000 – FFFF H | 1760 Header Exist Table[2] | 3EC0 – 3EFF H |
| 1553 Data Blocks (56 Additional Blocks) | 7200 – 7FFF H | RT Message Stack Pointer | 3EBE – 3EBF H |
| Reserved | 7022 – 71FF H | Reserved | 3EA6 – 3EBD H |
| RT Number Register[3] | 7020 – 7021 H | Channel Time Register (Lo) | 3EA4 – 3EA5 H |
| Reserved | 700C – 7019 H | Channel Time Register (Hi) | 3EA2 – 3EA3 H |
| Time Tag (Hi) | 700A – 700B H | Serial Number Register | 3EA0 – 3EA1 H |
| Time Tag (Lo) | 7008 – 7009 H | Error Counter (Lo) | 3E9E – 3E9F H |
| Time Tag Reset Register | 7007 H | Error Counter (Hi) | 3E9C – 3E9D H |
| Reserved | 7004 – 7006 H | Message Counter (Lo) | 3E9A – 3E9B H |
| Options Select Register | 7003 H | Message Counter (Hi) | 3E98 – 3E99 H |
| Reserved | 7001 – 7002 H | Reserved | 3E96 – 3E97 H |
| Channel Reset Register | 7000 H | RTid with Bad Block Number Register | 3E94 – 3E95 H |
| RT Message Stack (512 blocks) | 6000 – 6FFF H | Bad Block Number Register | 3E92 – 3E93 H |
| Broadcast SAid Control Table[3] | 5800 – 5FFF H | Internal Concurrent Monitor Next Message Pointer[1] | 3E90 – 3E91 H |
| RTid Control Table | 4800 – 57FF H | Reserved | 3E8A – 3E8F H |
| Data Block Look-up Table | 4000 – 47FF H | Clear Time Tag on Sync Register | 3E88 – 3E89 H |
| Channel Configuration Register | 3FFF H | More Channel Options Register | 3E86 – 3E87 H |
| Channel ID Register | 3FFE H | Channel Options Register | 3E84 – 3E85 H |
| Channel Status Register | 3FFD H | Reserved | 3E81 – 3E83 H |
| Start Register | 3FFC H | Firmware Revision Register | 3E80 H |
| Message Received Status Register | 3FFB H | Reserved | 34C0 – 3E7F H |
| Reserved | 3FF8 – 3FFA H | 1553 RT Vector Word Table[4] | 3480 – 34BF H |
| Time Tag Resolution Register | 3FF7 H | 1553 RT BIT Word Table[4] | 3440 – 347F H |
| Bit Count Register | 3FF6 H | RT Last Command Word Table[4] | 3400 – 343F H |
| Reserved | 3FF5 H | Reserved | 33FD – 33FF H |
| RT Response Time Register | 3FF4 H | Interrupt Condition Register | 33FC H |
| Error Injection Register | 3FF3 H | Old RT Message Stack (42 blocks) | 3300 – 33FB H |
| Reserved | 3FF2 H | Word Count Error Table[5] | 32E0 – 32FF H |
| Old RT Message Stack Pointer | 3FF0 – 3FF1 H | Reserved | 3267 – 32DF H |
| RT Protocol Options Register | 3FEF H | Mode Code Control Register | 3266 H |
| Reserved | 3FEC – 3FEE H | 1760 Checksum Limits Register[2] | 3264 – 3265 H |
| Channel Function Register | 3FEA – 3FEB H | Reserved | 3260 – 3263 H |
| Broadcast Control Register | 3FE8 – 3FE9 H | 1553 RT Status Word Table[4] | 3220 – 325F H |
| Reserved | 3F80 – 3FE7 H | RT Settings Table[4] | 3200 – 321F H |
| 1760 Header Value Transmit Table[2] | 3F40 – 3F7F H | 1553 Data Blocks (200 Blocks) | 0000 – 31FF H |
| 1760 Header Value Receive Table[2] | 3F00 – 3F3F H | | |

**Figure 6-1RT Memory Map**

1. See **Chapter 9: MIL-STD-1553 Channel Internal Concurrent Monitor (Advanced)**
2. Only for 1760 option
3. Only for single function channel (*PxS*); in a multifunction channel this register is reserved
4. On a single function channel (*PxS*) only the first location is used; the rest is reserved
5. On a single function channel (*PxS*) this register is reserved

## 6.3 Data Block Look-up Table

### 6.3.1 Data Block Look-up Table for Multifunction Modules

When a command is received by the channel, the RTid, which consists of the RT Address, T/R Bit and the Subaddress of the Command Word, is used to index the datablock Look-up Table and get the datablock number that the user assigned to the RTid. (For more details on RTid, see **6.3.1.3 RT Identifier (RTid)** on page 6-7.) The Data Block Look-up Table is located at 4000 – 47FF (H).

The user assigns datablocks to transmit RTids (and uses the datablock to set data to be sent out by this RTid) and receive RTids (the channel will store data received by the RTid in this datablock) by filling in the corresponding entry in the datablock Look-up Table as shown in the table below.

**Note:** Data Block 0 represents a default for all unassigned RTids and is not recommended for use by anyone interested in using the data. If an RTid does not have a Data Block assigned to it, the default Data Block (0) is used for both receive and transmit messages.

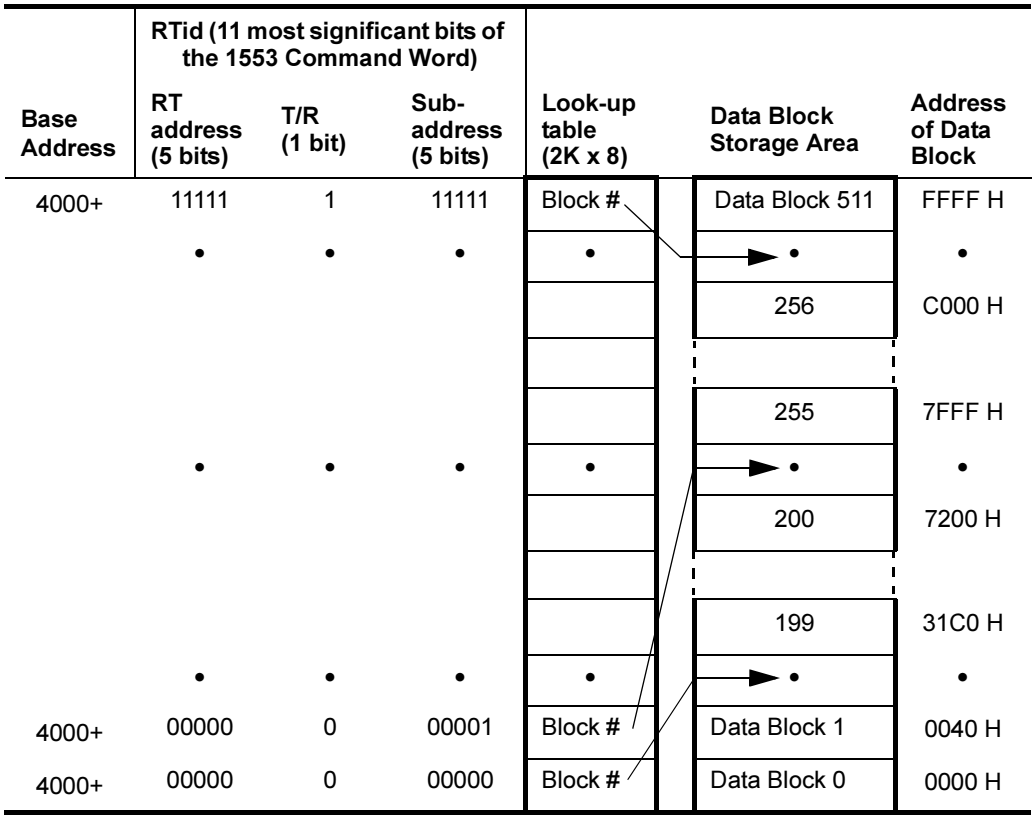



**Figure 6-2 Data Block Look-up Table for Multifunction Channel**

#### 6.3.1.1 Data Block Storage Areas

There are 512 datablocks available. Each block contains 32 1553 Data Words (64 bytes). Data Block 0 begins at address 0000. Data Block 1 begins at address 0040 (H), etc.

- 0 – 199 are stored in consecutive addresses, starting at address 0000 H.
- 200 – 255 are stored in consecutive addresses, starting at address 7200 H.
- 256 – 511 are stored in consecutive addresses, starting at address C000 H (available only in Expanded Block mode).

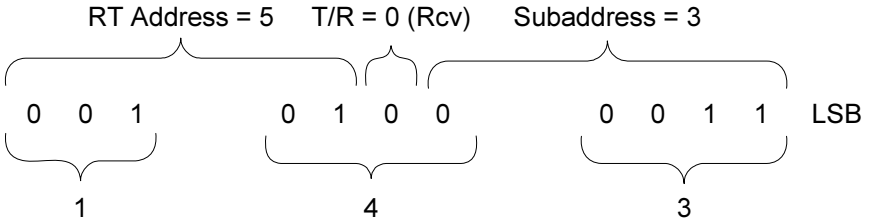
When Expanded Block mode is disabled, there are 256 available datablocks. When Expanded Block mode is enabled, 512 datablocks are available, but the Internal Concurrent Monitor message area is reduced from 409 to 204 messages.

In order to use Expanded Block mode, it must be enabled via the Module Function register. See **6.10.19 Channel Function Register** on page 6-27. In addition, set Bit 06 of the corresponding RTid entry in the RTid Control Table (0040 H) to 1. This adds an additional high bit (a value of 256) to the number found in the corresponding RTid entry in the Look-up table. For example, if an RTid is assigned to block number 100, the RTid is now assigned to block number 356. See **6.10.7 RTid Control Table** on page 6-20.

**To create an address to a Look-up table:**

1. Isolate the eleven most significant bits of the 1553 Command Word (RT Address, T/R, and Subaddress field), and determine their hex value.

**Example:** To allocate a Data Block for a 1553 receive message to RT#5, Subaddress 3.




Hex representation = 143 (H)

2. Add the Hex value of this part of the Command Word to the base address of the Look-up table (4000 H).

| | |
|---|---|
| 4000 | (H) |
| +143 | (H) |
| 4143 | (H) |

3. Write the Data Block number to address 4143 H.

**Example:** Writing a 1 to address 4143 H allocates block #1 for the data of this message. Read the 1553 data out by reading block #1, which starts at address 0040 H. Each Data Block, beginning at address 0000 is 64 bytes long (for up to 32 1553 Data Words). The address of a block is obtained by multiplying its block number by 64 (40 H).

#### 6.3.1.2 Data Storage

Within a specific Data Block, the 1553 Data Words must be stored and/or read in the following format:

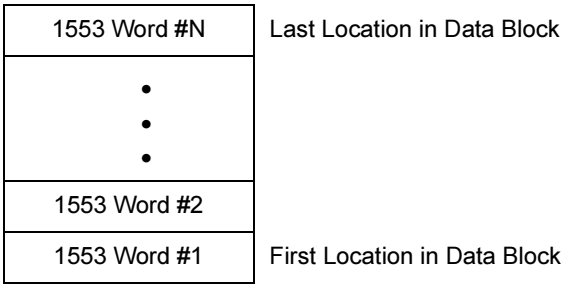



**Figure 6-3 Data Storage Sequence**

#### 6.3.1.3 RT Identifier (RTid)

The RTid is referred to frequently in the RT Control registers. The RTid is defined as an RT address, T/R bit value and RT subaddress combination. The structure of the RTid is illustrated below:

| 5 bits | 1 bit | 5 bits |
|---|---|---|
| RT ADDRESS | T/R[1] | SUBADDRESS |

**RT Identifier**

1. Transmit = 1 / Receive = 0

**Example:** RT#5, Transmit, Subaddress 6 would be represented as 00101 1 00110 (0166 H). This value can be isolated from a Command Word by shifting the Command Word 5 bits to the right.

**Mode Codes:** In the case of Mode Code the last 5 bits signify the Mode Code instead of the Subaddress.

See **To create an address to a Look-up table:** on page 6-6.

### 6.3.2 Data Block Look-up Table for Single Function (*PxS*) Modules

When a command is received by the channel, the SAid, which consists of the T/R Bit, Subaddress and Word Count, is used to index the datablock Look-up Table and get the datablock number that the user assigned to the SAid. (For more details on SAid, see **6.3.2.3 Subaddress Identifier (SAid)** on page 6-9.) The Data Block Look-up Table is located at 4000 – 47FF (H).

The user assigns datablocks to transmit SAids (and uses the datablock to set data to be sent out by this SAid) and receive SAids (the channel will store data received by the SAid in this datablock) by filling in the corresponding entry in the datablock Look-up Table as shown in the tables below.

**Note:** Data Block 0 represents a default for all unassigned SAids and is not recommended for use by anyone interested in using the data. If an SAid does not have a Data Block assigned to it, the default Data Block (0) is used for both receive and transmit messages.

| Base Address | SAid (11 least significant bits of the 1553 Command Word) T/R (1 bit) | Sub-address (5 bits) | Word Count (5 bits) | Look-up table (2K x 8) | Data Block Storage Area | Address of Data Block |
|---|---|---|---|---|---|---|
| 4000+ | 1 | 11111 | 11111 | Block # | Data Block 511 | FFFF H |
| | • | • | • | • | • | • |
| | | | | | 256 | C000 H |
| | | | | | 255 | 7FFF H |
| | • | • | • | • | • | • |
| | | | | | 200 | 7200 H |
| | | | | | 199 | 31C0 H |
| | • | • | • | • | • | • |
| 4000+ | 0 | 00001 | 00000 | Block # | Data Block 1 | 0040 H |
| 4000+ | 0 | 00000 | 00000 | Block # | Data Block 0 | 0000 H |

**Figure 6-4 Data Block Look-up Table for Single Function (*PxS*) Channel**

#### 6.3.2.1 Data Block Storage Areas

The Data Block Storage Areas for the single function (*PxS*) module are the same as the multifuction module, except that they use the SAid pointer instead of the RTid. See **6.3.1.1 Data Block Storage Areas** on page 6-6.

#### 6.3.2.2 Data Storage

Data storage for the single function (*PxS*) module is the same as the multifuction module. See **6.3.1.2 Data Storage** on page 6-7.

#### 6.3.2.3 Subaddress Identifier (SAid)

The SAid is defined as a T/R bit, subaddress and word count combination. The structure of the SAid is illustrated below:

| 1 bit | 5 bits | 5 bits |
|---|---|---|
| T/R[1] | SUBADDRESS | WORD COUNT |

**Subaddress Identifier**

1. Transmit = 1 / Receive = 0

**Example:** Transmit, Subaddress 6, Word Count 5 would be represented as 1 00110 00101 (04C5 H). This value can be isolated from a Command word by masking out the upper 5 bits (for example, Command word & 07FF H).

## 6.4 RT Settings Table

### 6.4.1 RT Settings Table for Multifunction Channel

The 32 locations (bytes) of the RT Settings table (3200 – 321F H) contains settings for each remote terminal, including which RTs are active. The first byte in the RT Settings table relates to RT #0, the next to RT #1, and the last location relates to RT #31.

To set RT as active, set Bit 00 of the corresponding RT Settings byte to logic 1.

| | | |
|---|---|---|
| RT#31 Settings Byte | 32nd byte | 321F H |
| • • • | • • • | • • • |
| RT#0 Settings Byte | 1st byte | 3200 H |

**Figure 6-5 RT Settings Table for Multifunction Channel – RT Mode**

**Note:** When operating in Broadcast mode, the active RT Look-up Table entry must be set for RT#31 as Not Active.

The following table describes the bits in each byte in the RT Settings table.

| Bit | Bit Name | Description |
|---|---|---|
| **05-07** | **0** | Reserved |
| **04** | **Status Word Duration** | 1 = One time<br>0 = Always<br>This bit determines the duration of user-defined 1553 RT Status Word. If this bit is set to 1, the user-defined Status Word for this RT is used only one time. Then all user-defined bits are cleared (only the RT address field remains as set). See **6.10.43 1553 RT Status Word Table** on page 6-37. |
| **03** | **Inactive Bus B** | 1 = Bus B Inactive<br>0 = Bus B Active<br>If Bit 00 is set to 1 and Bit 03 is set to 0, the channel will respond to all messages received over bus B for this RT. If Bit 03 is set to 1, all messages over bus B for this RT will be ignored. |
| **02** | **Inactive Bus A** | 1 = Bus A Inactive<br>0 = Bus A Active<br>If Bit 00 is set to 1 and Bit 02 is set to 0, the channel will respond to all messages received over bus A for this RT. If Bit 02 is set to 1, all messages over bus A for this RT will be ignored. |
| **01** | **Interrupt** | 1 = Interrupt<br>0 = No Interrupt<br>If Bit 01 is set to 1, if the RT is active and the Interrupt Condition register is enabled, then this channel will generate an interrupt as per the Interrupt Condition register. See **6.10.38 Interrupt Condition Register** on page 6-34.<br>**Note:** If the interrupt bit is set in the RT Settings Table, the interrupt setting in the RTid Control Table is ignored. To set an interrupt at the RTid level, make sure the interrupt bit in the RT Settings Table is disabled. |
| **00** | **Active** | 1 = RT Active<br>0 = RT Not Active<br>If Bit 00 is set to 0, the RT is not active, and none of the other bit settings are relevant. If it is set to 1, the other bits are checked.<br>**Note:** On a single function module (*PxS*), this bit is reserved. |

**RT Settings Byte**

### 6.4.2 RT Settings Table for Single Function (PxS) Channel

On a single function channel (*PxS*), the first byte (3200 H) is for the RT settings of the active RT and the fifth byte (3204 H) shows the RT number of the active RT. The rest of the bytes are reserved. (The active RT is selected via the RT Number register. See **6.10.1 RT Number Register (PxS Only)** on page 6-17.).

| | |
|---|---|
| Reserved | 3205 – 321F H |
| Active RT Number Byte (read only) | 3204 H |
| Reserved | 3201 – 3203 H |
| RT Settings Byte | 3200 H |

**Figure 6-6 RT Settings Table for Single Function (*PxS*) Channel – RT Mode**

The bits of the RT Settings byte are the same as for multifunction channel. See page 6-10.

#### 6.4.2.1 Active RT Number

The following table describes the bits of the Active RT Number Byte.

| Bit | Description |
|---|---|
| **05-07** | Reserved |
| **00-04** | The number of the active RT |

**Active RT Number**

## 6.5 RT Message Stack

The RT Message Stack is located at 6000 – 6FFF H.

In RT mode, the channel generates a message stack in the dual-port RAM. This stack contains information used for post-processing of RT messages. The stack is divided into 512 blocks, each containing four words, including a 32-bit Time Tag value. The stack operates as a circular buffer. Only messages relating to active RTs are stored, **Figure 6-7** illustrates one block. The RT Message Stack pointer points to the beginning of the next unused block. See **6.10.24 RT Message Stack Pointer** on page 6-30.

| | Byte Offset |
|---|---|
| Message Status Word | +6 |
| Time Tag Word Lo | +4 |
| Time Tag Word Hi | +2 |
| 1553 Command Word | 0 |

**Figure 6-7 RT Message Stack Block Structure**

#### How the Channel Updates an RT-to-RT Message

When an RT-to-RT message is received, where the channel is functioning as both RTs, the message stack is updated as follows:

Two message stack blocks are utilized.

1. The 1553 Receive Command Word is written into the *first* message stack block.
2. The 1553 Transmit Command Word and Message Status Word are written into the second stack block.
3. The Message Status Word is written into the first (Receive) message stack block.

Both the Receive side and Transmit side message stack blocks contain the identical Message Status Word, with the RT-to-RT bit set to 1, and the same Time Tag Word.

### 6.5.1 Old RT Message Stack

The original *Px* card had a single message stack with 42 message entries and a 16-bit Time Tag. In later versions of the *Px* card, a new message stack was added with 512 message entries and a 32-bit Time Tag.

For new applications it is recommended only to use the new message stack. However, for backward compatibility the old message stack is available. It is located in the dual-port RAM at 3300 – 33FB (H) and is divided into 42 blocks each containing 3 words, including a 16-bit Time Tag value which is the lower 16-bits of the 32-bit Time Tag Counter. The pointer to this stack is at 3FF0 (H).

Currently only the new message stack is enabled by default. To enable the old message stack, set Bit 02 of the RT Protocol Options register. See **6.10.18 RT Protocol Options Register** on page 6-27.

### 6.5.2 Message Status Word

The Message Status Word indicates the status of the message transfer. The channel creates this word. Do not confuse this word with the 1553 Status Word (see **6.10.43 1553 RT Status Word Table** on page 6-37). The contents of the Message Status Word are:

| Bit | Bit Name | Description |
|---|---|---|
| **15** | End of Message | Message transfer completed |
| **14** | Bus A / B | Bus on which the message was transferred:<br>0 = Bus B<br>1 = Bus A |
| **13** | 1760 Checksum Error | The calculated checksum (on an incoming message) does not match the last Data Word received |
| **11 – 12** | Reserved | Set to 0 |
| **10** | Tx Time Out | Channel, acting as receiver in RT-to-RT message, did not sense a transmitter Status Word (in 14 μsec.) |
| **09** | Superseding | A new command word to the same RT was detected in middle of receiving this message. The new command will be placed in the following message stack entry. |
| **08** | 1760 Header Error | Header Word received does not match the value set in the Header Value table.<br>See **6.9.1 Header Word** on page 6-15. |
| **07** | Invalid Word Received | At least one invalid 1553 Word received (i.e. bit count, Manchester code, parity) |
| **06** | Reserved | Set to 0 |
| **05** | Word Count Error (Receive Message) | Incorrect number of words received in the message |
| **04** | Broadcast Message | Broadcast Command Word received |
| **03** | Incorrect Sync Received | Sync of either the Status or the Data Word(s) is incorrect |
| **02** | Non- Contiguous Data (Receive Message) | Invalid gap between received 1553 Words |
| **01** | RT-RT Message | RT-to-RT message received |
| **00** | Error | Error occurred (The error type is defined in one of the other message status bit locations) |

**Remote Terminal Message Status Word**

**Note:** A logic 1 indicates the occurrence of a status flag

The Message Status Word is valid only when Bit 15, End of Message, is turned on.

### 6.5.3 Time Tag

**Read only**

The Time Tag value is a 32-bit word that can be used to determine the time elapsed since reset. The Time Tag uses a 32-bit, free-running counter whose resolution is set by the Time Tag Resolution register. The equation to determine the Time Tag resolution = (Time Tag Resolution register value + 1) × 4 μsec.

**Example:**

Time Tag Resolution register value = 0 → Counter's resolution = 4 μsec
Time Tag Resolution register value = 4 → Counter's resolution = 20 μsec

To reset the Time Tag counter (to 0) any time, write to the Time Tag Reset register. See **6.10.3 Time Tag Reset Register** on page 6-18.

When the first command of each message is received, the value of the 32-bit Time Tag Counter register is written to dual-port RAM.

**Note:**

1. The counter's value can be read at any time by reading the Time Tag counter addresses. (See **6.10.2 Time Tag Counter** on page 6-17.)
2. The counter can also be clocked and/or reset from an external source (*EXC-1553UNET/Px* only). (See **4.5.5 External Signals Connector Pinouts** on page 4-10.)

**Example:** How To Calculate Elapsed Time

Time Tag Resolution register = 03 (initialized before Start command)

Time Tag values (read during or after message transfers):

Low = 0040 (H)
High = 0010 (H)

Time elapsed since Start command
= (Time Tag register value) × (Time Tag Resolution value + 1) × (4 μsec)
= 100040 (H) × (03 + 1) × 4 μsec
= 1048640(Dec) × (4 × 4 μsec) = 16778240 μsec (16778.24 msec. = 16.778 sec.)

### 6.5.4 1553 Command Word

The Command Word location contains the 1553 Command Word associated with the message.

Only active RT Command Words are stored.

## 6.6 Mode Codes

The user can program the Subaddress code that will indicate that a Mode command has been received. Either or both of the following codes can be used: 11111 and 00000. The Mode Code Control register must be programmed as described in section **6.10.41 Mode Code Control Register** on page 6-36.

The channel handles all dual-redundant 1553B Mode Codes. The Word Count field is decoded according to MIL-STD-1553B. One of the Mode Codes (Synchronize with Data Word) is operated upon as a standard message transfer, using the Data Block Look-up Table. When the channel encounters the Synchronize with Data Word Mode Code, the Command Word's RT Address, T/R bit, and Subaddress fields are used as a pointer to the Look-up Table. The table

entries that are addressed when the T/R bit = 0 and Subaddress = 00000 or 11111 should contain a Data Block number (0 – 255) indicating where the Synchronize with Data Word's data word should be stored.

The data associated with Mode Codes (Transmit Last Command, Transmit Bit word, and Transmit Vector word) is set using the dedicated blocks in the dual-port RAM (described in **6.10.37 RT Last Command Word Table**, **6.10.36 1553 RT BIT Word Table**, and **6.10.35 1553 RT Vector Word Table** on page 6-33).

## 6.7 Broadcast Mode

To operate the module in Broadcast mode, select the appropriate bit settings as defined in **6.10.20 Broadcast Control Register** on page 6-28.

When operating in Broadcast mode, the active RT Look-up Table entry must be set for RT#31 as Not Active. The module reads the Broadcast Control register to determine whether the module is operating in Broadcast mode.

In Broadcast mode, the module stores the received message in a 1553 Data Block area in the same way as standard message formats. RT address, T/R bit, and Subaddress are used as a pointer to the Data Block Look-up Table memory.

## 6.8 Error Injection Feature

The channel allows two types of error injection:

- **Global (for all RTs)** – The global errors such as Sync and Non-Contiguous data are described in **6.10.16 Error Injection Register** on page 6-26. These errors are either ON or OFF for all RTs.
- **Per RT** – The ability to inject a 1553 Word Count error can be set per RT using the Word Count Error table. See **6.10.40 Word Count Error Table** on page 6-35.

**Note:** Error injection is not available on a single function channel (*PxS*).

## 6.9 1760 Option

### 6.9.1 Header Word

In the MIL-STD-1760 specification, the first Data Word of a message may be a Header Word, which is used for message identification. The Header Word is associated with a specific Subaddress.

To indicate that a specific subaddress will require a Header Word, set the corresponding entry in the Header Exist table to 1. Then set the corresponding entry in the Header Transmit/Receive Value table to the value you expect to receive in the first Data Word of the message. The Header value expected is either the predefined 1760 value, which is the default channel setting, or another value the user enters in the Header Value Transmit/Receive Table. The channel checks that the specified Header receive value was received. In addition, the Internal Concurrent Monitor checks that the specified Header transmit/receive value was received. If the wrong data was received, the **1760 Header error** bit is set

in the Message Status Word. See **6.5.2 Message Status Word** on page 6-13.

See **6.10.21 1760 Header Value Transmit Table**, **6.10.22 1760 Header Value Receive Table** and **6.10.23 1760 Header Exist Table** on page 6-30.

### 6.9.2 1760 Checksum Error

MIL-STD-1760 implements checksum error detection capabilities. Checksums are calculated as each Data Word is received. Upon an incoming message, the calculated checksum is compared to the last Data Word received. If it does not match, the Checksum Error bit is set in the Message Status Word.

**Note:** Error injection is not available on a single function channel (*PxS*).

## 6.10 Control Register Definitions

### 6.10.1 RT Number Register (*PxS* Only) Address: 7020 – 7021 (H)

The RT Number register contains the channel's RT address and related information for a single function channel (*PxS*).

On the *EXC-1553UNET/Px*: Bits 00 – 04 represent the RT address. Bit 05 is the parity bit. Upon reset, these bits default to the values set by onboard jumpers. The values of these bits must be modified to the desired RT address and parity by issuing a Stop command, modifying the register, and then issuing a Start command. See **6.10.11 Start Register** on page 6-23. Bit 06 is set to 0 (unlocked) by an onboard jumper. (A single function *EXC-1553UNET/PxS* can be ordered with the jumpers set to a specific RT address and parity with the Bit 06 set to 1 (locked). Contact Excalibur Sales for more information. See **1.8 Technical Support** on page 1-12.)

On the *ES-1553RUNET/Px*: Upon reset, bits 00 – 06 default to the values set by the connector pins. (See **4.6.3 Communications I/O Connector [J3]** on page 4-16.) To modify the values of bits 00 – 05 of this register, issue a Stop command, modify the register, and then issue a Start command. (See **6.10.11 Start Register** on page 6-23). Bit 06 is a read only bit indicating whether the RT number is locked via a connector pin, in which case the RT number cannot be modified and is taken from the connector pins.

| Bit | Bit Name | Description |
|---|---|---|
| **08 – 15** | Reserved | |
| **07** | RTERR | 0 = RT address parity OK<br>1 = RT address parity error<br>**(Read only)** |
| **06** | RTLOCK | 0 = RT number is unlocked and bits 00 – 05 can be modified by changing the values of this register<br>1 = RT number is locked and cannot be modified by changing the values of this register<br>**(Read only)** |
| **05** | RTPTY | RT Address Parity Bit. This bit is appended to the remote terminal address bus to supply parity. Odd parity is required for proper operation. |
| **00 – 04** | RTNUM | RT Address Bits. These five bits contain the remote terminal address. 00 is the Least Significant Bit (LSB). |

**RT Number Register**

### 6.10.2 Time Tag Counter Address: 7008 – 700B (H)

**Read only**

The Time Tag is a free-running 32-bit counter on the channel. The Time Tag is reset upon a power up or a software reset and starts counting. When it reaches the value FFFF FFFF (H), the counter wraps around to 0 and continues counting. To re-initialize to 0, write to the Time Tag Reset register.

The user may read the Time Tag counter at any time. Read the two 16-bit words of the Time Tag counter value sequentially, first Lo word, then Hi word.

**The counter must be read in the following sequence:**

1. Read 7008 H – Lo word (16 bit, read only)
2. Read 700A H – Hi word (16 bit, read only)

The Time Tag resolution register sets the resolution of the counter. See **6.10.13 Time Tag Resolution Register** on page 6-24.

**To calculate elapsed time between Time Tags:**

**Example:**

1. The Time Tag Resolution register is set to 0. (See **6.10.13 Time Tag Resolution Register** on page 6-24.)
2. Calculate the Time Tag Resolution:
   (Time Tag Resolution register value + 1) × 4 = (0 + 1) × 4 = 4 μsec
3. Calculate difference between Time Tags:
   150 (Time Tag 2) – 50 (Time Tag 1) = 100
4. Elapsed time
   100 × 4 = 400 μsec

### 6.10.3 Time Tag Reset Register

**Address: 700 (H)**

**Write only**

Write any value to the Time Tag Reset register to reset the channel's Time Tag Counter. Immediately after the reset, the counter will start to count from 0.

**Note:** The counter may also be reset from an external source (*EXC-1553UNET/Px* only). See **4.5.5 External Signals Connector Pinouts** on page 4-10.

### 6.10.4 Options Select Register

**Address: 700 (H)**

**Write only**

Write to the Options Select register to select whether RT address 11111 (RT #31) is interpreted as a valid RT address or as a Broadcast address.

| Bit | Description | |
|---|---|---|
| **01 – 07** | Reserved | |
| **00** | 1 | Broadcast option is active. RT#31 is Broadcast Address. No RT Status Word will be transmitted. |
| | 0 | Broadcast option is inactive, RT#31 is a regular RT. **Note:** Bit 00 has been retained for backward compatibility. For new application, use the Broadcast Control register instead. See **6.10.20 Broadcast Control Register** on page 6-28. |

**Options Select Register**

**Note:** The Options Select register is reset at power-up (all bits set to 0) or by a channel reset.

### 6.10.5 Channel Reset Register

**Address: 700 (H)**

Write any value to the Channel Reset register to reset the channel.

Channel Reset erases all locations in the dual-port RAM. Channel status, channel ID and Firmware Revision registers are written by the channel after the reset operation is completed.

### 6.10.6 Broadcast SAid Control Table (*PxS* Only)

**Address: 5800 – 5FFF (H)**

The Broadcast SAid Control Table is a block of memory, one byte per SAid, to store SAid-specific settings, and to implement SAid-specific features.

| | SAid (11 least significant bits of the 1553 Command Word) | | | | |
|---|---|---|---|---|---|
| **Base Address** | **T/R (1 bit)** | **Subaddress (5 bits)** | **Word Count (5 bits)** | **Broadcast SAid Control table (2K x 8)** | |
| 5800+ | 1 | 11111 | 11111 | SAid Illegalization byte | 5FFF H |
| | • | • | • | • | • |
| | • | • | • | • | • |
| | • | • | • | • | • |
| 5800+ | 0 | 00000 | 00001 | SAid Illegalization byte | 5801 H |
| 5800+ | 0 | 00000 | 00000 | SAid Illegalization byte | 5800 H |

**Figure 6-8 Broadcast SAid Control Table**

| Bit | Bit Name | Description |
|---|---|---|
| **03 – 07** | Reserved | |
| **02** | Illegalization | 1 = The Message Error bit (Bit 10) is set in the 1553 RT Status Word of the next Mode Code 2 or Mode Code 18 message after the current BC-to-RT or Mode Code Message (see **6.10.43 1553 RT Status Word Table** on page 6-37). |
| **00 – 01** | Reserved | |

**Broadcast SAid Control Table Byte**

**Note:** The Broadcast SAid Control Table is only for the single function channel (*PxS*). On a multifunction channel, this register is reserved.

### 6.10.7 RTid Control Table

**Address: 4800 – 57FF (H)**

The RTid Control Table is a block of memory, one word per RTid, to store RTid-specific settings, and to implement RTid-specific features.

| Base Address | RTid (11 most significant bits of the 1553 Command Word) RT address (5 bits) | T/R (1 bit) | Sub-address (5 bits) | Word Alignment[1] | RTid Control table (2K x 16) | |
|---|---|---|---|---|---|---|
| 4800+ | 11111 | 1 | 11111 | 0 | RTid Information word | 57FE H |
| | • | • | • | 0 | • | • |
| | • | • | • | 0 | • | • |
| | • | • | • | 0 | • | • |
| 4800+ | 00000 | 0 | 00001 | 0 | RTid Information word | 4802 H |
| 4800+ | 00000 | 0 | 00000 | 0 | RTid Information word | 4800 H |

**Figure 6-9 RTid Control Table**

1. Additional bit for “even” addressing.

| Bit | Bit Name | Description |
|---|---|---|
| **12 – 15** | Next Buffer | These bits are used for multibuffering. They contain the number of the next buffer to be used by the firmware, when either receiving or transmitting data. This field is updated by the channel as soon as the processing of a command begins. For example, when the channel begins processing a command using buffer 2, it updates the NextBuffer field to “3.” This field can be updated by the user, though care should be taken not to update it at the same time as the firmware, or the update could be lost. Initially, this field is set to “0” and is incremented by 1 for each successive use of the next buffer. |
| **08 – 11** | Multibuffers | These bits are written by the user to direct the firmware to use multibuffering.<br>0 = (**Default**) Multibuffering is turned off<br>1 = Two buffers are used<br>2 = Three buffers are used<br>...<br>15 = 16 buffers are used<br>**Note:** Multibuffering can not be used together with double buffering (Bit 04); the user must select one mechanism or the other. |
| **07** | Reserved | |
| **06** | Expanded Data Block Bit | High order bit for block numbers above 255 (Expanded Block mode only) |

**RTid Information Word**

| Bit | Bit Name | Description |
|---|---|---|
| **05** | Double buffer[1] datablock usage flag | The bit indicates which block of the double buffering pair to use for storing the data for this RTid.<br>0 = (**Default**) The channel detects a receive message. The channel stores the data at the even-numbered block number indicated in the Look-up Table. When the channel completes writing all the Data Words to the block, the channel sets this bit to 1. This indicates to write to the odd-numbered block the next time receive data comes in for this RTid.<br>1 = The channel detects a receive message. The channel stores the data at the odd-numbered block, whose number is one more than the even-number indicated in the Look-up Table. When the channel completes writing all the Data Words to the block, the channel sets this bit to 0. This indicates to write to the even-numbered block the next time receive data comes in for this RTid. |
| **04** | Double Buffering (Receive) selected[1] | The bit indicates that the channel will double buffer data for the receive messages for this RTid.<br>When the channel receives messages for an RTid, the data is stored in the assigned datablock. If no datablock is assigned, data is stored in the default datablock (number 0). When two messages arrive for the same RTid, the data of the second message will overwrite the data of the first message.<br>To preserve the data of the first message long enough to be able to read it before it gets overwritten, use a double buffering scheme to save the data of the last two messages; i.e., use two buffers, alternatively, so that the channel can capture data to one buffer, and simultaneously the user can read data from the other buffer.<br>To implement double buffering, the channel requires that the datablock assigned to this RTid be an even number. The channel then reserves the following odd-numbered block as the paired block for use in double buffering.<br>**Note:** If Double Buffering is enabled for this RTid, and the block number selected is odd, double buffering does not occur. Instead:<br>• A flag at Bit 02 in the Message Received Status register is set (see **6.10.12 Message Received Status Register** on page 6-24);<br>• The selected (odd) block number is written to the Bad Block Number register (3E92 H) to indicate the error (see **6.10.30 Bad Block Number Register** on page 6-31);<br>• The RTid is written to the RTid with Bad Block Number register (3E94 H, see **6.10.29 RTid with Bad Block Number Register** on page 6-31). |
| **03** | Inactive[1] | The RTid is inactive – message is not processed at all; the RT does not send back a Status Word.<br>**Note:** RT response time must be set to at least 5 μsec., otherwise there will be extraneous words on the bus. |
| **02** | Illegalization | The ME bit (Bit 10) is set in the 1553 RT Status Word (see **6.10.43 1553 RT Status Word Table** on page 6-37). Only the Status Word is sent back.<br>For **BC-to-RT** (receive) messages – processing continues as per regular algorithm, and the ME bit is written in the STW.<br>For **RT-to-BC** (transmit) messages – no data is sent.<br>For **RT-to-RT** messages – receive part: the ME bit is written in the STW; transmit part: no data is sent<br>For **Mode Code** messages – the ME bit is written in the STW<br>**Note:** On a single function channel (*PxS*), illegalization is done based on the SAid, not the RTid. See **6.3.2.3 Subaddress Identifier (SAid)** on page 6-9. |
| **01** | Interrupt on Error[2] | Generates an interrupt on error |
| **00** | Interrupt on end of message[2] | Generates an interrupt on end of message |

**RTid Information Word (Continued)**

1. Not applicable to Mode Codes
2. If the interrupt bit is set in the RT Settings Table, the interrupt setting in the RTid Control Table is ignored. To set an interrupt at the RTid level, make sure the interrupt in the RT Settings Table is disabled.

### 6.10.8 Channel Configuration Register

**Address: 3FF (H)**

Use the Channel Configuration register to set the operating mode of the channel.

Set the Channel Configuration register before issuing a Start command to the channel. To modify the Channel Configuration register, issue a Stop command, modify the register, and then issue a Start command. (See **6.10.11 Start Register** on page 6-23).

| Hex Value | Operating Mode |
|---|---|
| **02** | RT mode |

**Channel Configuration Register**

### 6.10.9 Channel ID Register

**Address: 3FF (H)**

The Channel ID register contains a fixed value that can be read by the initialization routine to detect the presence of the channel. The one-byte value of this register is 45 (H), ASCII value E.

### 6.10.10 Channel Status Register

**Address: 3FF (H)**

The Channel Status register indicates the status of the channel. In addition, this register indicates which options have been selected. Do not modify this register. Status bits are active if set to '1'.

| Bit | Description |
|---|---|
| **07** | 1 = Always set |
| **05 – 06** | Indeterminate |
| **04** | 1 = Channel Halted<br>0 = Channel Running |
| **03** | 1 = Self-Test OK |
| **02** | 1 = Timers OK |
| **01** | 1 = RAM OK |
| **00** | 1 = Channel Ready |

**Channel Status Register**

**Note:** Channel operation stops after the Start bit in the Start register is cleared. Following this, the channel sets Bit 04 (Channel Halted). Certain registers may be modified only after the Channel Halted bit has been set. After receiving a subsequent Start command (by writing to the Start register), the channel resets the Channel Halted bit. The condition of this bit after power-up or software reset is logic '1'.

### 6.10.11 Start Register

**Address: 3FF (H)**

The Start register controls the Start/Stop operation of the channel. The user can Start or Stop the RT operation, modify RT parameters, for example: the Error Injection register or RT Response Time register, and then issue a new Start command in real time.

See also **6.10.10 Channel Status Register**, Bit 04 (Channel Halted/Running).

| Bit | Description |
|---|---|
| **01 – 07** | 0 |
| **00** | 1 = Start<br>0 = Stop |

**Start Register**

**Note:** You can start the module externally by sending a minimum LVTTL pulse of 100 nsec. to the EXSTARTn pin. See **4.5.5 External Signals Connector Pinouts** on page 4-10.

### 6.10.12 Message Received Status Register

**Address: 3FF (H)**

The Message Received Status register indicates that a 1553 message has been received. A logic '1' indicates active condition. This bit is also set for messages with errors.

| Bit | Description |
|---|---|
| **01 – 07** | 0 |
| **00** | Message Content |

**Message Received Status Register**

**Note:** After reading, reset the Message Complete bit; the channel does not reset this bit.

### 6.10.13 Time Tag Resolution Register

**Address: 3FF (H)**

The 8-bit value in the Time Tag Resolution register represents the resolution of the Time Tag Counter in units of 4 μsec.

To determine the Time Tag Counter's resolution, use the following equation:

= (Time Tag Resolution register value + 1) × 4 μsec.

A value of 0 corresponds to a resolution of 4 microseconds; a value of 1 corresponds to a resolution of 8 microseconds, etc.

Set the Time Tag Resolution register before issuing a Start command to the channel. To modify the Time Tag Resolution register, issue a Stop command, modify the register, and then issue a Start command. (See **6.10.11 Start Register** on page 6-23.)

### 6.10.14 Bit Count Register

**Address: 3FF (H)**

The Bit Count register sets the total number of bits in the 1553 Word, including Sync (3) and Parity (1). This register is used by the channel only for messages for which the Bit Count Error bit is set in the Error Injection register. (See **6.10.16 Error Injection Register** on page 6-26). If, the Bit Count Error bit is not set, a (valid) 20-bit word is transmitted regardless of the contents of the Bit Count register.

Set the Bit Count register before issuing a Start command to the channel. To modify the Bit Count register, issue a Stop command, modify the register, and then issue a Start command. (See **6.10.11 Start Register** on page 6-23.)

| Bit | Description | | | |
|---|---|---|---|---|
| 03 – 07 | 0 | | | |
| 00 – 02 | Bit 02 | Bit 01 | Bit 00 | Number of 1553 bits sent per word |
| | 0 | 0 | 0 | 17 (-3) |
| | 0 | 0 | 1 | 18 (-2) |
| | 0 | 1 | 0 | 19 (-1) |
| | 0 | 1 | 1 | 20 |
| | 1 | 0 | 0 | 21 (+1) |
| | 1 | 0 | 1 | 22 (+2) |
| | 1 | 1 | 0 | 23 (+3) |

**Bit Count Register**

**Note:** On a single function channel (*PxS*) this register is reserved. Error injection is not available.

### 6.10.15 RT Response Time Register

**Address: 3FF (H)**

The RT Response Time register sets the Response Time of the remote terminal. The resolution of the Response Time register is 155 nsec. per bit. The minimum time is approximately 4 μsec., which is achieved by writing a 0 to this register. Any value above zero results in:

Response Time = 4 μsec.+ (RT Response Time register x 155 nsec.)

Tolerance of response time: ± 1 μsec.

Set the Response Time register before issuing a Start command to the channel. To modify the Response Time register, issue a Stop command, modify the register, and then issue a Start command. (See **6.10.11 Start Register** on page 6-23.)

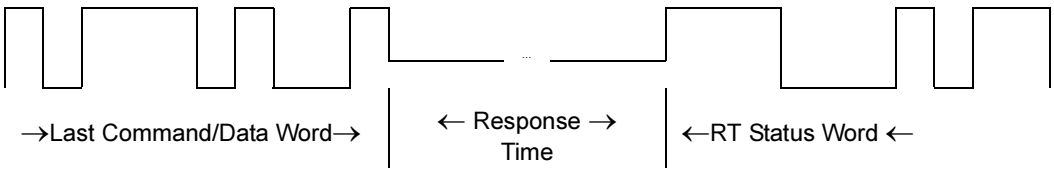



**Figure 6-10 RT Response Time Definition**

**Example:** To request a Response time of 9 μsec:
Write 32 to the RT Response Time register
$32 \times 0.155 \cong 5$ μsec + 4 μsec = 9 μsec

### 6.10.16 Error Injection Register

**Address: 3FF (H)**

The Error Injection register is a global register that allows the user to select the type of error to be injected in a transmitted message. When the channel receives a Start command (issued by writing to the Start register), the channel reads this register.

To modify the Error Injection register, issue a Stop command, modify the register, and then issue a Start command. (See **6.10.11 Start Register** on page 6-23)

| Bit | Description |
|---|---|
| **07** | Data Word Sync Error<br>(Data Words sent with Command Sync) |
| **06** | Data Word Parity Error<br>(Data Words sent with Even Parity) |
| **05** | Status Word Synchronization Error<br>(Status Word Sent With Data Sync) |
| **04** | Status Word Parity Error<br>(Status Word sent with Even Parity) |
| **03** | Reserved – set to 0 |
| **02** | Non-Contiguous Data<br>(Between First and Second Data Word) |
| **01** | Bit Count Error<br>See **6.10.14 Bit Count Register** on page 6-25. |
| **00** | Reserved – set to 0 |

**Error Injection Register**

**Note:** On a single function channel (*PxS*) this register is reserved. Error injection is not available.

### 6.10.17 Old RT Message Stack Pointer

**Address: 3FF0 – 3FF1 (H)**

The Old RT Message Stack pointer indicates the Old RT Message Stack position. After the entire message is received, the pointer is updated (incremented by 6). This word is initialized to 3300(H) and circulates in the message stack between 3300(H) and 33FB(H). This pointer has been retained for backward compatibility. See **6.5.1 Old RT Message Stack** on page 6-12.

For information on the current RT Message Stack pointer, see **6.10.24 RT Message Stack Pointer** on page 6-30.

### 6.10.18 RT Protocol Options Register

**Address: 3F (H)**

Bit 02 of the RT Protocol Options register is used to select which message stacks to use.

Bit 01 of the RT Protocol Options register is used to select a 1553 environment: MIL-STD-1553A or MIL-STD-1553B.

Depending on the 1553 environment selected, certain bits in the 1553 RT Status Word will be affected. See **6.10.43 1553 RT Status Word Table** on page 6-37.

Bit 00 of the register is used to set the Status Response (Suppress or Send status) mode of operation.

If set to 'send' after a Receive message, the RT will can respond with a 1553 Status Word even if an invalid 1553 Data Word was received.

To define Mode Codes as SA-31 and process as 1553A compatible, see **6.10.41 Mode Code Control Register** on page 6-36.

The RT Protocol Option register must be set before issuing a Start command to the channel. To modify the RT Protocol Option register, issue a Stop command, modify the register, then, issue a Start command. (See **6.10.11 Start Register** on page 6-23).

| Bit | Description | |
|---|---|---|
| **03 – 07** | 0 | |
| **02** | Message Stack | 0 = (**Default**) Only the new message stack is available, 512 blocks<br>1 = Both message stacks are available; new (512 blocks) and old (42 blocks)<br>See **6.2 RT Memory Map** on page 6-4 for addresses. |
| **01** | Environment | 0 = 1553B<br>1 = 1553A Mode Code compatibility<br>Mode Codes are user-defined (except for MC-0), subaddresses are set to 0, no Data Words. RT sends back a Status Word only. |
| **00** | On Error | 0 = Suppress Status<br>1 = Send Status |

**Status Response Register**

### 6.10.19 Channel Function Register

**Address: 3FEA – 3FEB (H)**

Set Bit 00 of the Channel Function register to 1 to enable Expanded Block mode. Expanded Block mode increases the available block numbers to 511, and reduces the Internal Concurrent Monitor from 409 to 204 messages.

### 6.10.20 Broadcast Control Register

**Address: 3FE8 – 3FE9 (H)**

Set the Broadcast Control register to specify whether RT address 11111 (RT #31) should be regarded as a valid RT number or as the Broadcast address.

In Broadcast mode, the channel stores the received message in a 1553 Data Block area in the same way as standard message formats. RT address, T/R bit, and Subaddress are used as a pointer to the Data Block Look-up Table memory.

**Note:**

- When operating in Broadcast mode, the active RT Look-up Table entry must be set for RT#31 as Not Active.
- This register replaces the use of Bit 00 of the Options Select register. See **6.10.4 Options Select Register** on page 6-18.

| Bit | Description |
|---|---|
| **01 – 15** | 0 |
| **00** | 1 = RT #31 is Broadcast Address<br>0 = RT #31 is Regular RT |

**Broadcast Control Register**

### 6.10.21 1760 Header Value Transmit Table

**Address: 3F40 – 3F7F H**

**1760 Option only**

Write to the 1760 Header Value Transmit table to set the expected value of the first Data Word in a RT-to-BC message. If the wrong data was sent, the Internal Concurrent Monitor will set an error bit. See Bit 06 in the **Message Status Word: RT/ Internal Concurrent Monitor** on page 9-3.

The 1760 option provides predefined values, and these are preset on each channel. The user can change the preset values.

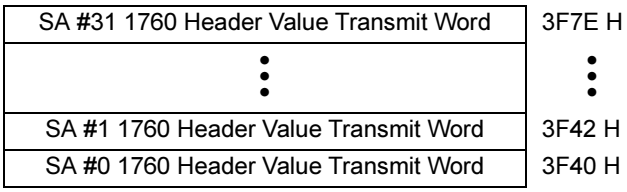



**Figure 6-11 1760 Header Value Transmit Table**

| Transmit Subaddress | Header Value | Address |
|---|---|---|
| **1** | **0421 H** | 3F42 H |
| **11** | **0420 H** | 3F56 H |
| **14** | **0423 H** | 3F5C H |

**Predefined 1760 Transmit Header Values**

### 6.10.22 1760 Header Value Receive Table

**Address: 3F00 – 3F3F (H)**

**1760 Option only**

Write to the 1760 Header Value Receive table to set the expected value of the first Data Word in a BC-to-RT message. The channel checks that the specified Header receive value was received. In addition, the Internal Concurrent monitor checks that the specified header value was received. If the wrong data was sent, the **1760 Header Error** bit is set in the Message Status Word. See **6.5.2 Message Status Word** on page 6-13.

The 1760 option provides predefined values, and these are preset on each channel. The user can change the preset values.

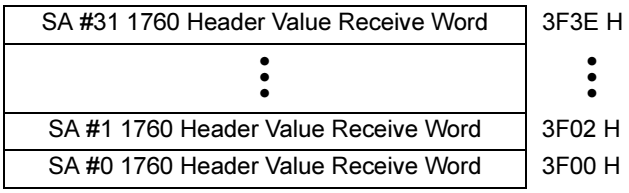



**Figure 6-12 1760 Header Value Receive Table**

| Receive Subaddress | Header Value | Address |
|---|---|---|
| **11** | **0400 H** | 3F16 H |
| **14** | **0422 H** | 3F1C H |

**Predefined 1760 Receive Header Values**

### 6.10.23 1760 Header Exist Table

**Address: 3EC0 – 3EFF (H)**

**1760 Option only**

The 1760 Header Exist Table contains 32 entries corresponding to 32 Subaddresses. Each entry may be set to indicate whether the channel should expect a Header word for BC-to-RT or RT-to-RT messages directed to that subaddress.

For those Header Value Table entries for which MIL-STD-1760 provides predefined values, the corresponding Header Exist Table entries are preset on each channel.

To set other values, enable the Header Exist Table entry for this Subaddress (set it to 1) and write the value to the Header Value (Transmit/Receive) Table.

| | |
|---|---|
| SA #31 1760 Header Exist Word | 3EFE H |
| ⋮ | ⋮ |
| SA #1 1760 Header Exist Word | 3EC2 H |
| SA #0 1760 Header Exist Word | 3EC0 H |

**Figure 6-13 1760 Header Exist Table**

| Bit | Description |
|---|---|
| **09-15** | Reserved |
| **08** | 1 = Channel should expect a Header word in a transmit message (RT-to-BC or RT-to-RT)<br>0 = Channel should **not** expect a Header word in a transmit message |
| **01 – 07** | Reserved |
| **00** | 1 = Channel should expect a Header word in a receive message (BC-to-RT)<br>0 = Channel should **not** expect a Header word in a receive message |

**1760 Header Exist Table**

| Associated Subaddress | Header Value | Address |
|---|---|---|
| **1** | **0100 H** | 3EC2 H |
| **11** | **0101 H** | 3ED6 H |
| **14** | **0101 H** | 3EDC H |

**Predefined 1760 Headers**

### 6.10.24 RT Message Stack Pointer

**Address: 3EBE – 3EBF (H)**

The RT Message Stack pointer indicates the next word to be written to the RT message stack. After the entire message is received, the message stack pointer is updated (incremented by 8). This word is initialized to 6000 (H) and circulates in the message stack between 6000(H) and 6FFF(H).

### 6.10.25 Channel Time Register Lo & Hi

**Address: 3EA4 – 3EA5 (H)**
**3EA2 – 3EA3 (H)**

This register holds the channel time value, which is stored in non-volatile flash memory and loaded at power-up. This value can be modified by calling the Set_ModuleTime_Px function. (See the *1553Px Family Software Tools Programmer's Reference*.) The factory default value is FFFF FFFF (H).

### 6.10.26 Serial Number Register

**Address: 3EA0 – 3EA1 (H)**

This register holds the board's serial number, which is stored in non-volatile flash memory and loaded at power-up. The value is binary coded. For example, a value of 1234 (H) represents the serial number 4660.

### 6.10.27 Error Counter Lo & Hi

**Address: 3E9E – 3E9F (H)**
**3E9C – 3E9D (H)**

Error Counter is a running 32-bit counter of message errors.

### 6.10.28 Message Counter Lo & Hi

**Address: 3E9A – 3E9B (H)**
**3E98 – 3E99 (H)**

Message Counter is a running 32-bit counter of all messages received.

### 6.10.29 RTid with Bad Block Number Register

**Address: 3E94 – 3E95 (H)**

When using double-buffering, odd-numbered block numbers are invalid. The RTid with Bad Block Number register indicates the associated RTid for which the user attempted to set an odd-numbered block when using double-buffering.

### 6.10.30 Bad Block Number Register

**Address: 3E92 – 3E93 (H)**

When using double-buffering, odd-numbered block numbers are invalid. The Bad Block Number register indicates the selected invalid odd-numbered block.

### 6.10.31 Clear Time Tag on Sync Register

**Address: 3E88 – 3E89 (H)**

Write 1 to the lower byte (3E88 H) of the Clear Time Tag on Sync register to indicate that the channel should clear the Time Tag counter (7008 – 700B H) (resets to 0) upon receipt of a Mode Code 1 message (synchronize). A value of 0 disables this function.

Write 1 to the higher byte (3E89 H) of the Clear Time Tag on Sync register to indicate that the channel should clear the Time Tag counter (7008 – 700B H) (resets to 0) upon receipt of a Mode Code 17 message (synchronize with data). A value of 0 disables this function.

**Note:** This register setting does not take effect until the channel is restarted.

### 6.10.32 More Channel Options Register

**Address: 3E86 – 3E87 (H)**

**Read only**

The More Channel Options register is a 16-bit register that provides additional channel information.

| Bit | Description |
|---|---|
| **06 – 15** | Reserved |
| **05** | 1 = Expanded Block mode is available in BC mode<br>0 = Expanded Block mode is not available in BC mode |
| **04** | 1 = Enhanced Monitor mode is available in Sequential Fixed-Block Monitor mode<br>0 = Enhanced Monitor mode is not available in Sequential Fixed-Block Monitor mode |
| **03** | 1 = Expanded Block mode is available in Sequential Fixed-Block Monitor mode<br>0 = Expanded Block mode is not available in Sequential Fixed-Block Monitor mode |
| **02** | 1 = Channel is single function (*PxS*)<br>0 = Channel is multifunction (*Px*) |
| **01** | 1 = Onboard Loopback option is available<br>0 = Onboard Loopback option is not available |
| **00** | 1 = Channel is only available in Monitor mode<br>0 = Channel is available in all modes |

**More Channel Options Register**

### 6.10.33 Channel Options Register

**Address: 3E84 – 3E85 (H)**

**Read only**

The Channel Options register is a 16-bit register that provides information about the internal processor and firmware.

| Bit | Description |
|---|---|
| **15** | 1 = PxIII |
| **14** | Reserved; set to 1 |
| **13** | 1 = Expanded Block mode is in use in RT mode |
| **12** | 1 = Channel is on a removable card (PCMCIA or ExpressCard)<br>0 = Channel is on an add-in board |
| **11** | 1 = Replay mode is in use (BC mode only) |
| **10** | 1 = PxII |
| **09** | 1 = 1760 |
| **08** | 1 = 1553 |
| **00 – 07** | 4D H Always set; indicates Internal Concurrent Monitor |

**Channel Options Register**

### 6.10.34 Firmware Revision Register

**Address: 3E8 (H)**

The Firmware Revision register indicates the revision level of the on-channel firmware. The value 18 (H) would read as revision level '1.8'.

### 6.10.35 1553 RT Vector Word Table

**Address: 3480 – 34BF (H)**

The RT Vector Word locations are reserved for the 32 1553 Vector words. (On a single function channel (*PxS*), only one word is used at 3480 H and the rest is reserved.) Load the desired Vector words into the corresponding locations in the block. The first word is for RT#0, the next word is for RT#1, and the last word is for RT#31. These words are used to implement the Transmit Vector Word Mode Code.

| | |
|---|---|
| RT #31 1553 RT Vector Word | 34BE H |
| ⋮ | ⋮ |
| RT #1 1553 RT Vector Word | 3482 H |
| RT #0 1553 RT Vector Word | 3480 H |

**Figure 6-14 1553 RT Vector Word Table**

**Note:** For a description of the BC's reaction to the SRQ bit and the Vector Word, see, **7.10 Service Request (SRQ) Processing** on page 7-15.

### 6.10.36 1553 RT BIT Word Table

**Address: 3440 – 347F (H)**

The RT BIT (Built-in Test) word locations are reserved for the 32 1553 BIT words. (On a single function channel (*PxS*), only one word is used at 3440 H and the rest is reserved.) Load the desired BIT words into the corresponding locations in the block. The first word is for RT#0, the next word is for RT#1, and the last word is for RT#31. These words are used to implement the Transmit BIT Word Mode Code.

| | |
|---|---|
| RT #31 1553 RT BIT Word | 347E H |
| ⋮ | ⋮ |
| RT #1 1553 RT BIT Word | 3442 H |
| RT #0 1553 RT BIT Word | 3440 H |

**Figure 6-15 1553 RT BIT Word Table**

### 6.10.37 RT Last Command Word Table

**Address: 3400 – 343F (H)**

The Last Command Word locations are reserved for the 32 1553 Last Command Words. (On a single function channel (*PxS*), only one word is used at 3400 H and the rest is reserved.) The channel writes to these locations at the end of each message transfer (for active RTs only). The first word is for RT#0, the next word is for RT#1, and the last word is for RT#31. These words are used for the implementation of the Transmit Last Command Word Mode Code.

**Note:** Only Command Words of valid messages containing no errors are recorded in this table.

| | |
|---|---|
| RT #31 Last Command Word | 343E H |
| ⋮ | ⋮ |
| RT #1 Last Command Word | 3402 H |
| RT #0 Last Command Word | 3400 H |

**Figure 6-16 RT Last Command Word Table**

### 6.10.38 Interrupt Condition Register

**Address: 33F (H)**

The Interrupt Condition register allows the user to enable an interrupt trigger. The bits work in conjunction with the Interrupt bit in the RT Settings Table. When a message is received by an RT for which the Active RT interrupt bit is set, the channel will check the Interrupt Condition register.

If the channel has completed receiving the Command Word and the Begin Data bit is also set, an interrupt trigger will be generated.

If the channel has completed processing the message and the Message Complete bit is also set, an interrupt trigger will be generated.

Set the Interrupt Condition register before issuing a Start command to the channel. To modify the Interrupt Condition register, issue a Stop command, modify the register, and then issue a Start command. (See **6.10.11 Start Register** on page 6-23.)

| Bit | Description |
|---|---|
| **02 – 07** | 0 |
| **01** | 1 = Message Complete |
| **00** | 1 = Begin Data |

**Interrupt Condition Register**

### 6.10.39 Old RT Message Stack

**Address: 3300 – 33FB H**

This message stack has been retained for backward compatibility. For more information, see **6.5.1 Old RT Message Stack** on page 6-12.

### 6.10.40 Word Count Error Table

**Address: 32E0 – 32FF (H)**

The Word Count Error is selected by writing to the Word Count Error table, which contains 32 bytes (one per Remote Terminal). The first byte is for RT#0, the second to RT#1, and the last byte is for RT#31. The contents of each location controls the number of 1553 Words (±3 words) in the message. The variation is an offset, relative to the 1553 Command Word's Word Count field. The resulting message (if an error is programmed) must contain at least one Data Word.

Upon power-up and software reset, the channel sets the Word Count Error Table to the default value, 0.

**Note:** On a single function channel (*PxS*) this register is reserved. Error injection is not available.

The user must set the Word Count Error register before issuing a Start command to the channel. To modify the Word Count Error register, issue a Stop command, modify the register, and then issue a Start command. See **6.10.11 Start Register** on page 6-23.

| | |
|---|---|
| RT #31 Word Count Error Byte | 32FF H |
| ⋮ | ⋮ |
| RT #1 Word Count Error Byte | 32E1 H |
| RT #0 Word Count Error Byte | 32E0 H |

**Figure 6-17 Word Count Error Table**

| Register Value | Word Count Offset |
|---|---|
| FD H | -3 Words |
| FE H | -2 Words |
| FF H | -1 Word |
| 00 H | No Error Injection |
| 01 H | +1 Word |
| 02 H | +2 Words |
| 03 H | +3 Words |

**Word Count Error Byte Values**

### 6.10.41 Mode Code Control Register

**Address: 326 (H)**

The Mode Code Control register allows the user to specify which 1553 Subaddress value indicates the reception of a 1553 Mode command.

Set the Mode Code Control register before issuing a Start command to the channel. To modify the Mode Code Control register, issue a Stop command, modify the register, and then issue a Start command. (See **6.10.11 Start Register** on page 6-23.)

| Bit | Description | | |
|---|---|---|---|
| **02 – 07** | 0 | | |
| **00 – 01** | **Bit 01** | **Bit 00** | **Subaddresses recognized as Mode Code** |
| | 0 | 0 | 31 and 0 |
| | 0 | 1 | 0 |
| | 1 | 0 | 31 |
| | 1 | 1 | 0 and 31 |

**Mode Code Control Register**

### 6.10.42 1760 Checksum Limits Register

**Address: 3264 – 3265 (H)**

**1760 Option only**

Write a value to the 1760 Checksum Limits register to set the Data Blocks for which the channel should calculate a checksum value. The channel will calculate a checksum value for those Data Blocks whose numerical index is less than the value stored in this register. Maximum value is 256 (512 when using Expanded Block mode).

**Example:** Write 20 to the Checksum Limits register when you want Data Blocks 0 – 19 to have a checksum value. This causes the channel to:

1. Transmit a checksum value as the last word in the Data Block (transmit message).
2. Check the last word in the Data Block for a checksum value (receive message).

### 6.10.43 1553 RT Status Word Table Address: 3220 – 325F H

These locations (3220 – 325F H) are reserved for the 32 1553 RT Status Words. (On a single function channel (*PxS*), only one word is used at 3220 H and the rest is reserved.) Load the desired Status Words into their respective locations in the block. The first word relates to RT#0, the next word to RT#1, while the last word relates to RT#31.

Whenever the RT has to respond with a Status Word, the channel sends out the Status Word as the user has defined it with the addition of certain other bits that the channel may set, as described below.

The user may set the RT Status Word for one-time use only. See Bit 04 of the RT Settings Table (see **6.4 RT Settings Table** on page 6-9). If Duration is set to 1 ('one-time'), the RT Status Word user-defined bits (bits 00 – 10) will be used only once, then cleared and set by the channel according to the rules described below. The top 5 bits, i.e. the RT Address Field remain unchanged.

**Note:** If an error occurred in the Command Word, the channel cannot send out an RT Status Word – the RT is unknown.

| | |
|---|---|
| RT #31 1553 RT Status Word | 325E H |
| ⋮ | ⋮ |
| RT #1 1553 RT Status Word | 3222 H |
| RT #0 1553 RT Status Word | 3220 H |

**Figure 6-18 1553 RT Status Word Table**

#### 6.10.43.1 RT Status Word Bits

In both MIL-STD-1553A and MIL-STD-1553B environments bits 11 – 15 are the RT Address Field and Bit 10 is the Message Error bit. Bits 00 – 09 are reserved in a 1553A environment but in a 1553B environment apply as shown in the following table.

| Bit | MIL-STD-1553A | MIL-STD-1553B | Description |
|---|---|---|---|
| **11 – 15** | RT Address | RT Address | |
| **10** | Message Error | Message Error | Invalid or illegal word/s received in preceding command |
| **09** | Reserved | Instrumentation | Always 0, to distinguish between cmd and status |
| **08** | Reserved | Service Request (SRQ) | Indicates to BC that RT needs servicing |
| **05 – 07** | Reserved | Reserved | |
| **04** | Reserved | Broadcast | Preceding Command Word was a broadcast command |
| **03** | Reserved | Busy | RT cannot send data in response to BC command |
| **02** | Reserved | Subsystem Flag | RT fault exists; data being requested may be invalid |
| **01** | Reserved | Dynamic Bus | Acceptance of offer by active BC to be the next BC |
| **00** | Reserved | Terminal Flag | RT fault condition (used with Mode Codes 6,7,19) |

**1553A and 1553B RT Status Word Bits**

The 1553 environment, 1553A or 1553B, is set by the user in the RT Protocol Options register. See **6.10.18 RT Protocol Options Register** on page 6-27.

**Message Error Bit (Bit 10)**

If the message is of type Send Status (see **6.10.18 RT Protocol Options Register** on page 6-27), and an error occurred in the data of a message, the channel will set the Message Error (ME) bit in the Status Word to 1 prior to sending it out.

If the message is of type Suppress Status, and an error occurred in the data of a message, the channel will not send out a Status Word at all. But, on the next message, the channel will send out the Status Word with the ME bit set to 1. The ME bit will be reset to the user-defined value when the next valid command is received by the RT (unless the next valid command is Mode Code Transmit Status Word or Mode Code Transmit Last Command).

If the RTid has been set illegal by the user, the channel will always set the ME bit to 1 in the Status Word sent back in response to a command for that RTid, whether the message is of type Send or Suppress Status.

**Service Request BIT (SRQ – Bit 08)**

Setting the SRQ bit indicates to the BC that a subaddress on this RT requires servicing. (For BC processing of this bit, see **7.10 Service Request (SRQ) Processing** on page 7-15).

**Note:** Simulated RTs do not support SRQ processing.

**Broadcast (Bit 04)**

If a Broadcast message is received, the channel does not send out a Status Word. But, each active RT, on the next message it receives, will send out the Status Word with the Broadcast Bit set to 1. (In a Broadcast RT-to-RT, the transmitting RT, even if active, will never have the broadcast bit set).

The Broadcast Bit will be reset (to the user-defined value) when the next valid command is received by the RT (unless the next valid command is Mode Code Transmit Status Word or Mode Code Transmit Last Command).

**Busy (Bit 03)**

In the case of Transmit commands, when the user has set the Busy bit, no Data Word will be transmitted by the RT following the transmission of the Status Word.

### 6.10.44 RT Settings Table

**Address: 3200 – 321F H**

The RT Settings Table is described earlier in this chapter. See **6.4 RT Settings Table** on page 6-9.

# 7 MIL-STD-1553 Channel BC/Concurrent-RT Operation (Advanced)

Chapter 7 describes how to operate the MIL-STD-1553 channel in Bus Controller/Concurrent-RT mode via the memory registers, which enables you to develop your own software tools. When using Excalibur's software tools, this chapter is not required.

The topics included are:

## 7.1 BC/Concurrent-RT Mode Overview

Each channel can simultaneously operate as the Bus Controller and up to 32 Remote Terminals. The messages and the instruction stack are loaded as for BC operation.

In BC/Concurrent-RT mode, load message blocks with the RT's 1553 Status and Data Words for those Remote Terminals that you are actively simulating. These words must be loaded into the appropriate locations in the message blocks in the sequence that the 1553 Words appear on the 1553 bus.

**Note:** The requirement for loading the message blocks only applies to RTs that the user is actively simulating. For inactive RTs (not simulated by the channel), leave the corresponding locations blank in the associated 1553 message blocks.

The Remote Terminals simulated in BC/Concurrent-RT mode have a minimum response time of approximately 4 μsec.

**To determine if the channel is installed and ready to operate:**

Perform the following procedure after a power-up or a software reset.

1. Check the Channel **ID Register** (test for value = 45 H).
2. Check the Channel **Status Register** (test for Channel Ready bit = 1).

The channel is installed and ready when both registers contain the correct values. For software reset operations, set these values to 0 immediately prior to writing to the channel Software Reset register.

**Note:** Throughout this manual, writing a '1' to the Start register is referred to as issuing a Start command.

## 7.2 BC/Concurrent-RT Memory Map

| | | | |
|---|---|---|---|
| Internal Concurrent Monitor Message Block Area[1] | 8000 – FFFF H | | |
| Instruction Stack / Message Block Area | 7100 – 7FFF H | Reserved | 3FD4 – 3FD7 H |
| Reserved | 700C – 70FF H | SRQ Counter | 3FD2 – 3FD3 H |
| Time Tag (Hi) | 700A – 700B H | SRQ Message Status Register | 3FCE – 3FD1 H |
| Time Tag (Lo) | 7008 – 7009 H | SRQ Message 2 Register | 3F88 – 3FCD H |
| Time Tag Reset Register | 7007 H | SRQ Message 1 Register | 3F80 – 3F87 H |
| Reserved | 7001 – 7006 H | 1760 Header Value Transmit Table[2] | 3F40 – 3F7F H |
| Channel Reset Register | 7000 H | 1760 Header Value Receive Table[2] | 3F00 – 3F3F H |
| Instruction Stack/Message Block Area | 4000 – 6FFF H | 1760 Header Exist Table[2] | 3EC0 – 3EFF H |
| Channel Configuration Register | 3FFF H | Reserved | 3EA6 – 3EBF H |
| Channel ID Register | 3FFE H | Channel Time Register (Lo) | 3EA4 – 3EA5 H |
| Channel Status Register | 3FFD H | Channel Time Register (Hi) | 3EA2 – 3EA3 H |
| Start Register | 3FFC H | Serial Number Register | 3EA0 – 3EA1 H |
| Interrupt Condition Register | 3FFB H | Error Counter (Lo) | 3E9E – 3E9F H |
| Message Status Register | 3FFA H | Error Counter (Hi) | 3E9C – 3E9D H |
| RT Response Time Register | 3FF9 H | Message Counter (Lo) | 3E9A – 3E9B H |
| Reserved | 3FF7 – 3FF8 H | Message Counter (Hi) | 3E98 – 3E99 H |
| Loop Count Register | 3FF6 H | Reserved | 3E92 – 3E97 H |
| Bit Error Register | 3FF5 H | Internal Concurrent Monitor Next Message Pointer[1] | 3E90 – 3E91 H |
| Word Count Register | 3FF4 H | Channel Function Register | 3E8E – 3E8F H |
| BC Response Time Register | 3FF3 H | BC Protocol Options Register | 3E8C – 3E8D H |
| Reserved | 3FF2 H | Send Time Tag on Sync Register | 3E8A – 3E8B H |
| Message Stack Pointer | 3FF0 – 3FF1 H | Clear Time Tag on Sync Register | 3E88 – 3E89 H |
| Frame Time Multiplier Register | 3FEE – 3FEF H | More Channel Options Register | 3E86 – 3E87 H |
| Frame Time Resolution Register | 3FEC – 3FED H | Channel Options Register | 3E84 – 3E85 H |
| Instruction Counter | 3FEA – 3FEB H | Reserved | 3E81 – 3E83 H |
| Minor Frame Time Register | 3FE8 – 3FE9 H | Firmware Revision Register | 3E80 H |
| Minor Frame Time Multiplier Register | 3FE6 – 3FE7 H | Reserved | 3426 – 3E7F H |
| Replay Register | 3FE4 – 3FE5 H | Asynchronous Start Flag | 3424 – 3425 H |
| Reserved | 3FDE – 3FE3 H | Asynchronous Frame Pointer Register | 3422 – 3423 H |
| Zero Cross Bit Index Register | 3FDC – 3FDDH | Asynchronous Message Count Register | 3420 – 3421 H |
| Error Word Index Register | 3FDA – 3FDB H | RT Settings Table | 3400 – 341F H |
| Sync Pattern Register | 3FD8 – 3FD9 H | Instruction Stack /Message Block Area | 0000 – 33FF H |

**Figure 7-1 BC/Concurrent-RT Memory Map**

1. See **Chapter 9: MIL-STD-1553 Channel Internal Concurrent Monitor (Advanced)**
2. 1760 Option only

## 7.3 Instruction Stack

The Instruction Stack is used to program the channel. The stack is divided into instruction blocks, each containing four words. The block contains control information (that the user writes) and status information (that the channel writes).

**Figure 7-2** illustrates one instruction block.

| **Control and status information is stored in the memory in the following sequence:** | **Byte Offset** |
|---|---|
| Message Status Word | +6 |
| Intermessage Gap Time Counter | +4 |
| Intermessage Gap Time | +2 |
| Message Block Pointer | 0 |

**Figure 7-2 Instruction Block Structure – BC/Concurrent-RT Mode**

### 7.3.1 Message Status Word

The Message Status Word indicates the status of the message transfer. The channel creates this word. Do not confuse this word with the RT 1553 Status Word. (See **2.4 1553 RT Status Word** on page 2-7). The contents of the Message Status Word are described below.

| Bit | Bit Name | Description |
|---|---|---|
| **15** | **End Of Message** | Message transfer completed |
| **14** | **Checksum Error** (1760 Option only) | The calculated checksum (on an incoming message) does not match the last Data Word received. See **7.11.2 Checksum** on page 7-16. |
| **13** | **Incorrect 1553 Channel** | Remote Terminal response was not received on the active 1553 channel. |
| **12** | **Message Error Bit** | Message Error bit (Bit 10) in the RT Status Word was set. |
| **11** | **RT Status Bit** | A bit was set in the RT Status Word (other than the Message Error bit). The error bit is not set in conjunction with this bit. |
| **10** | **Invalid Message Error** | A 1553 message-level error occurred (e.g. Word Count, incorrect sync); details in the bits described below. |
| **09** | **Response Time Failure** | RT responded late – see **7.12.15 BC Response Time Register** on page 7-23. |
| **08** | **1760 Header Word** (1760 Option only) | Header Word received does not match the value set in the Header Value Table. See **7.11.1 Header Word** on page 7-16. |
| **07** | **Invalid Word Received** | At least one invalid 1553 Word received (e.g., bit count, Manchester code, parity). |
| **06** | **Word Count High** | RT transmitted too many words. |
| **05** | **Word Count Lo** | RT transmitted too few words. |
| **04** | **Incorrect RT Address** | 1553 Status Word received did not contain the correct RT address. |
| **03** | **Incorrect Sync Received** | Sync of either the status or Data Word(s) is incorrect. |
| **02** | **Non-Contiguous Data** | Invalid gap between received 1553 Words. |
| **01** | **Reserved** | Set to 0 |
| **00** | **Error** | Error occurred. The error type is defined in one of the other message status bit locations. |

**Message Status Word**

**Note:**

- A logic 1 indicates occurrence of status flag.
- The Message Status Word is valid only when Bit 15, End of Message, is turned on.
- To ensure data integrity, the channel sets a special status value of 7F00 H (NO_ALTER) to indicate that a message is currently being transmitted or received. Check this value before attempting to change the Data words of the message.

### 7.3.2 Intermessage Gap Time

The Intermessage Gap Time (IGT) value is a 16-bit word that the user writes, that allows a unique intermessage delay time to be inserted between the current message and the next message. The minimum IGT is approximately 8 μsec. The maximum IGT is approximately 10 msec. that can be extended up to approximately 80 seconds, using the IGT counter value. (See **7.3.3 Intermessage Gap Time Counter/Message Function Select** on page 7-6.) The value in the word is added to this minimum time. The resolution of this word is 155 nsec. per bit.

### 7.3.3 Intermessage Gap Time Counter/Message Function Select

The 13 low bits are the Intermessage Gap Time counter (IGT_counter). It is written by the user, allowing to increase the Intermessage Gap Time by repeating the number of times the Intermessage Gap Time value is used.

The 3 high bits are used to select functionality for the message.

Bit 13 is used to instruct the channel to generate an interrupt when the specific message is completed.

The user sets bits 14 and 15 to instruct the channel to generate checksums and checksum error detection and injection. See **7.11.2 Checksum** on page 7-16.

| Bit | Bit Name | Description |
|---|---|---|
| **15** | **Chk_Sum_On**<br>(1760 option only) | BC-to-RT: Generate a checksum<br>*or*<br>RT-to-BC: Checks that the correct Checksum was transmitted |
| **14** | **Chk_Sum_Err_Inj**<br>(1760 option only) | BC-to-RT message: Injects an incorrect value into the checksum; Bit 15 must be set in order to set Chk_Sum_Err_Inj<br>**Note:** On a single function channel (*PxS*) this bit is reserved. Error injection is not available. |
| **13** | **Int_On_Select_Msg** | 1 = Generate an interrupt when this specific message is completed. To set general interrupt conditions.<br>See **7.12.9 Interrupt Condition Register** on page 7-19 |
| **00 – 12** | **IGT_counter** | Write a value to increase the IGT by repeating the number of times the IGT value is used.<br>For example, if the counter is set to 0, then the gap time is not repeated; and depends on the contents of the IGT location. If the gap time counter is 1, then the gap time is repeated once and equals the IGT value × 2, etc.<br>**Note:** To ensure maximum IGT accuracy when using the IGT_counter, use the largest possible value for the IGT word and the smallest value for the IGT_counter, for a given desired intermessage gap time. |

**Intermessage Gap Time Counter**

### 7.3.4 Message Block Pointer

The Message Block pointer is a 16-bit word that the user writes to point to the beginning of a 1553 message block.

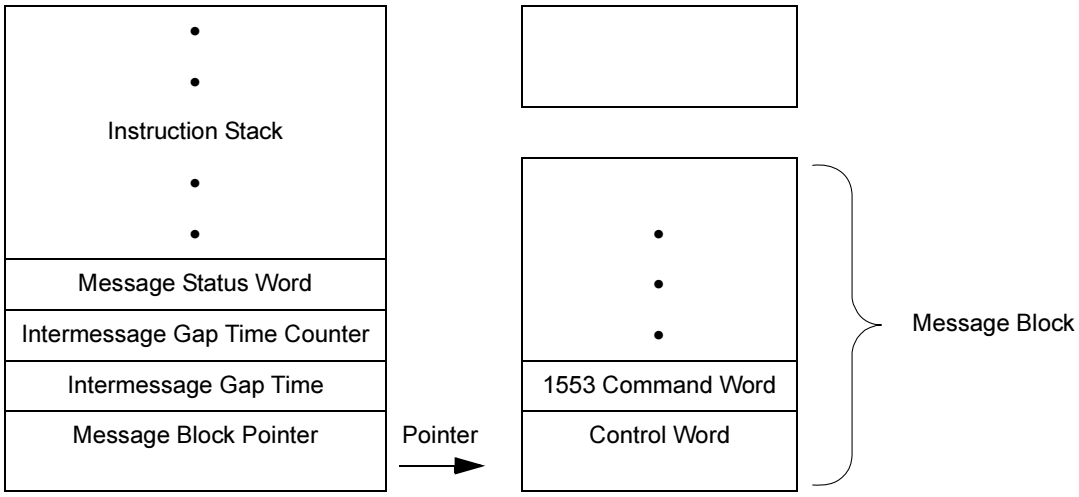



**Figure 7-3 Message Block Pointer – BC/Concurrent-RT Mode**

## 7.4 Message Block

The message block can be loaded anywhere in the Instruction Stack/Message Block area. (See **Figure 7-1 BC/Concurrent-RT Memory Map** on page 7-3.) Message blocks do not have to be stored in sequential locations in the memory since the Message Block pointers point to the message blocks in sequence.

Each block contains a 1553 message plus its Control word. This Control word is written into the first word of each block. The Control word instructs the channel which type of message to transmit (i.e., RT-to-RT, Mode Code, Broadcast, Error injection, etc.). The size of the message block is variable and depends on the size of the message itself.

The descriptions of the various message block formats (i.e., BC-to-RT, RT-to-BC and RT-to-RT) are illustrated in section **7.4.1 Message Block Formats** on page 7-8.

For a description of each bit see **7.4.2 Control Word** on page 7-10.

### 7.4.1 Message Block Formats

The Message block contains, or will contain after response from an RT, the entire 1553 message as it appears on the 1553 bus, including Command Word(s), Data Word(s), and Status Word(s). Examples of Message block formats are:

**Example No. 1: Transmit Command Operating as BC Only**

**Block *before* execution**

| | |
|---|---|
| | |
| | |
| 1553 Transmit Command | |
| Control Word | First Location in Block |

**Block after execution**

| | |
|---|---|
| 1553 Data Word | First Transmitting Remote Terminal (not simulated) |
| • | • |
| • | • |
| 1553 Data Word | • |
| RT Status Word | First Transmitting Remote Terminal (not simulated) |
| 1553 Transmit Command | |
| Control Word | First Location in Block |

**Example No. 2: Receive Command Operating as Both BC and Receiving RT**

**Block *before* execution**

| | |
|---|---|
| RT Status Word | Simulated by board (Loaded by user) |
| 1553 Data Word | • |
| • | • |
| • | • |
| 1553 Data Word | Simulated by board (Loaded by user) |
| 1553 Receive Command | |
| Control Word | First Location in Block |

**Block *after* execution**

| | |
|---|---|
| RT Status Word | |
| 1553 Data Word | |
| • | |
| • | |
| 1553 Data Word | |
| 1553 Receive Command | |
| Control Word | First Location in Block |

**Example No. 3: RT-to-RT Command Operating as BC and Receiving RT**

**Block before execution**

| | |
|---|---|
| (Receive) RT Status Word | Simulated by board (Loaded by user) |
| Leave empty for Data +N | |
| • | |
| • | |
| Leave empty for Data #1 | |
| Leave empty for (transmit) RT Status Word | |
| 1553 Transmit Command | |
| 1553 Receive Command | |
| Control Word | First Location in Block |

**Block after execution**

| | |
|---|---|
| (Receive) RT Status Word | |
| 1553 Data Word | From transmitting Remote Terminal (not simulated) |
| • | • |
| • | • |
| 1553 Data Word | • |
| (Transmit) RT Status Word | From transmitting Remote Terminal (not simulated) |
| 1553 Transmit Command | |
| 1553 Receive Command | |
| Control Word | First Location in Block |

**Figure 7-4 Message Block Formats**

### 7.4.2 Control Word

Logic 1 enables the function, 0 disables the function.

| Bit | Bit Name | Description |
|---|---|---|
| **15** | **Stop On Error** | Message error stops BC operation. Restart by writing to the Instruction Counter register and issue a Start command. |
| **14** | **Parity Error** | Selects Even parity in 1553 Word. Also set Bit 08 |
| **13** | **Halt/Continue** | 1 = Halt; stops BC transfer operation.<br>0 = Run or Continue. |
| **12** | **Word Count Error** | Transmits fewer or more words than are indicated by the Word Count field. (See **7.12.14 Word Count Register** on page 7-23. This function is valid for BC-to-RT messages only.) |
| **11** | **Bit Errors** | Transmits invalid number of bits or invalid Zero Cross bit in 1553 Words. (See **6.10.14 Bit Count Register** on page 6-25.) |
| **10** | **Incorrect Sync** | Transmits incorrect Sync. Data type Sync is transmitted in the Command Word. Also set Bit 08 |
| **09** | **Non-Contiguous Data** | Transmits the first 1553 Data Word with an invalid Gap Time (between Command and Data Word). Also set Bit 08 |
| **08** | **Error Placement/ Error Injection Enable** | Bit 08 applies to Parity, Sync and Bit Count error injection placement for BC-to-RT, Broadcast Receive, and Mode Code receive with Data messages:<br>0 = Inject error in Command Word<br>1 = Inject error in Data Words<br>**Note:**<br>• For other message types<br>0 = Disable error injection<br>1 = Enable error injection<br>• For Word Count Error. WCerror needs only Bit 12 to be set. The firmware does not look at Bit 08. |
| **07** | **Bus A/B** | Selects active 1553 bus: logic 1 selects bus A; logic 0 selects bus B. |
| **06** | **Auto Bus Switch** | On error, the BC will retry message transfer on alternate bus<br>Auto-retry must be selected |
| **04 – 05** | **Auto Retry Code** | On error, selects the number of retries before transferring the next message:<br>**Bit 05** / **Bit 04** / **Description**<br>0 / 0 / No Retries<br>0 / 1 / 1 Retry<br>1 / 0 / 2 Retries<br>1 / 1 / 3 Retries |
| **00 – 03** | **Command Code** | **03** / **02** / **01** / **00** / **Description**<br>0 / 0 / 0 / 0 / Transmit Command (RT to BC)<br>0 / 0 / 0 / 1 / Receive Command (BC to RT)<br>0 / 0 / 1 / 0 / RT-to-RT Command<br>0 / 0 / 1 / 1 / Mode Code<br>0 / 1 / 0 / 0 / Broadcast Receive Command<br>0 / 1 / 0 / 1 / Broadcast RT-to-RT Command<br>0 / 1 / 1 / 0 / Broadcast Mode Code<br>0 / 1 / 1 / 1 / Skip Message [1]<br>1 / 0 / 0 / 0 / Jump Command [2]<br>1 / 1 / 1 / 1 / Minor Frame Command [3] |

**BC/Concurrent-RT Control Word**

1. See **7.4.4 Skip Message** on page 7-11
2. See **7.4.5 Jump Command Operation** on page 7-11
3. See **7.5 Minor Frame Operation** on page 7-12

### 7.4.3 Halt Operation

Normally set the Halt Operation bit to logic 0 before writing to the Start register. In realtime (during BC execution), the user sets this bit to logic 1. When operating on that particular message block's Control word, the channel will halt transfer operations until the bit is reset to logic 0.

When the channel detects that the Halt bit is set, it sets the Wait For Continue bit in the Message Status register. (See **7.12.10 Message Status Register** on page 7-20.) Use the Wait For Continue bit to find out when the channel has arrived at the halted instruction block. When the channel detects that the Halt bit (Continue mode) has been reset, the channel will reset the Wait For Continue bit in the Message Status register and continue BC operation.

The Halt operation can be implemented only in message blocks that have *not* yet been executed by the channel.

**Note:** The Halt operation can be used in conjunction with the Jump command. See **7.4.5 Jump Command Operation** on page 7-11.

### 7.4.4 Skip Message

The Skip Message command allows the user to skip a message defined in a certain message block. To do so, modify the Command field in the Control Word. This lets the user selectively send a message in the current frame. The 'skip' takes place immediately and does not wait until the Intermessage Gap Time expires.

### 7.4.5 Jump Command Operation

The channel's BC transfer cycle can be modified by setting the Jump command in the BC Control word. The Jump command instructs the channel to operate on a new instruction stack or new stack entry in the same stack. This Control word is followed by a Stack Pointer word instead of the usual 1553 Command Word. In addition, the stack pointer is followed by an Instruction Count value. The Jump command is tested *after* the channel has tested the Halt/Continue bit in the Control word. The 'jump' takes place immediately and does not wait until the Intermessage Gap Time expires.

The memory structure of the jump command is illustrated below.

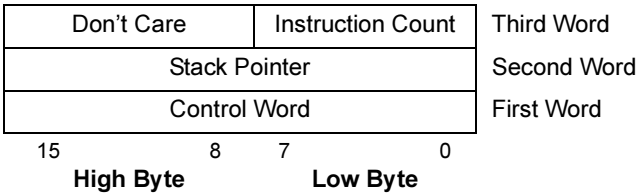



**Figure 7-5 Jump Command Message Block Structure**

## 7.5 Minor Frame Operation

The Minor Frame type of message can be used in the following ways:

- To function as a "delay time" between groups of messages.
- To produce a list of messages that will be sent out over the bus at different frequencies.

Minor frame time is defined as the time elapsed from the beginning of a minor frame to beginning of the next minor frame. To set up minor frame operation, each minor frame must begin with a minor frame command. (See **7.12.20 Minor Frame Time Register** and **7.12.21 Minor Frame Time Multiplier Register** on page 7-25.) The maximum value possible for the Minor Frame Time is 800 milliseconds.

**Example:** **Figure 7-6** shows a configuration of four minor frames, in which Message A is sent in every frame, Message B is sent in every other frame, and Message C is sent once. Each minor frame goes out at 10 msec. (100Hz). If each minor frame is 10 msec. long, Message A is sent every 10 msec., Message B is sent every 20 msec., and Message C is sent every 40 msec.

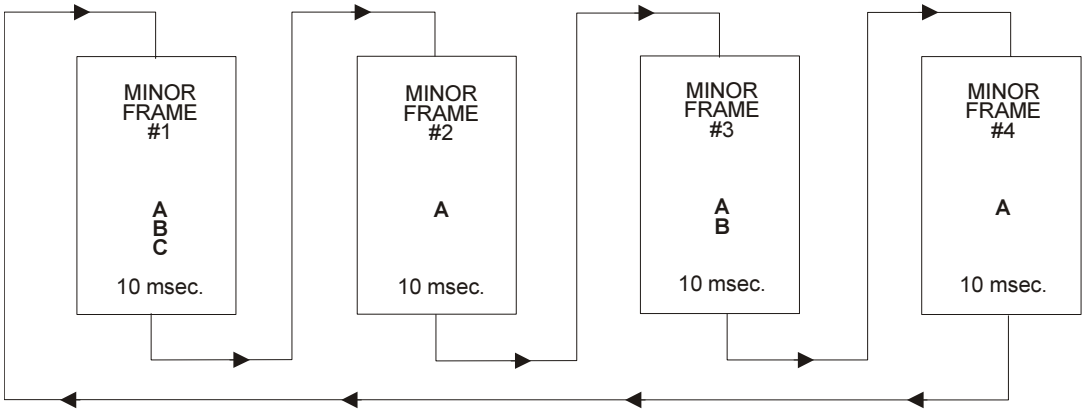



**Figure 7-6 Minor Frame Sequencing**

**Notes:**

1. The MINOR_FRAME message does not appear as a real message on the data bus.
2. Frame Time should not exceed the total time of all the minor frames in the minor frame sequence. (See **7.12.21 Frame Time Multiplier Register** on page 7-24.)

## 7.6 Asynchronous Frame Operation

During standard operation, the channel sets up a frame of messages and then sends them out synchronously over the bus. The user can set up multiple frames of messages, and select which one to send out.

Asynchronous Frame operation allows the user to transmit a frame asynchronously. This means that in the middle of the transmission of the messages of a frame (frame 1), another frame (frame 2) can be transmitted, and then the channel returns to continue transmitting the messages of the original (or synchronous) frame (frame 1).

To transmit an asynchronous frame, the user must write the number of messages

in the asynchronous frame into the Asynchronous Message Count register, place a pointer to the beginning of the asynchronous frame in the Asynchronous Frame Pointer register, and then set the Asynchronous Start Flag register to a non-zero value. This will send out the asynchronous frame over the bus. (See **7.12.44 Asynchronous Start Flag Register**, **7.12.45 Asynchronous Frame Pointer Register** and **7.12.46 Asynchronous Message Count Register** on page 7-32.)

## 7.7 Remote Terminal Simulation

**Not for single function channel (*PxS*)**

When the channel is simulating both the Bus Controller and one or more Remote Terminals, the user must write the simulated Remote Terminal 1553 Status Word and Data Word(s) into the message block in the sequence in which they are to be transmitted over the 1553 bus. (See **7.4.1 Message Block Formats** on page 7-8.)

**Note:**

- The rules for the 1553 RT Status Word **do not** apply when simulating an RT: the user must provide the message Status Word; insert the 1760 header in the data; error injections and interrupts related to RTs are *not* available. (See **6.10.43 1553 RT Status Word Table** on page 6-37.)
- The **Service Request bit (SRQ)** is not supported in RT simulation.
- On a single function channel (*PxS*) you cannot have a Concurrent RT when the channel is in BC mode.

To indicate to the channel which Remote Terminals are to be simulated, write to the 32-byte Active Remote Terminal table. Each entry in the 32-byte table corresponds to a specific Remote Terminal.

The first byte is for RT #0, the second is for RT #1, and the last byte is for RT #31 (for a total of 32 locations). A table entry value of 1 enables the Remote Terminal simulation by the channel; a value of 0 disables the simulation by the channel.

| | |
|---|---|
| RT#31 RT Settings byte | 341F H |
| RT#30 RT Settings byte | |
| • • • | • • • |
| RT#0 RT Settings byte | 3400 H |

**Figure 7-7 RT Settings Table – BC/Concurrent RT mode**

| Bit | Description |
|---|---|
| **01 – 07** | Reserved – set to 0 |
| **00** | 1 = RT Active<br>0 = RT Not Active |

**RT Settings Byte Definition – BC/Concurrent RT mode**

## 7.8 Continuous or One-Shot Message Transfers

The channel can transfer all programmed messages once, in a continuous loop, or for *n* number of times.

**One-shot mode** In One-Shot mode, after receiving a Start command, the channel transfers all messages, sets the Message Complete bit in the Message Status register, issues an interrupt (if programmed), turns off Bit 00 of the Start register, and waits for a new Start command. Use the Start Register to select One-Shot mode. See **7.12.8 Start Register** on page 7-18.

**n-Times mode** In n-Times mode, load the Loop Count register with the number of times to transmit the messages (frame) and set the Loop and Start bits in the Start register. The user can transmit messages from 1 to 255 times. (See **7.12.8 Start Register** on page 7-18 and **7.12.12 Loop Count Register** on page 7-21.) The Frame Time registers, on page 7-25, determine the time between frames. See also **7.8.1 Frame Time Calculations** on page 7-14.

**Continuous Loop mode** In Continuous Loop mode, the channel will retransmit the message frame at a predetermined, user-programmable rate. Use the Start register and the Loop Count register, to select Continuous Loop mode. (See **7.12.8 Start Register** on page 7-18 and **7.12.12 Loop Count Register** on page 7-21.) In Continuous Loop mode, all messages relating to the (active) Stack pointer and Instruction counter are continuously looped until you halt the channel's operation by clearing Bit 00 of the Start register. See also **7.8.1 Frame Time Calculations** on page 7-14.

### 7.8.1 Frame Time Calculations

The Frame Time is a function of two control registers, the Frame Time Multiplier register and the Frame Time Resolution register. The internal Frame Time is loaded when a Start command is received. After all instructions are executed (1 frame), the channel waits until the internal Frame Time counts down to 0 before reloading the Frame Time and transmits the next frame.

**Note:** If the Frame Time is less than the time required to transmit all messages within 1 frame, the subsequent frames will be transmitted with the minimum delay between them. The minimum delay is approximately 20 μsec, measured as dead time on the bus.

The channel reads the **Frame Time Resolution Register**, multiplies it by the **Frame Time Multiplier Register**, and uses the product as the maximum number of 'clock ticks' to wait per frame. Each clock tick is 155 nanoseconds.

The maximum value of the Frame Time Resolution register is FFFF H (65535) 'clock ticks', which is equivalent to a frame time of:

FFFF (65535) × 155 nanoseconds per clock tick = 10158 microseconds

To enter a frame time up to 10158 microseconds, the Frame Time Multiplier register is 0 and

$$\text{desired resolution register} = \frac{(\text{desired time in microseconds})*1000}{155}$$

To enter a Frame Time that is greater than 10158 microseconds, use the

multiplier as well. For the greatest accuracy, the Frame Time Multiplier should have the minimum possible value.

The following algorithm first calculates the minimum Frame Time Multiplier and finds the appropriate resolution to obtain the desired frame time.

$$\text{frametime\_multiplier} = \frac{\text{desired time in microseconds}}{\text{10158 microseconds}}$$

$$\text{frametime\_resolution} = \left( \frac{\text{desired time in microseconds}}{\text{frametime\_ multiplier + 1}} \text{ X } 1000 \right) / 155$$

**Example:** To calculate a Frame Time of 500 msec.

500 milliseconds = 500000 microseconds

$$\text{frametime\_multiplier} = \frac{500000}{10158} = 49$$

$$\text{frametime\_resolution} = \left( \left( \frac{500000}{49 + 1} \text{ X } 1000 \right) / 155 \right) = 64516$$

Frame Time Resolution = 64516 Dec (FC04 H)

Frame Time Multiplier = 50 Dec (0031 H)

Before issuing the Start command, set:

- The Frame Time Multiplier register to 0031 H
- The Frame Time Resolution register to FC04 H

For descriptions of these registers see **7.12.17 Frame Time Multiplier Register** on page 7-24 and **7.12.18 Frame Time Resolution Register** on page 7-24.

**Note:** The maximum frame time that the channel can handle is 2.1 seconds (2,100,000 microseconds). To set this frame time, set the Frame Time Resolution register to FFA5 and Frame Time Multiplier register to 00CE. The minimum frame time is 0 microseconds.

## 7.9 Mode Codes

The channel handles all dual-redundant 1553B Mode Codes; the Word Count field is decoded according to MIL-STD-1553B. The channel does not implement the two Quad-redundant Mode Codes, Selected transmitter Shutdown and Override Selected transmitter Shutdown.

## 7.10 Service Request (SRQ) Processing

The SRQ bit is set by an RT in the 1553 RT Status Word (see **6.10.43 1553 RT Status Word Table** on page 6-37). Setting the SRQ bit indicates to the BC that the RT/ Subaddress requires servicing.

The BC provides the following service:

1. The channel will send out a Mode Code 16 (transmit Vector Word) to get the Vector Word from the RT which contains more information about what needs service. This message is stored in SRQ Message 1 (3F80 – 3F87 H). See **7.12.29 SRQ Message 1 Register** on page 7-27.

2. The BC will then build and send out a transmit message (RT-to-BC) to this RT, with the Subaddress and Word Count as indicated in the corresponding bit positions of the Vector Word. This message is stored in SRQ Message 2 (3F88 – 3FCD H). See **7.12.28 SRQ Message 2 Register** on page 7-27 and **6.10.35 1553 RT Vector Word Table** on page 6-33.
3. If the interrupt SRQ message bit is set in the Interrupt Condition register (see **7.12.9 Interrupt Condition Register** on page 7-19) an interrupt will be generated when the BC completes steps 1 and 2.
4. The SRQ Counter is incremented by 1. See **7.12.26 SRQ Counter** on page 7-27.

To disable Service Request processing set Bit 01 in the **7.12.38 BC Protocol Options Register** on page 7-30.

## 7.11 1760 Option

### 7.11.1 Header Word

In the MIL-STD-1760 specification, the first Data Word of a message may be a Header Word, which is used for message identification. The Header Word is associated with a specific Subaddress.

To indicate that a specific subaddress will require a Header Word, set the corresponding entry in the 1760 Header Exist table to 1. Then set the corresponding entry in the 1760 Header Transmit/Receive Value table to the value you expect to receive in the first Data Word of the message. The Header value expected is either the predefined 1760 value, which is the default channel setting, or another value the user enters in the 1760 Header Value Transmit/ Receive Table.

1760 Header is not supported for RT-to-RT messages.

See **7.12.30 1760 Header Value Transmit Table**, **7.12.31 1760 Header Value Receive Table** and **7.12.32 1760 Header Exist Table** on page 7-28.

### 7.11.2 Checksum

The 1760 option implements checksum generation and checksum error detection capabilities. Checksums are calculated as each Data Word is sent or received. If the checksum flag is set on an outgoing message, the checksum will be sent in place of the last Data Word. On an incoming message, the calculated checksum is checked against the last Data Word received. If it does not match, the Checksum Error bit is set in the Message Status Word.

The user gets to select, per message to generate or receive Checksum in the Intermessage Gap Time Counter. See **7.3.3 Intermessage Gap Time Counter/Message Function Select** on page 7-6.

If Checksum is selected the user may also request that the Checksum be sent out with an incorrect value as an error injection mechanism. See **7.3.3 Intermessage Gap Time Counter/Message Function Select** on page 7-6.

**Note:** For an RT-to-BC message where the RT is active, Checksums and headers are sent by specifying the header or the checksum in the message data. In this way header or checksum errors can be injected directly by the user.

## 7.12 Control Register Definitions

### 7.12.1 Instruction Stack/ Message Block Area

**Address: 7100 – 7FFF (H)**

This area is available to the programmer for Instruction Stacks and Message Blocks.

### 7.12.2 Time Tag

**Address: 7008 – 700B (H)**

**Read only**

The Time Tag is a free-running 32-bit counter on the channel. The Time Tag is reset to 0 upon a power up or a software reset and starts counting. When it reaches the value FFFF FFFF (H), the counter wraps around to 0 and continues counting. To re-initialize to 0, write to the Time Tag Reset register.

The user may read the Time Tag counter at any time. Read the two 16-bit words of the Time Tag counter value sequentially, first Lo word, then Hi word.

**The counter must be read in the following sequence:**

1. Read 7008 H – Lo word (16 bit, read only)
2. Read 700A H – Hi word (16 bit, read only)

The Time Tag resolution is 4 μsec.

**To calculate elapsed time between Time Tags:**

**Example:**

1. Calculate difference between Time Tags:
   150 (Time Tag 2) – 50 (Time Tag 1) = 100
2. Elapsed time:
   100 × 4 = 400 μsec

### 7.12.3 Time Tag Reset Register

**Address: 700 (H)**

**Write only**

Write to the Time Tag Reset register to reset the channel's Time Tag Counter (data field = don't care). Immediately after the reset, the counter will start to count from 0.

**Note:** The counter may also be reset from an external source (*EXC-1553UNET/Px* only). See **4.5.5 External Signals Connector Pinouts** on page 4-10.

### 7.12.4 Channel Reset Register

**Address: 700 (H)**

Write any value to the Channel Reset register to reset the channel.

Channel Reset erases all locations in the dual-port RAM. Channel status, channel ID and Firmware Revision registers are written by the channel after the reset operation is completed.

### 7.12.5 Channel Configuration Register

**Address: 3FF (H)**

Before issuing a Start command to the channel, set the operating mode of the channel via the Channel Configuration register.

To modify the Channel Configuration register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

| Hex Value | Operating Mode |
|---|---|
| **04** | BC/Concurrent-RT |

**Channel Configuration Register Value: BC/RT Mode**

### 7.12.6 Channel ID Register

**Address: 3FF (H)**

The Channel ID register contains a fixed value that can be read by your initialization function to detect the presence of the channel. The one-byte value of this register is 45 (H), ASCII value E.

### 7.12.7 Channel Status Register

**Address: 3FF (H)**

The Channel Status register indicates the status of the channel. In addition, this register indicates which options have been selected, as described below. Do not modify this register. Status bits are active if set to 1.

| Bit | Description |
|---|---|
| **07** | 1 = Always set |
| **05 – 06** | Indeterminate |
| **04** | 1 = Channel Halted<br>0 = Channel Running |
| **03** | 1 = Self-Test OK |
| **02** | 1 = Timers OK |
| **01** | 1 = RAM OK |
| **00** | 1 = Channel Ready |

**Channel Status Register**

**Note:** Channel operation stops after the Start bit is cleared in the Start register. Following this, the channel sets Bit 04 (Channel Halted). Certain registers may be modified only after the Channel Halted bit has been set. After receiving a subsequent Start command (by writing to the Start register), the channel resets the Channel Halted bit. The condition of this bit after power-up or software reset is logic 1.

### 7.12.8 Start Register

**Address: 3FF (H)**

The Start register controls the Start/Stop operation of the channel. Writing the appropriate bit (Bit 00) to the Start register starts the Bus Controller transfer operation. When operating in Continuous Loop or *n*-Times mode, the user must set the Start and Loop bits in the Start register. The Loop and *n*-Times number

are selected via the Loop Count register. In the One-Shot and *n*-Times modes, the channel resets the Start bit in the register after *all* messages have been transferred. The channel does not reset any bit while in Continuous Loop mode. To halt the Loop operation between messages, set Bit 00 to 0. In order to halt the operation at the end of the entire frame, set Bit 02 to 0 (Bit 02 is not tested between message transfers). Related data Bit 04 in the Channel Status register indicates when the channel has been halted. (See **7.12.7 Channel Status Register** on page 7-18.)

| Bit | Description |
|---|---|
| **03 – 07** | 0 |
| **02** | 1 = Loop mode<br>0 = One-Shot mode |
| **01** | 0 |
| **00** | 1 = Start Operation<br>0 = Stop |

**Start Register**

**Note:** You can start the module externally by sending a minimum LVTTL pulse of 100 nsec. to the EXSTARTn pin. See **4.5.5 External Signals Connector Pinouts** on page 4-10.

### 7.12.9 Interrupt Condition Register

**Address: 3FF (H)**

The Interrupt Condition register allows the user to set interrupt triggers. When a condition occurs that is enabled in this register, an interrupt is generated. A logic 1 enables the interrupt condition. To determine which condition caused the interrupt, check the Message Status register.

The Interrupt Condition register must be set before issuing a Start command to the channel. To modify the Interrupt Condition register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

| Bit | Description |
|---|---|
| **06 – 07** | 0 |
| **05** | SRQ message |
| **04** | End Minor Frame |
| **03** | Message Error |
| **02** | End of Frame |
| **01** | Message Complete |
| **00** | 0 |

**Interrupt Condition Register**

**Note:** The interrupt will be sent at the end of the message for all interrupt conditions. When an interrupt is configured for an End of Frame or End Minor Frame, the interrupt pulse occurs immediately after the last message transmission in the frame/minor frame is complete.

### 7.12.10 Message Status Register

**Address: 3FF (H)**

The Message Status register indicates the status of the current message being processed. The definition of each status bit is given below. Logic 1 indicates that the condition is activated.

| Bit | Bit Name | Description |
|---|---|---|
| **05 – 07** | Reserved | Set to 0 |
| **04** | End Minor Frame | The last word in the last message in the Minor Frame has been sent. |
| **03** | Message Error | The message has been sent. As a result, the Error bit has been set in the Message Status Word. |
| **02** | End Of Frame | The last word of the last message in the frame has been sent. |
| **01** | Message Complete | The last word of the message has been sent. |
| **00** | Wait For Continue | A message with the Halt bit set has been encountered. Reset the Halt bit in the Control word to continue. |

**Message Status Register**

**Note:** Status bits are *not* reset by the channel. After reading them, the user must reset them.

### 7.12.11 RT Response Time Register

**Address: 3FF (H)**

The RT Response Time register sets the Response Time of the Remote Terminals being simulated by the channel. The resolution of the Response Time register is 155 nsec. per bit. The minimum time is approximately 4 μsec., which is achieved by writing a 0 to this register. Any value above zero will result in a Response Time equal to 4 μsec. plus the contents of the register x 155 nsec. The actual response time has a tolerance of ± 1 μsec.

The Response Time register must be set before issuing a Start command to the channel. To modify the RT Response Time register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18).

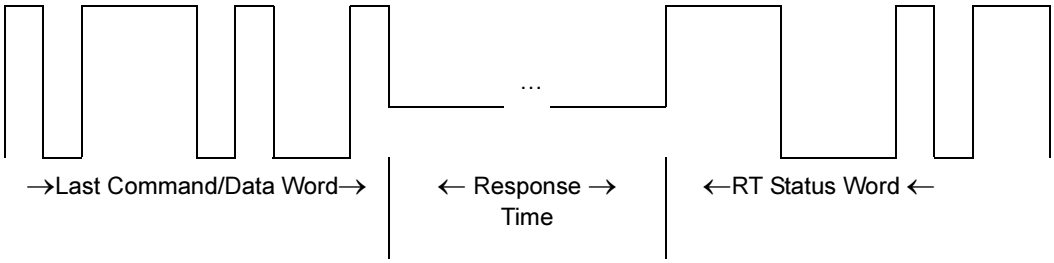



**Figure 7-8 RT Response Time Definition**

**Example:** To request a Response time of 9 μsec:
Write 32 to the RT Response Time register
$32 \times 0.155 \cong 5$ μsec + 4 μsec = 9 μsec

### 7.12.12 Loop Count Register

**Address: 3FF (H)**

The Loop Count register is used in conjunction with the Loop bit in the Start register. If the Loop bit in the Start register is set, then set the Loop Count register to specify the number of times the Message frame will be transmitted. A value of zero is interpreted as a request for continuous looping.

The Loop Counter register must be set before issuing a Start command to the channel. To modify the Loop Counter register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

| Bit | Value | Description |
|---|---|---|
| **00 – 07** | **0** | Transmits in Continuous Loop |
| | **1 – 255** | Sends Message Frame n-times as defined |

**Loop Count Register**

### 7.12.13. Bit Error Register

**Address: 3FF (H)**

There are two types of bit errors, **Bit Count Error** and **Zero Crossing Error**. Both are enabled by setting the **Bit Error bit** (Bit 11), in the Control Word, see **7.4.2 Control Word** on page 7-10. The Bit Error register determines the nature of the error to be injected. If the Bit Error bit of the Control Word is not set, a valid 20-bit word is transmitted regardless of the contents of the Bit Error register.

Each of these errors can be injected into a Command Word or Data Word (see **Error Placement/Error Injection Enable bit**, (Bit 08), in the Control Word). If the error is to be injected in a Data Word, the Data Word is selected via the Error Word Index register, see **7.12.24 Error Word Index Register** on page 7-26.

**Note:** On a single function channel (*PxS*) this register is reserved. Error injection is not available.

#### 7.12.13.1 Bit Count Error

The total number of bits sent in a 1553 Word, including Sync (3) and Parity (1) is more or less than the normal 20 bits.

The bit count is selected using bits 00 – 02 of the Bit Error register.

| Bit | Description | | | |
|---|---|---|---|---|
| 00 – 02 | Bit 02 | Bit 01 | Bit 00 | Number of 1553 bits sent per word |
| | 0 | 0 | 0 | 17 (-3) |
| | 0 | 0 | 1 | 18 (-2) |
| | 0 | 1 | 0 | 19 (-1) |
| | 0 | 1 | 1 | 20 |
| | 1 | 0 | 0 | 21 (+1) |
| | 1 | 0 | 1 | 22 (+2) |
| | 1 | 1 | 0 | 23 (+3) |

**Bit Error Register Bit Count Error Settings**

#### 7.12.13.2 Zero Crossing Error

The bit pattern is altered, from normal Manchester II zero cross coding to a skewed form of the coding. Zero Crossing error will affect the bit selected via the Zero Cross Bit Index register. See **7.12.23 Zero Cross Bit Index Register** on page 7-26.

The Zero Cross coding is selected using bits 04 – 07 of the Bit Error register:

| Bit | Description | | | | |
|---|---|---|---|---|---|
| 04 – 07 | Bit 07 | Bit 06 | Bit 05 | Bit 04 | Zero crossing coding used |
| | 0 | 0 | 0 | 0 | zc at 500 nano (normal) |
| | 0 | 0 | 0 | 1 | zc at 600 nano (legal) |
| | 0 | 0 | 1 | 0 | zc at 650 nano (illegal) |
| | 0 | 0 | 1 | 1 | zc at 700 nano (illegal) |
| | 0 | 1 | 0 | 0 | zc at 400 nano (legal) |
| | 0 | 1 | 0 | 1 | zc at 350 nano (illegal) |
| | 0 | 1 | 1 | 0 | zc at 300 nano (illegal) |
| | 0 | 1 | 1 | 1 | Reserved |
| | 1 | 0 | 0 | 0 | full bit high |
| | 1 | 0 | 0 | 1 | full bit low |
| | 1 | 0 | 1 | 0 | full bit dead |
| | Other values | | | | Reserved |

**Bit Error Register Zero Crossing Error Settings**

Bit 03 of the Bit Error register is reserved.

Set the Bit Error register before issuing a Start command to the channel. To modify the Bit Error register, issue a Stop command, modify the register, and

then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

### 7.12.14 Word Count Register

**Address: 3FF (H)**

The Word Count register controls the number of 1553 Data Words (±3) in the message and allows the user to inject a Word Count error. The error is an offset relative to the 1553 Command Word Word Count field. This register is used by the Channel only for messages for which the Word Count Error bit is set in the Control word register. (See **7.4.2 Control Word** on page 7-10.) If the Word Count Error bit is not set, a correct number of words is transmitted regardless of the contents of the Word Count register.

The Word Count register must be set before issuing a Start command to the channel. To modify the Word Count register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

| Register Value | Word Count Offset |
|---|---|
| FD H | -3 Words |
| FE H | -2 Words |
| FF H | -1 Word |
| 00 H | No Error Injection |
| 01 H | +1 Word |
| 02 H | +2 Words |
| 03 H | +3 Words |

**Word Count Register Values**

**Note:** On a single function channel (*PxS*) this register is reserved. Error injection is not available.

### 7.12.15 BC Response Time Register

**Address: 3FF (H)**

The BC Response Time register sets the BC's Response Time window, whose value determines the maximum wait time until an RT's Status Response is considered invalid by the BC. The resolution of the BC Response Time register is 155 nsec. per bit, The minimum time is approximately 2 μsec.

The BC Response Time register must be set before issuing a Start command to the channel. To modify the BC Response Time register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

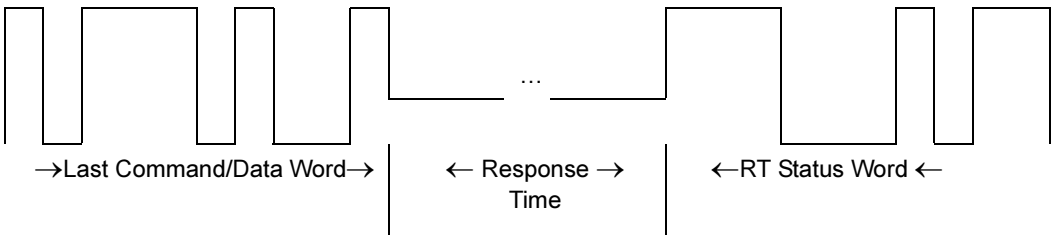



**Figure 7-9 BC Response Time Definition**

**Example:** To request a Response time of 14 μsec:
Write 90 to the BC Response Time register
$90 \times 0.155 \cong 14$ μsec

### 7.12.16 Message Stack Pointer

**Address: 3FF0 – 3FF1 (H)**

The Message Stack pointer points to the Instruction stack. The Instruction stack can reside anywhere between the locations 0000 – 33FF (H), 4000 – 6FFF (H) and 7100 – 7FFF (H).

The Stack Pointer register must be set before issuing a Start command to channel. To modify the Stack Pointer register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

### 7.12.17 Frame Time Multiplier Register

**Address: 3FEE – 3FEF (H)**

The Frame Time Multiplier register contains the 16-bit Frame Time value for Continuous and *n*-Times modes operation. The value written to the Frame Time Multiplier register is multiplied by the value set in the Frame Time Resolution register described below. The value set must equal the desired multiplication factor -1.

The Frame Time Multiplier register must be set before issuing a Start command to the channel. To modify the Frame Time Multiplier register, issue a Stop command, modify the register, then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

For information on how to use Frame Times, see **7.8.1 Frame Time Calculations** on page 7-14.

### 7.12.18 Frame Time Resolution Register

**Address: 3FEC – 3FED (H)**

The 16-bit Frame Time Resolution value represents the resolution of the Frame Time counter in increments of 155 nsec. (See **7.8 Continuous or One-Shot Message Transfers** on page 7-14.)

The Frame Time Resolution register must be set before issuing a Start command to the channel. For an example of how to calculate Frame Time Multiplier and Frame Time Resolution, see **7.8 Continuous or One-Shot Message Transfers** on page 7-14. To modify the Frame Time Resolution register, issue a Stop command, modify the register, then, issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

For information on how to use Frame Times, see **7.8.1 Frame Time Calculations** on page 7-14.

### 7.12.19 Instruction Counter

**Address: 3FEB, 3FEA (H)**

The Instruction Counter must be loaded with the number of instructions (1553 Messages) to execute in the current frame. The value must be greater than 0 before the user writes to the Start register to begin a transmission. Set the Instruction counter to 1 for one message, 2 for two messages, etc. The channel updates the Instruction counter by decrementing the value and writing it back to

memory at the end of each message transfer.

The Instruction Counter register must be set before issuing a Start command to the channel. To modify the Instruction Counter register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.) When in Continuous Loop mode, the Instruction Counter register cycles from the initial value down to 1.

The low register (3FEA H) contains the MSB; the high register (3FEB H) contains the LSB. Therefore, for an Instruction Counter less than 256, use the LSB only, address 3FEB H.

### 7.12.20 Minor Frame Time Register

**Address: 3FE8 – 3FE9 (H)**

**Write only**

This 16-bit Minor Frame Time register is used to set the length of a single minor frame. (See **7.5 Minor Frame Operation** on page 7-12.) The resolution of the Minor Frame Time register is 1 μsec. per bit. The maximum value is approximately 65 msec., which can be extended by the multiplier set in the Minor Frame Time Multiplier register.

The Minor Frame Time register must be set before issuing a Start command to the channel. To modify the Minor Frame Time register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

### 7.12.21 Minor Frame Time Multiplier Register

**Address: 3FE6 – 3FE7 (H)**

**Write only**

The Minor Frame Time Multiplier register is a multiplier of the Minor Frame Time register described above. The value written by the user to the Minor Frame Time Multiplier register allows the user to extend the Minor Frame Time beyond the 65 msec. maximum in the Minor Frame Time register. The maximum Minor Frame Time Multiplier is 255. The maximum Minor Frame Time possible using both registers is approximately 800 milliseconds.

**Example:** To generate a Minor Frame Time of 1 sec., set the Minor Frame Time register to F424 H (62,500 Dec.), and set the Minor Frame Time Multiplier register to 10 H (16 Dec.).

The Minor Frame Time Multiplier register must be set before issuing a Start command to the channel. To modify the Minor Frame Time Multiplier register, issue a Stop command, modify the register, and then issue a Start command. (See **7.12.8 Start Register** on page 7-18.)

### 7.12.22 Replay Register

**Address: 3FE4 – 3FE5 (H)**

Regular message processing is accomplished using Intermessage Time Gaps between messages. The channel waits between messages as per the intermessage gap time.

With the Replay register, the user sets the *absolute time* at which each message is to be transmitted over the bus. The Bus Controller compares this absolute time with its internal Time Tag to determine if the time has come to transmit the message over the bus. Generally the user will reset the BC's Time Tag just prior to running and will select time absolute time 0 for the first message to be transmitted.

For example, the user may request that the first message be transmitted at time 1000 microseconds, the second message at 2000 microseconds and so on.

This timing method is useful for replaying a previously recorded scenario based on the recorded Time Tags.

The absolute time is set by the user in units of 32 microseconds.

| Bit | Description |
|---|---|
| **01 – 15** | Reserved |
| **00** | 1 = Replay mode<br>0 = Intermessage gap mode |

**Replay register**

### 7.12.23 Zero Cross Bit Index Register

**Address: 3FDC – 3FDD (H)**

When a zero cross error is injected into a Message (see **7.12.13 Bit Error Register** on page 7-21 ), this register determines which bit within the word will contain the error. A zero in this register will cause the error to be injected into the first Data Word, a 1 into the second bit etc. Valid values for this register are 0 through 15.

**Note:** On a single function channel (*PxS*) this register is reserved. Error injection is not available.

### 7.12.24 Error Word Index Register

**Address: 3FDA – 3FDB (H)**

When a data error is injected into a Message (see **7.4.2 Control Word** on page 7-10), this register determines which error word will contain the error. A zero in this register will cause the error to be injected into the first Data Word, a 1 into the second word etc. If the number is greater than the number of words in the message, no error will be injected.

**Note:** On a single function channel (*PxS*) this register is reserved. Error injection is not available.

### 7.12.25 Sync Pattern Register

**Address: 3FD8 – 3FD9 (H)**

The low six bits of this register each represent a half bit time for use in sync error error injection. (See Bit 10 of the Control Word, **7.4.2 Control Word** on page 7-10.) When the sync error is requested and the BC Protocol register Bit 02 is set, the sync will be transmitted according to the pattern set in this register. A 0023 H in this register (100011 binary) would be sent as one bit times high, three bit times low and two bit times high.

See **7.12.38 BC Protocol Options Register** on page 7-30.

**Note:** On a single function channel (*PxS*) this register is reserved. Error injection is not available.

### 7.12.26 SRQ Counter

**Address: 3FD2 – 3FD3 (H)**

The SRQ counter contains the number of messages processed since BC was run.

See **7.10 Service Request (SRQ) Processing** on page 7-15.

### 7.12.27 SRQ Message Status Register

**Address: 3FCE – 3FD1 (H)**

The SRQ Message Status Register contains the SRQ Message 1 at 3FCE (H) and SRQ Message 2 at 3FD1H).

See **7.10 Service Request (SRQ) Processing** on page 7-15.

### 7.12.28 SRQ Message 2 Register

**Address: 3F88 – 3FCD (H)**

The SRQ Message 2 register contains the Transmit message sent to the Subaddress identified in the Vector word received from SRQ processing. The message consists of:

| Control Word | Command Word | 1553 Status Word | Up to 32 Data Words |
|---|---|---|---|

See **7.10 Service Request (SRQ) Processing** on page 7-15.

### 7.12.29 SRQ Message 1 Register

**Address: 3F80 – 3F87 (H)**

The SRQ Message 1 register contains the Transmit Vector Mode Code message sent out by the BC in response to SRQ by an RT. The message consists of:

| Control Word | Command Word | 1553 Status Word | Vector Word |
|---|---|---|---|

See **7.10 Service Request (SRQ) Processing** on page 7-15.

### 7.12.30 1760 Header Value Transmit Table

**Address: 3F40 – 3F7F (H)**

**1760 Option only**

Write to the 1760 Header Value Transmit table to set the expected value of the first Data Word in a RT-to-BC message. The channel checks that the specified Header receive value was received. In addition, the Internal Concurrent monitor checks that the specified header value was received. If the wrong data was sent, the **1760 Header Error** bit is set in the Message Status Word. See **7.3.1 Message Status Word** on page 7-5.

The 1760 option provides predefined values, and these are preset on each channel. The user can change the preset values.

| Transmit Subaddress | Header Value | Address |
|---|---|---|
| **1** | **0421 H** | 3F42 H |
| **11** | **0420 H** | 3F56 H |
| **14** | **0423 H** | 3F5C H |

**Predefined 1760 Transmit Header Values**

### 7.12.31 1760 Header Value Receive Table

**Address: 3F00 – 3F3F H**

**1760 Option only**

Write to the 1760 Header Value Receive table to set the expected value of the first Data Word in a BC-to-RT message. If the wrong data was sent, the Internal Concurrent Monitor will set an error bit. See Bit 06 of the Message Status Word (**7.3.1 Message Status Word** on page 7-5).

The 1760 option provides predefined values, and these are preset on each channel. The user can change the preset values.

| Receive Subaddress | Header Value | Address |
|---|---|---|
| **11** | **0400 H** | 3F16 H |
| **14** | **0422 H** | 3F1C H |

**Predefined 1760 Receive Header Values**

### 7.12.32 1760 Header Exist Table

**Address: 3EC0 – 3EFF (H)**

**1760 Option only**

The 1760 Header Exist table contains 32 entries corresponding to 32 RT subaddresses. Each entry may be set to indicate whether, or not, the channel should expect a header word for RT-to-BC or RT-to-RT messages directed to that Subaddress.

For those Header Value Table entries for which MIL-STD-1760 provides predefined values, the corresponding Header Exist Table entries are preset on each channel.

To set other values, enable the Header Exist Table entry for this RT (set it to 1) and write the value to the Header Value (Transmit/Receive) Table.

| Bit | Description |
|---|---|
| **09-15** | Reserved |
| **08** | 1 = Channel should expect a Header word in a transmit message (RT-to-BC or RT-to-RT)<br>0 = Channel should **not** expect a Header word in a transmit message |
| **01 – 07** | Reserved |
| **00** | 1 = Channel should expect a Header word in a receive message (BC-to-RT)<br>0 = Channel should **not** expect a Header word in a receive message |

**1760 Header Exist Table**

| Subaddress | Header Value | Address |
|---|---|---|
| **1** | **0100 H** | 3EC2 H |
| **11** | **0101 H** | 3ED6 H |
| **14** | **0101 H** | 3EDC H |

**Predefined 1760 Headers**

### 7.12.33 Channel Time Register Lo & Hi

**Address: 3EA4 – 3EA5 (H)**
**3EA2 – 3EA3 (H)**

This register holds the channel time value, which is stored in non-volatile flash memory and loaded at power-up. This value can be modified by calling the Set_ModuleTime_Px function. (See the *1553Px Family Software Tools Programmer's Reference*.) The factory default value is FFFF FFFF (H).

### 7.12.34 Serial Number Register

**Address: 3EA0 – 3EA1 (H)**

This register holds the board's serial number, which is stored in non-volatile flash memory and loaded at power-up. The value is binary coded. For example, a value of 1234 (H) represents the serial number 4660 (decimal).

### 7.12.35 Error Counter Lo & Hi

**Address: 3E9E – 3E9F (H)**
**3E9C – 3E9D (H)**

Error Counter is a running 32-bit counter of message errors. The counter counts retries separately.

### 7.12.36 Message Counter Lo & Hi

**Address: 3E9A – 3E9B (H)**
**3E98 – 3E99 (H)**

Message Counter is a running 32-bit counter of all messages received. The counter counts retries separately.

### 7.12.37 Channel Function Register

**Address: 3E8E – 3E8F (H)**

Set Bit 00 of the Channel Function register to 1 to expand the Message Block Area. When the Message Block Area is expanded, there is no Concurrent Monitor. Note that in most cases it is not recommended to use Expanded Block mode in BC/Concurrent-RT mode, since the standard Message Block Area is generally sufficient.

### 7.12.38 BC Protocol Options Register

**Address: 3E8C – 3E8D (H)**

| Bit | Description |
|---|---|
| **04** | 1 = Adds intermessage gap time when skipping a message. See **7.4.4 Skip Message** on page 7-11.<br>0 = Does not add intermessage gap time when skipping a message |
| **03** | 1 = 250 nsec intermessage gap time resolution<br>0 = 1 µsec intermessage gap time resolution (default) |
| **02** | 1 = Enable Sync Pattern Error injection. If this bit is set, Sync Errors are injected based on the Sync Pattern register. (See **7.12.25 Sync Pattern Register** on page 7-26.)<br>0 = Disable Sync Pattern Error injection. Sync Errors in Command Words will cause Data Sync to be sent and Sync Error in Data Words will cause Command Sync to be sent. |
| **01** | 1 = Disable SRQ processing. If this bit is set, SRQ bit in the RT Status Word is ignored. See **7.10 Service Request (SRQ) Processing** on page 7-15.<br>0 = Enable SRQ processing |
| **00** | 1 = Simulate MIL-STD-1553A protocol. If set to 1553A protocol, Mode Codes are assumed not to have any data.<br>0 = Simulate MIL-STD-1553A protocol |

**More Channel Options Register**

### 7.12.39 Send Time Tag on Sync Register

**Address: 3E8A – 3E8B (H)**

Set the Send Time Tag on Sync register to indicate that the channel should send the current Time Tag value (with a resolution of 64 µsec.) as the 16-bit Data Word upon transmitting a Mode Code 17 message (synchronize with data). A value of 0 disables this function.

### 7.12.40 Clear Time Tag on Sync Register

**Address: 3E88 – 3E89 (H)**

Write 1 to the lower byte (3E88 H) of the Clear Time Tag on Sync register to indicate that the channel should clear the Time Tag counter (7008 – 700B H) (reset to 0) upon the transmission of a Mode Code 1 message (synchronize). A value of 0 disables this function.

Write 1 to the higher byte (3E89 H) of the Clear Time Tag on Sync register to indicate that the channel should clear the Time Tag counter (7008 – 700B H) (reset to 0) upon the transmission of a Mode Code 17 message (synchronize with data). A value of 0 disables this function.

**Note:** This register setting does not take effect until the channel is restarted.

### 7.12.41 More Channel Options Register

**Address: 3E86 – 3E87 (H)**

**Read only**

The More Channel Options register is a 16-bit register that provides additional channel information.

| Bit | Description |
|---|---|
| **06 – 15** | Reserved |
| **05** | 1 = Expanded Block mode is available in BC mode<br>0 = Expanded Block mode is not available in BC mode |
| **04** | 1 = Enhanced Monitor mode is available in Sequential Fixed-Block Monitor mode<br>0 = Enhanced Monitor mode is not available in Sequential Fixed-Block Monitor mode |
| **03** | 1 = Expanded Block mode is available in Sequential Fixed-Block Monitor mode<br>0 = Expanded Block mode is not available in Sequential Fixed-Block Monitor mode |
| **02** | 1 = Channel is single function (*PxS*)<br>0 = Channel is multifunction (*Px*) |
| **01** | 1 = Onboard Loopback option is available<br>0 = Onboard Loopback option is not available |
| **00** | 1 = Channel is only available in Monitor mode<br>0 = Channel is available in all modes |

**More Channel Options Register**

### 7.12.42 Channel Options Register

**Address: 3E84 – 3E85 (H)**

**Read only**

The Channel Options register is a 16-bit register that provides information about the internal processor and firmware.

| Bit | Description |
|---|---|
| **15** | 1 = PxIII |
| **14** | Reserved – set to 1 |
| **13** | 1 = Expanded Block mode is in use in RT mode |
| **12** | 1 = Channel is on a removable card (PCMCIA or ExpressCard)<br>0 = Channel is on an add-in board |
| **11** | 1 = Replay mode is in use (BC mode only) |
| **10** | 1 = PxII |
| **09** | 1 = 1760 |
| **08** | 1 = 1553 |
| **00 – 07** | 4D H Always set; indicates Internal Concurrent Monitor |

**Channel Options Register**

### 7.12.43 Firmware Revision Register

**Address: 3E8 (H)**

The Firmware Revision register indicates the revision level of the on-channel firmware. The value 18 (H) would be read as revision level: 1.8.

### 7.12.44 Asynchronous Start Flag Register

**Address: 3424 – 3425 (H)**

**Write only** To indicate that it is now time to send a selected frame asynchronously, write a 1 to the Asynchronous Start Flag register. The channel will automatically reset this value to 0 when it sends the frame.

### 7.12.45 Asynchronous Frame Pointer Register

**Address: 3422 – 3423 (H)**

**Write only** To send asynchronously to this register, write the address at the beginning of the selected frame.

### 7.12.46 Asynchronous Message Count Register

**Address: 3420 – 3421 (H)**

**Write only** Write the number of messages contained in the Asynchronous Frame. The maximum number of messages allowed in a frame is determined by the amount of available space in the message stack area of the channel and the size of the individual messages.

# 8 MIL-STD-1553 Channel Bus Monitor Operation (Advanced)

Chapter 8 describes how to operate the MIL-STD-1553 channel in Bus Monitor mode via the memory registers, which enables you to develop your own software tools. When using Excalibur's software tools, this chapter is not required.

The topics covered are:

## 8.1 Bus Monitor Mode Overview

The Bus Monitor can operate in one of two modes, Sequential mode and Look-up Table mode:

**Sequential mode:** 1553 Message Blocks are stored in sequential locations in memory. Sequential mode supports two types of operations, Fixed-Block and Linked-List:

- **Fixed-Block operation:** 1553 messages are stored at fixed sequential blocks in the memory. Sequential Fixed-Block mode supports Trigger capability.
- When using Fixed-Block operation, you have the following options, Regular Monitor, Expanded Monitor and Enhanced Monitor. For more information on these modes, see **8.4.1 Message Block Fixed-Block Operation** on page 8-5. To check whether your board supports Expanded Monitor or Enhanced Monitor, use the **8.12.31 More Channel Options Register** on page 8-28.
- **Linked-List operation:** 1553 messages are packed one after another in the memory, separated by a header.

**Look-up Table mode:** Each 1553 message is stored in a unique Message Block. In Look-up Table mode, the channel addresses the user-programmable Look-up Table when it receives a 1553 Command Word. The Command Word's RT address, T/R bit, and Subaddress fields make up the 11-bit pointer to a Look-up Table with 2048 (2K × 8) locations.

Use the Channel Configuration Register to program the desired mode of operation. See **8.12.4 Channel Configuration Register** on page 8-19.

**To determine if the channel is installed and ready to operate:**

Perform the following procedure after a power-up or a software reset.

1. Check the Channel ID register (test for value = 45 H).
2. Check the Channel Status register (test for Channel Ready bit = 1).

The channel is installed and ready when both registers contain the correct values (as written above). For software reset operations, set these values to 0 immediately prior to writing to the channel Software Reset register.

**Note:** Throughout this manual, writing a 1 to the Start register is referred to as issuing a Start command.

## 8.2 Sequential Fixed-Block Memory Map

| | | | |
|---|---|---|---|
| Reserved | FCB0 – FFFF H | Mode Code Control Register | 3FEA H |
| Fourth Message Block Area (400 Blocks)[1] | 7FB0 – FCAF H | Broadcast Control Register | 3FE8 – 3FE9 H |
| Third Message Block Area (47 Blocks) | 7100 – 7FAF H | Reserved | 3F80 – 3FE7 H |
| Reserved | 700C – 70FF H | 1760 Header Value Transmit Table[2] | 3F40 – 3F7F H |
| Time Tag (Hi) | 700A – 700B H | 1760 Header Value Receive Table[2] | 3F00 – 3F3F H |
| Time Tag (Lo) | 7008 – 7009 H | 1760 Header Exist Table[2] | 3EC0 – 3EFF H |
| Time Tag Reset Register | 7007 H | Expanded Current Message Block Register | 3EBE – 3EBF H |
| Reserved | 7001 – 7006 H | Reserved | 3EAC – 3EBD H |
| | | Pretrigger Message Counter (Lo) | 3EAA –3EAB H |
| | | Pretrigger Message Counter (Hi) | 3EA8 –3EA9 H |
| Channel Reset Register | 7000 H | Reserved | 3EA6 – 3EA7 H |
| Reserved | 6FD1 – 6FFF H | Channel Time Register (Lo) | 3EA4 – 3EA5 H |
| Second Message Block Area (153 Blocks) | 4000 – 6FD0 H | Channel Time Register (Hi) | 3EA2 – 3EA3 H |
| Channel Configuration Register | 3FFF H | Serial Number Register | 3EA0 – 3EA1 H |
| Channel ID Register | 3FFE H | Error Counter (Lo) | 3E9E – 3E9F H |
| Channel Status Register | 3FFD H | Error Counter (Hi) | 3E9C – 3E9D H |
| Start Register | 3FFC H | Message Counter (Lo) | 3E9A – 3E9B H |
| Interrupt Condition Register | 3FFB H | Message Counter (Hi) | 3E98 – 3E99 H |
| Message Status Register | 3FFA H | Reserved | 3E90 – 3E97 H |
| Reserved | 3FF8 – 3FF9 H | Monitor Response Time Reg. | 3E8E – 3E8F H |
| Time Tag Resolution Register | 3FF7 H | 1553A Register | 3E8C – 3F8D H |
| Reserved | 3FF6 H | Channel Function Register | 3E8A – 3E8B H |
| Current Message Block Register | 3FF5 H | Clear Time Tag on Sync Register | 3E88 – 3E89 H |
| Block Trigger Value Register | 3FF4 H | More Channel Options Register | 3E86 – 3E87 H |
| Trigger Word #1 | 3FF2 – 3FF3 H | Channel Options Register | 3E84 – 3E85 H |
| Trigger Mask #1 | 3FF0 – 3FF1 H | Reserved | 3E81 – 3E83 H |
| Trigger Word #2 | 3FEE – 3FEF H | Firmware Revision Register | 3E80 H |
| Trigger Mask #2 | 3FEC – 3FED H | Message Block Area (200 Blocks) | 0000 – 3E7F H |
| Trigger Control Register | 3FEB H | | |

**Figure 8-1 Bus Monitor – Sequential Fixed-Block Memory Map**

1. These 400 blocks are either used for Expanded Monitor or for the additional information saved when using Enhanced Monitor.
2. 1760 Option only

## 8.3 Sequential Linked-List Memory Map

| | | | |
|---|---|---|---|
| Reserved | 700C – FFFF H | Broadcast Control Register | 3FE8 – 3FE9 H |
| Time Tag (Hi) | 700A – 700B H | 1760 Header Value Transmit Table[1] | 3F40 – 3F7F H |
| Time Tag (Lo) | 7008 – 7009 H | 1760 Header Value Receive Table[2] | 3F00 – 3F3F H |
| Time Tag Reset Register | 7007 H | 1760 Header Exist Table[2] | 3EC0 – 3EFF H |
| Reserved | 7001 – 7006 H | Reserved | 3FA6 – 3FBF H |
| | | Channel Time Register (Lo) | 3EA4 – 3EA5 H |
| | | Channel Time Register (Hi) | 3EA2 – 3EA3 H |
| Channel Reset Register | 7000 H | Serial Number Register | 3EA0 – 3EA1 H |
| Reserved | 4000 – 6FFF H | Error Counter (Lo) | 3E9E – 3E9F H |
| Channel Configuration Register | 3FFF H | Error Counter (Hi) | 3E9C – 3E9D H |
| Channel ID Register | 3FFE H | Message Counter (Lo) | 3E9A – 3E9B H |
| Channel Status Register | 3FFD H | Message Counter (Hi) | 3E98 – 3E99 H |
| Start Register | 3FFC H | Reserved | 3E90 – 3E97 H |
| Interrupt Condition Register | 3FFB H | Monitor Response Time Reg. | 3E8E – 3E8F H |
| Message Status Register | 3FFA H | 1553A Register | 3E8C – 3E8D H |
| Reserved | 3FF8 – 3FF9 H | Initial Link Register | 3E8A – 3E8B H |
| Time Tag Resolution Register | 3FF7 H | Clear Time Tag on Sync Register | 3E88 – 3E89 H |
| Reserved | 3FF6 H | More Channel Options Register | 3E86 – 3E87 H |
| End Buffer Pointer | 3FF4 – 3FF5 H | Channel Options Register | 3E84 – 3E85 H |
| Next Message Pointer | 3FF2 – 3FF3 H | Reserved | 3E81 – 3E83 H |
| Reserved | 3FEB – 3FF1 H | Firmware Revision Register | 3E80 H |
| Mode Code Control Register | 3FEA H | Message Block Spill Area | 3E00 – 3E7F H |
| Broadcast Control Register | 3FE8 – 3FE9 H | Message Block Area | 0000 – 3DFF H |

**Figure 8-2 Bus Monitor – Sequential Linked-List Memory Map**

1. 1760 Option only

## 8.4 Sequential Mode Message Block Area

The Sequential Mode Message Block area is partitioned into either blocks of fixed length or into a Linked-List of blocks of varying lengths. The Channel Configuration Register determines the type of partitioning. See **8.12.4 Channel Configuration Register** on page 8-19.

For a description of the Time Tag function, see **8.8 Time Tag Word** on page 8-13.

### 8.4.1 Message Block Fixed-Block Operation

When using Fixed-Block operation, you have the following options:

- **Regular Monitor:** 200 blocks are used to store Bus Monitor data.
- **Expanded Monitor:** 800 blocks are used to store Bus Monitor data.
- **Enhanced Monitor:** 400 blocks are used to store Bus Monitor data, and an additional 400 blocks are used to store additional information about each word.
- For more information, see**8.4.1.1 Expanded Monitor** on page 8-6.

In Fixed-Block operation, the Message Block area is divided into 200, 400 or 800 blocks of 80 bytes each. The first block starts at address 0000 (H), the second at 0050 (H), the third at 00A0 (H), etc. The Trigger option can be used only in Sequential mode with Fixed-Block operation. (See **8.10 Trigger Operation** on page 8-14.)

**Figure 8-3** shows the memory map of each block.

| Information is stored in the memory in the following sequence | Byte offset |
|---|---|
| Message Counter (Lo) | +78 |
| 1553 Data Word | • |
| • | • |
| • | • |
| • | • |
| 1553 Data Word | +8 |
| 1553 Command Word | +6 |
| Time Tag (Hi) | +4 |
| Time Tag (Lo) | +2 |
| Message Status Word | 0 |

**Figure 8-3 Bus Monitor Message Block – Fixed-Block Operation**

#### 8.4.1.1 Expanded Monitor

Expanded Monitor uses 800 blocks to store Bus Monitor data (instead of 200 blocks when using Regular Monitor). When using Expanded Monitor, each block is the same as shown in **Figure 8-3**. Use the **8.12.29 Channel Function Register** on page 8-27, for setting Expanded Monitor.

**Note:** Expanded Monitor is only available in Sequential Monitor mode in Fixed-Block operation. To check whether your board supports Expanded Monitor mode, use the **8.12.31 More Channel Options Register** on page 8-28.

#### 8.4.1.2 Enhanced Monitor

Enhanced Monitor uses 400 blocks for Bus Monitor data, and an additional 400 blocks for additional information about each word. The blocks in the first 400 blocks correspond to the blocks in the additional 400 blocks. For example, block 1 in the regular message buffer corresponds to block 1 of the additional information buffer. The

When using Enhanced Monitor, all words are saved, including those with errors. For example, Data Words without Command Words are saved. When using Regular Monitor or Expanded Monitor, these are disregarded. When an error message has more than 36 words, the message is stored in two or more data blocks, the first 36 words in the first data block, and the remainder in the next data block(s). Use the **8.12.29 Channel Function Register** on page 8-27, for setting Enhanced Monitor.

**Note:** Enhanced Monitor is only available in Sequential Monitor mode in Fixed-Block operation. To check whether your board supports Enhanced Monitor mode, use the **8.12.31 More Channel Options Register** on page 8-28.

**Figure 8-4** shows the memory map of each block of additional information when using Enhanced Monitor.

| Information is stored in the memory in the following sequence | Byte offset |
|---|---|
| Message Counter (Lo) | +78 |
| Additional Message Info | +76 |
| • | • |
| • | • |
| • | • |
| Additional Message Info | +6 |
| Reserved | +4 |
| Reserved | +2 |
| Enhanced Message Status Word | 0 |

**Figure 8-4 Bus Monitor Message Block – Additional Information Block in Enhanced Monitor**

The bits of the Additional Message Info are as follows:

| Bit | Description |
|---|---|
| **12 – 15** | 'C' = Command (or Status) Word sync pattern<br>'D' = Data Word sync pattern |
| **08 – 11** | 'A' = Word was received on Bus A<br>'B' = Word was received on Bus B |
| **04 – 07** | '0' = Word was received contiguously with the previous word<br>'1' = Word was received non-contiguously |
| **03** | Set to 0 |
| **02** | Set to 0 |
| **01** | 0 = No Manchester error occurred<br>1 = Manchester error occurred |
| **00** | 0 = No Parity error occurred<br>1 = Parity error occurred |

**Additional Information**

The bits of the Enhanced Message Status Word are as follows:

| Bit | Description |
|---|---|
| **15** | 1 = End of message; message transfer completed. |
| **06 – 14** | Reserved |
| **00 – 05** | Number of words in the block |

**Enhanced Message Status Word**

### 8.4.2 Message Block Linked-List Operation

In Linked-List operation, the Message Block area is divided into a linked list of message blocks. The length of each message block varies according to message size. The first two locations in each block comprise the message header. This header contains the address of the next Message Block header. The header of the last 1553 block received contains *xx*FF (End of File), indicating that there are no more messages stored. After a message is processed and stored in memory, the header of the preceding message block is updated from *xx*FF to the address (of the header in the block) of the newly stored message.

The Linked-List method can store more data than Fixed-Block operation.

The buffer never wraps around in the middle of a message. When the buffer is full, or if there is no room left to store the entire next message, the next message will be stored in the first location of the Message Block Area (0000 H). The header of the last message will point to that location in the Message Block Area. In this special case, to know exactly where the message ends, use the End Buffer Pointer. The End Buffer Pointer points to the address after the last location of the message, indicating the length of the message. See **8.12.13 End Buffer Pointer** on page 8-23.

**Figure 8-5** illustrates the contents of the Message block. For a description of the Time Tag function, see **8.8 Time Tag Word** on page 8-13.

| Information is stored in the memory in the following sequence: | Byte Offset |
|---|---|
| End Of File: XXFF | • |
| 1553 Data Word | • |
| • | • |
| • | • |
| • | • |
| 1553 Data Word | +A |
| 1553 Command Word | +8 |
| Time Tag (Hi) | +6 |
| Time Tag (Lo) | +4 |
| Message Status Word | +2 |
| Message Header - Address of Next Block | 0 |

**Figure 8-5 Bus Monitor Message Block – Linked-List Operation**

## 8.5 Look-up Table Mode Memory Map

| | | | |
|---|---|---|---|
| Reserved | 700C – FFFF H | Broadcast Control Register | 3FE8 – 3FE9 H |
| Time Tag (Hi) | 700A – 700B H | Reserved | 3F80 – 3FE7 H |
| Time Tag (Lo) | 7008 – 7009 H | 1760 Header Value Transmit Table[1] | 3F40 – 3F7F H |
| Time Tag Reset Register | 7007 H | 1760 Header Value Receive Table[2] | 3F00 – 3F3F H |
| Reserved | 7001 – 7006 H | 1760 Header Exist Table[2] | 3EC0 – 3EFF H |
| | | Reserved | 3FA6 – 3FBF H |
| | | Channel Time Register (Lo) | 3EA4 – 3EA5 H |
| Channel Reset Register | 7000 H | Channel Time Register (Hi) | 3EA2 – 3EA3 H |
| Reserved | 4800 – 6FFF H | Serial Number Register | 3EA0 – 3EA1 H |
| Data Block Look-up Table (2K × 8) | 4000 – 47FF H | Error Counter (Lo) | 3E9E – 3E9F H |
| Channel Configuration Register | 3FFF H | Error Counter (Hi) | 3E9C – 3E9D H |
| Channel ID Register | 3FFE H | Message Counter (Lo) | 3E9A – 3E9B H |
| Channel Status Register | 3FFD H | Message Counter (Hi) | 3E98 – 3E99 H |
| Start Register | 3FFC H | Reserved | 3E90 – 3E97 H |
| Interrupt Condition Register | 3FFB H | Monitor Response Time Register | 3E8E – 3E8F H |
| Message Status Register | 3FFA H | 1553A Register | 3E8C – 3E8D H |
| Reserved | 3FF8 – 3FF9 H | Reserved | 3E8A – 3E8B H |
| Time Tag Resolution Register | 3FF7 H | Clear Time Tag on Sync Register | 3E88 – 3E89 H |
| Reserved | 3FF4 – 3FF6 H | More Channel Options Register | 3E86 – 3E87 H |
| Last Block Register | 3FF2 – 3FF3 H | Channel Options Register | 3E84 – 3E85 H |
| Reserved | 3FEB – 3FF1 H | Reserved | 3E81 – 3E83 H |
| Mode Code Control Register | 3FEA H | Firmware Revision Register | 3E80 H |
| Broadcast Control Register | 3FE8 – 3FE9 H | Message Block Area | 0000 – 3FE7 H |

**Figure 8-6 Bus Monitor Look-up Table Mode Memory Map**

1. 1760 Option only

## 8.6 Look-up Table Mode

In Look-up Table mode, the channel can store 128 unique messages by using a 2K × 8 Look-up Table in dual-port RAM. Each byte in the table is divided into a 7-bit block number and an Interrupt Select bit, as described below. Data Block numbers (0 – 127 decimal) each consisting of 80 bytes are loaded into the table. The first block starts at address 0, the second at 50 (H), etc. Set the Interrupt Select bit to specify which messages will set the interrupt flag. The Interrupt Condition register must also be programmed.

| Bit | Description |
|---|---|
| **07** | 1 Interrupt Select bit is enabled |
| **00 – 06** | Block Numbers (0 – 127) |

**Look-up Table Byte Structure**

When a 1553 message is received, the Command Word's RT address, T/R Bit, and Subaddress fields are used as an 11-bit index to the Look-up Table. This index is used to extract the Data Block number from the Look-up Table.

| | 11 most significant bits of the 1553 Command Word | | | | | |
|---|---|---|---|---|---|---|
| **Base Address** | **RT address (5 bits)** | **T/R (1 bit)** | **Sub-address (5 bits)** | **Look-up table (2K x 8)** | **Data Block Storage Area** | **Address of Data Block** |
| 4000+ | 11111 | 1 | 11111 | Block # | Data Block 127 | 27B0 H |
| | • | • | • | • | • | • |
| | • | • | • | • | • | • |
| | • | • | • | • | • | • |
| 4000+ | 00000 | 0 | 00011 | Block # | Data Block 3 | 00F0 H |
| 4000+ | 00000 | 0 | 00010 | Block # | Data Block 2 | 00A0 H |
| 4000+ | 00000 | 0 | 00001 | Block # | Data Block 1 | 0050 H |
| 4000+ | 00000 | 0 | 00000 | Block # | Data Block 0 | 0000 H |

**Figure 8-7 Look-up Table**

**To create the address to the table:**

1. Isolate the eleven (most significant) bits of the 1553 Command Word (RT Address, T/R, and Subaddress field), and determine their hex value.

**Example:** To allocate a Data Block for a 1553 receive message to RT#5, Subaddress #3.

| RT Address #5 | | | T/R #0 (Rcv) | | | | Subaddress #3 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | LSB |

Hex representation = 143 (H)

2. Add the hex value of this part of the Command Word to the base address of the Look-up table (4000 H).

```
  4000   (H)
+ 143    (H)
------------
  4143   (H)
```

3. Write the 8-bit Data Block number to this location. Each Data Block, beginning at address 0000 is 80 bytes long (for up to 32 1553 Data words). The address of a block is obtained by multiplying its block number by 80 (50 (H)).

**The block addresses are calculated as follows:**

- Block 0 is located at location 0000 (H).
- Block 1 is located at location 0050 (H).
- The location of the block is obtained by multiplying the block number by 80 (50 (H)).

To identify the location of the current, or last, 1553 message, use the Last Block register. The Last Block register is updated at the end of each message reception. See **8.12.15 Last Block Register** on page 8-23.

### 8.6.1 Look-up Table Mode Message Block Area

| Information is stored in the memory in the following sequence: | Byte Offset |
|---|---|
| See **Appendix B MIL-STD-1553 Message Formats** on page B-1 | • • • • |
| 1553 Command Word | +6 |
| Time Tag (Hi) | +4 |
| Time Tag (Lo) | +2 |
| Message Status Word | 0 |

**Figure 8-8 Look-up Table Mode Operation**

## 8.7 Message Status Word

The Message Status Word is identical for all Bus Monitor modes. The Message Status Word indicates the status of the message transfer. The channel creates this Word. Do not confuse it with the 1553 Status Word. (See **6.10.43 1553 RT Status Word Table** on page 6-37.) The contents of the Message Status Word are:

| Bit | Bit Name | Description |
|---|---|---|
| **15** | End of Message | Message transfer completed. |
| **14** | Trigger Found | Trigger message was received and stored. This status is valid for Sequential Fixed-Block mode with the following modes:<br>**Store After** mode: the Trigger Found bit will be set only in the *first* Trigger message.<br>**Store Only** mode: the Trigger Found bit will be set in *every* Trigger message. (See **8.10 Trigger Operation** on page 8-14) |
| **13** | RT-RT | RT-to-RT message was received. |
| **12** | Message Error Bit | Message Error bit (Bit 10) in the RT Status Word was set. |
| **11** | RT Status Bit | A bit other than the Message Error bit in the RT Status Word was set. The Error Bit is *not* set in conjunction with this bit. |
| **10** | Bad RT2RT Format | *either* Second Command Word in an RT-to-RT message is not a transmit command<br>or Missing Gap following Second Command Word in an RT-to-RT message |
| **09** | Checksum Error | The calculated checksum (on the incoming message) does not match when checked against the last Data Word received.<br>(See **8.9 1760 Option** on page 8-13.) |
| **08** | Bus A / B | Bus on which the message was transferred:<br>0 = Bus B<br>1 = Bus A |
| **07** | Invalid Word Received | At least one invalid 1553 Word received (i.e. bit count, Manchester code, parity). |
| **05 – 06** | Word Count/ Header Error | |

| Bit 06 | Bit 05 | Description |
|---|---|---|
| 0 | 0 | Reserved |
| 0 | 1 | Word Count Low |
| 1 | 0 | Word Count High |
| 1 | 1 | 1760 Header Error – Header Word received does not match the value set in the Header Value Table. |

| | | |
|---|---|---|
| **04** | Incorrect RT Address | Received 1553 Status Word did not contain the correct RT address. |
| **03** | Incorrect Sync Received | Sync of either the Status or the Data Word(s) is incorrect. |
| **02** | Non-Contiguous Data | Invalid gap between received 1553 Words. |
| **01** | Response Time Error | Response Time error occurred in the message. |
| **00** | Error | Error occurred. (The error type is defined in one of the other message status bit locations.) |

**Message Status Word**

**Note:**

- The Message Status Word is valid only when Bit 15, End of Message, is turned on.
- When the channel completes receiving a message over the bus, it writes the Message Status Word for this message in its message storage location. The channel then zeros out the Message Status Word in the next message storage location, in preparation for receiving the next message over the bus.

## 8.8 Time Tag Word

In all Bus Monitor modes, each incoming message is stored with a Time Tag value. The Time Tag value is a free-running 32-bit counter on the channel. The Time Tag is reset to 0 upon power-up or a software reset and starts counting. When it reaches the value of FFFF FFFF (H) the counter wraps around to 0 and continues counting. To re-initialize to 0, write to the Time Tag Reset register. (See **8.12.2 Time Tag Reset Register** on page 8-19.)

The Time Tag value can be used to determine the time elapsed between 1553 messages. The equation to determine the Time Tag resolution = (Time Tag Resolution register value + 1) × 4 μsec.

The Time Tag counter's value is written to the dual-port RAM during the reception of the (first) command of each message.

**Note:** In addition to reading the Time Tag value in the message stack, you can also read the counter's value at any time in the Time Tag counter. See **8.12.1 Time Tag Hi & Lo** on page 8-19.

## 8.9 1760 Option

In the MIL-STD-1760 specification, the first Data Word of a message may be a Header Word, which is used for message identification. The Header Word is associated with a specific RT subaddress.

To indicate that a specific subaddress will require a Header Word, set the corresponding entry in the 1760 Header Exist table to 1. Then set the corresponding entry in the 1760 Header Transmit/Receive Value table to the value you expect to receive in the first Data Word of the message. The Header value expected is either the predefined 1760 value, which is the default channel setting, or another value the user enters in the 1760 Header Value Transmit/ Receive Table.

See **8.12.18 1760 Header Value Transmit Table**, **8.12.19 1760 Header Value Receive Table** and **8.12.20 1760 Header Exist Table** on page 8-25.

## 8.10 Trigger Operation

**Triggers are supported only in Sequential Fixed-Block mode**

A trigger is a filter that the user can set to tell the channel when and how to store 1553 messages. The channel can be programmed to store messages in the following ways:

**Trigger Action:**

| | |
|---|---|
| **Store All** | Stores all 1553 messages, without regard to triggers; no triggers are active |
| **Store Only** | Stores only messages that meet the trigger condition |
| **Store After** | Stores only the trigger message and messages that come after the trigger message |

Triggering is done based on a **Trigger Source**. The possible sources are:

- **1553 Command Word**
- **Message Status Word**

One or two triggers may be defined. Each trigger specifies a condition of the trigger source, which if it is fulfilled, will cause the **Trigger Action** to occur. Only one trigger source can be defined for the two triggers.

When two triggers are specified, if *either* of the trigger conditions is true, the trigger action will occur. Each trigger is defined using two registers:

- **Trigger Word Registers (1 and 2)**
- **Trigger Mask Registers (1 and 2)**

Use the Trigger Word register to define a particular 1553 Command Word or a Message Status Word as a trigger. For example, the user can use the Message Status Word as the trigger source to store all messages on bus A, only messages with errors, or messages with errors received over bus B, etc. See **8.10.1 Trigger Word Registers (1 and 2)** on page 8-15.

The Trigger Mask register defines which bits of the trigger word (defined in the Trigger Word register) are relevant and which can be ignored ('don't care'). The Trigger Mask registers must be defined when using the trigger function. See **8.10.2 Trigger Mask Registers (1 and 2)** on page 8-16.

Set the Trigger Control register to specify the following trigger conditions:

- Trigger source (1553 Command Word or Message Status Word)
- Type of storage (Store All, Store Only, or Store After)
- Active trigger word (Trigger Word #1 and/or #2)

See **8.10.3 Trigger Control Register** on page 8-17.

The Trigger Word, Trigger Mask and Trigger Control registers must be set before issuing a Start command to the channel. To modify these registers, set the Initialize bit in the Start register to 10 (H), modify the Trigger Word, Trigger Mask and Trigger Control registers, then issue a Start command 81 (H).

### 8.10.1 Trigger Word Registers (1 and 2)

**Address: Word 1 3FF2 – 3FF3 (H)**
**Word 2 3FEE – 3FEF (H)**

Use the Trigger Word register to define a particular 1553 Command Word or a Message Status Word as a trigger. Load these locations (illustrated below) with the desired 1553 Command Word or Message Status Word, which will be used as the trigger source.

The user must also define the Trigger Mask registers when using the trigger function:

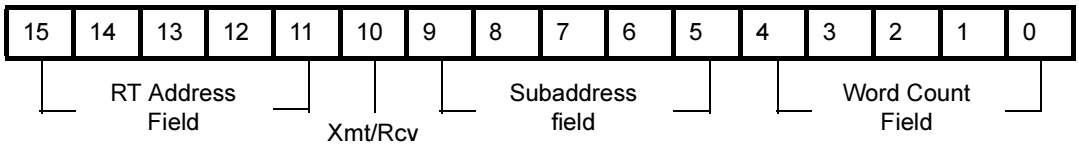
| For: | Define: |
|---|---|
| Trigger Word 1 (3FF2 – 3FF3 H) | Trigger Mask 1 (3FF0 – 3FF1 H) |
| Trigger Word 2 (3FEE – 3FEF H) | Trigger Mask 2 (3FEC – 3FED H) |

See **8.10.2 Trigger Mask Registers (1 and 2)** on page 8-16.

To define which trigger is to be active (Trigger #1, Trigger #2, or both) use the Trigger Control register. (See **8.10.3 Trigger Control Register** on page 8-17.)

#### 8.10.1.1 Using the 1553 Command Word: Trigger Word Registers

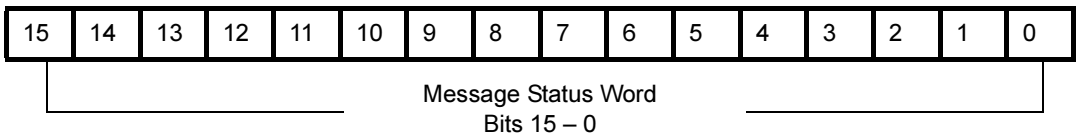
Use a 1553 Command Word as a trigger when it is necessary to filter messages based on information found in the Command Word. For example, to filter messages from a particular RT, or with a particular Word Count, set the Trigger Word register with those parameters defined in the 1553 Command Word.




#### 8.10.1.2 Using the Message Status Word: Trigger Word Registers

Use a Message Status Word as a trigger when it is necessary to filter messages based on information found in the Message Status Word. Do not confuse the Message Status Word with the 1553 Status Word. (See **6.10.43 1553 RT Status Word Table** on page 6-37.) To filter messages transferred over bus A (vs. bus B), or error messages, set the Trigger Word register with those parameters defined in the Message Status Word.

For an explanation see **8.7 Message Status Word** on page 8-12.



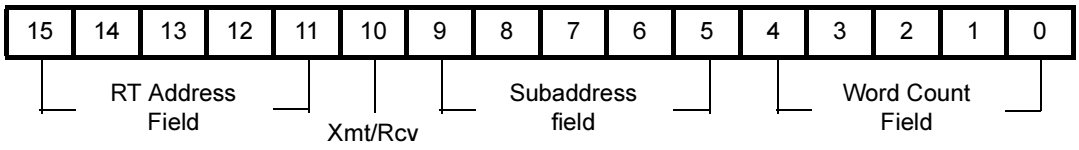
### 8.10.2 Trigger Mask Registers (1 and 2)

**Address: Word 1 3FF0 – 3FF1 (H)**
**Word 2 3FEC – 3FED (H)**

Set the Trigger Mask register to define which bits of the trigger word (defined in the Trigger Word register) are relevant and which can be ignored ("don't care"). The Trigger Mask registers must be defined when using the trigger function. All bits in this register should be set to 1, except for those bits you want to be "don't care" in the incoming Command Word or Message Status Word.

#### 8.10.2.1 Using the 1553 Command Word: Trigger Mask Registers

After setting the Trigger Word register with a 1553 Command Word, write 0s to the bits in the Trigger Mask register that you want to be "don't care" in the 1553 Command Word trigger.

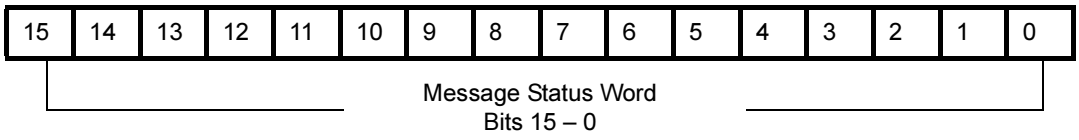



1 = Trigger on corresponding bit value in Trigger Word register
0 = Corresponding bit value in Trigger Word register is "don't care"

#### 8.10.2.2 Using the Message Status Word: Trigger Mask Registers

After setting the Trigger Word register with a Message Status Word, write 0s to the bits in the Trigger Mask register that you want to be "don't care" in the Message Status Word trigger.

For an explanation see **8.7 Message Status Word** on page 8-12.




1 = Trigger on corresponding bit value in Trigger Word register
0 = Corresponding bit value in Trigger Word register is "don't care"

### 8.10.3 Trigger Control Register

**Address: 3FE (H)**

The Trigger Control register is relevant only in Sequential Fixed-Block mode. Set the Trigger Control register to specify the following trigger conditions:

a. Trigger source (1553 Command Word or Message Status Word)
b. Type of storage (Store All, Store Only, or Store After)
c. Active trigger word (Trigger Word #1 and/or #2). If both Trigger Words are active, if either condition a. or b. is met, the trigger will occur.

**Note:** Logic 1 enables the function.

| Bit | Description |
|---|---|
| **07** | Trigger Source:<br>0 = 1553 Command Word<br>1 = Message Status Word |
| **05 – 06** | Reserved |
| **04** | 1 = Store After |
| **03** | 1 = Store Only |
| **02** | 1 = Store All |
| **01** | 1 = Enable Trigger Word #2 |
| **00** | 1 = Enable Trigger Word #1 |

**Trigger Control Register**

**Example:** Defining a Trigger

**Conditions:**

- Define the Command Word 0825 (H) as Trigger word #1 (Receive Command for RT#1, Subaddress #1, and 5 words).
- Ignore the Word Count field.
- Use Trigger word #1 (Disable Trigger word #2).

**Procedure:**

1. Set Trigger Word #1 register to 0825 (H)
2. Set Trigger Mask #1 register to FFE0 (H)
3. Set Trigger Control register to 09 (H)

**Note:** To use trigger(s), at least one of the bits Store All, Store Only, or Store After, must be set.

## 8.11 Interrupts

Interrupts are supported in all three Bus Monitor modes.

### 8.11.1 Sequential Fixed-Block Mode

There are various options for setting interrupts in Sequential Fixed-Block mode, depending on the values set in the Interrupt Condition register (see **8.12.8 Interrupt Condition Register** on page 8-21):

| Bit Description | Bit No. | Set values in | Cause an Interrupt |
|---|---|---|---|
| **Trigger Word Received** | 00 | Trigger Word, Mask and Control registers[1] | Depending on the store condition set in the Trigger Control register:<br>Store Only — *Each* time the trigger word is received<br>Store After — For all messages received *after* the *first* time the trigger is received<br>See **8.10.3 Trigger Control Register** on page 8-17. |
| **Message Complete** | 01 | | Each time any message is received |
| **Block Trigger Match** | 02 | Block number (0 – 199) in the Block Trigger Value register | Set each time that block is updated (i.e. when the Current Message Block register value equals the Block Trigger Value register. See **8.12.11 Current Message Block Register** on page 8-23 and **8.12.12 Block Trigger Value Register** on page 8-23.) |

1. See **8.10 Trigger Operation** on page 8-14.

### 8.11.2 Linked-List Mode

Set the Interrupt Condition register, Bit 01 (Message Complete). An interrupt will occur each time any message is received. (See **8.12.8 Interrupt Condition Register** on page 8-21.)

### 8.11.3 Look-up Table Mode

An interrupt may be enabled or disabled for each block number assigned in the Look-up Table. Each byte in the table is divided into a 7-bit block and Interrupt Select bit. Set the Interrupt Select bit for the desired block. In addition, set the Interrupt Condition register, Bit 01 (Message Complete). (See **8.12.8 Interrupt Condition Register** on page 8-21.) This will cause an interrupt each time a message is received by those blocks that have the Interrupt Select bit set.

See **8.6 Look-up Table Mode** on page 8-10.

## 8.12 Control Register Definitions

### 8.12.1 Time Tag Hi & Lo

**Address: 700A – 700B (H)**
**7008 – 7009 (H)**

**Read only**

The Time Tag is a free-running 32-bit counter on the channel. The Time Tag is reset to 0 upon a power up or a software reset and starts counting. When it reaches the value FFFF FFFF (H), the counter wraps around to 0 and continues counting. To re-initialize to 0, write to the Time Tag Reset register.

The user may read the Time Tag counter at any time. Read the two 16-bit words of the Time Tag counter value sequentially, first Lo word, then Hi word.

**The counter must be read in the following sequence:**

1. Read 7008 H – Lo word (16 bit, read only)
2. Read 700A H – Hi word (16 bit, read only)

The Time Tag Resolution register sets the resolution of the counter.

**To calculate elapsed time between Time Tags:**

**Example:**

1. The Time Tag Resolution register is set to 0. See **8.12.10 Time Tag Resolution Register** on page 8-22.
2. Calculate the Time Tag Resolution:
   (Time Tag resolution register value + 1) × 4 = (0 + 1) × 4 = 4 μsec
3. Calculate difference between Time Tags:
   150 (Time Tag 2) – 50 (Time Tag 1) = 100
4. Elapsed time
   100 × 4 = 400 μsec

### 8.12.2 Time Tag Reset Register

**Address: 700 (H)**

**Write only**

Write to the Time Tag Reset register to reset the channel Time Tag Counter (data field = don't care). Immediately after the reset, the counter will start to count from 0.

**Note:** The counter may also be reset from an external source (*EXC-1553UNET/Px* only). See **4.5.5 External Signals Connector Pinouts** on page 4-10.

### 8.12.3 Channel Reset Register

**Address: 700 (H)**

Write any value to the Channel Reset register to reset the channel.

Channel Reset erases all locations in the dual-port RAM. Channel status, Channel ID and Firmware Revision registers are written by the channel after the reset operation is completed.

### 8.12.4 Channel Configuration Register

**Address: 3FF (H)**

Before issuing a Start command to the channel, set the operating mode of the channel via the Channel Configuration register. To modify the Channel Configuration register, issue a Stop command, modify the register, and then

issue a Start command. (See **8.12.7 Start Register** on page 8-21.)

| Hex Value | Operating Mode |
|---|---|
| **08** | Bus Monitor Sequential Fixed -block |
| **10** | Bus Monitor Sequential Linked-List |
| **20** | Bus Monitor Look-up table |

**Channel Configuration Register Values: Monitor Mode**

### 8.12.5 Channel ID Register

**Address: 3FF (H)**

The Channel ID register contains a fixed value that can be read by your initialization routine to detect the presence of the channel. The one-byte value of this register is: 45 (H), ASCII value E.

### 8.12.6 Channel Status Register

**Address: 3FF (H)**

The Channel Status register indicates the status of the channel. In addition, this register indicates which options have been selected. Do not modify this register. Status bits are active if set to 1.

| Bit | Description |
|---|---|
| **07** | 1 = Always set |
| **05 – 06** | Indeterminate |
| **04** | 1 = Channel Halted<br>0 = Channel Running |
| **03** | 1 = Self-Test OK |
| **02** | 1 = Timers OK |
| **01** | 1 = RAM OK |
| **00** | 1 = Channel Ready |

**Channel Status Register**

**Note:** Channel operation stops after the Start bit in the Start register is cleared. Following this, the channel sets Bit 04 (Channel Halted). Certain registers may be modified only after the Channel Halted bit has been set. After receiving a subsequent Start command (by writing to the Start register), the channel resets the Channel Halted bit. The condition of this bit after power-up or software reset is logic 1.

### 8.12.7 Start Register

**Address: 3FF (H)**

The Start register controls the Start/Halt operation of the channel.

| Bit | Bit Name | Description |
|---|---|---|
| **01 – 07** | **Reserved** | Set to 0 |
| **00** | **Start/Halt** | 1 = Start Operation<br>0 = Halt Operation |

**Start Register**

**Note:** You can start the module externally by sending a minimum LVTTL pulse of 100 nsec. to the EXSTARTn pin. See **4.5.5 External Signals Connector Pinouts** on page 4-10.

### 8.12.8 Interrupt Condition Register

**Address: 3FF (H)**

Set the Interrupt Condition register to enable interrupt triggers. When a condition enabled in this register occurs, an interrupt is generated. Logic 1 enables the interrupt condition. Check the Message Status register to determine which condition caused the interrupt. (See **8.12.9 Message Status Register** on page 8-22.)

The Interrupt Condition register must be set before issuing a Start command to the channel. To modify the Interrupt Condition register, issue a Stop command, modify the register, then issue a Start command (See **8.12.7 Start Register** on page 8-21.)

**Note:** For all interrupt conditions, the interrupt will be sent at the end of the message.

| Bit | Description |
|---|---|
| **03 – 07** | 0 |
| **02** | 1 = Block Trigger Match (valid only in Sequential Fixed-Block mode) |
| **01** | 1 = Message complete (valid in all modes) |
| **00** | 1 = Trigger Word Received (valid only in Sequential Fixed-Block mode) |

**Interrupt Condition Register**

### 8.12.9 Message Status Register

**Address: 3FF (H)**

The Message Status register indicates the status of the current message being processed. Each status bit is described in the table below. Logic 1 indicates that the condition is activated.

| Bit | Description |
|---|---|
| **03 – 07** | 0 |
| **02** | 1 = Block Trigger Match |
| **01** | 1 = Message Reception In Progress |
| **00** | 1 = Trigger Word Received – Sequential Fixed-block mode<br>*or*<br>Trigger Word Busy – Linked-List and Look-up table mode<br>The busy bit is set when the channel is processing a message. It is set together with message reception in progress, but is reset approximately 5 msec. after the end of each message. For consecutive messages with short intermessage gap times, the busy bit may not be reset between messages. |

**Message Status Register**

**Note:** Status bits are not reset by the channel. They must be reset after reading them.

### 8.12.10 Time Tag Resolution Register

**Address: 3FF (H)**

The 8-bit value in the Time Tag Resolution register represents the resolution of the Time Tag Counter in units of 4 μsec.

To determine the Time Tag Counter's resolution, use the following equation:

= (Time Tag Resolution register value + 1) × 4 μsec.

A value of 0 corresponds to a resolution of 4 microseconds; a value of 1 corresponds to a resolution of 8 microseconds, etc.

Set the Time Tag Resolution register before issuing a Start command to the channel. To modify the Time Tag Resolution register, issue a Stop command, modify the register, and then issue a Start command. (See **8.12.7 Start Register** on page 8-21.)

### 8.12.11 Current Message Block Register

**Address: 3FF (H)**

**Sequential Fixed-Block mode only**

Read the Current Message Block register to determine the Current Message Block number (0 – 199). The value is incremented by the channel as each message is received. The first counter increment (to 1), which indicates that the first message has been received and stored, occurs at the beginning of the *second* 1553 message transfer operation. To determine the arrival of the first 1553 message, check the Message Status Word of the *first* Message block. The End of Message bit (Bit 15) in the Message Status Word will be set.

**Note:** When Expanded Monitor mode or Enhanced Monitor mode is enabled, the **8.12.21 Expanded Current Message Block Register** register is used instead of this register.

### 8.12.12 Block Trigger Value Register

**Address: 3FF (H)**

**Sequential Fixed-Block mode only**

Set the Block Trigger Value register to a block number (0 – 199). This will set a bit in the Message Status register each time that a block is updated, i.e. when the Current Message Block value equals the Block Trigger Value register. It will also cause an interrupt each time that block is updated if in the Interrupt Condition register Bit 02 (Block Trigger Match) is set. (See **8.12.8 Interrupt Condition Register** on page 8-21.)

Set the Block Trigger Value register before issuing a Start command to the channel. To modify the Block Trigger Value register, issue a Stop command, modify the register, and then issue a Start command. (See **8.12.7 Start Register** on page 8-21.)

### 8.12.13 End Buffer Pointer

**Address: 3FF4 – 3FF5 (H)**

**Linked-List mode only**

The End Buffer pointer points to the address following the last word in the final message in the Message Block area. The End Buffer pointer is updated each time a final message is written into the buffer. Final messages that are longer than the remaining available space in the Message Block area do not wrap around to the start of the buffer. They are spilled into the Message Block Spill area, which is contiguous to the Message Block area. The value of this register varies from 3400 (H) (end of Message Block area) to 347E (H) (end of Message Block Spill area). Until the first buffer wrap around occurs, this register contains 0000 (H).

### 8.12.14 Next Message Pointer

**Address: 3FF2 – 3FF3 (H)**

**Linked-List mode only**

The Next Message pointer is a 16-bit pointer that indicates the address of the 1553 message about to be written. The Next Message pointer register is updated at the end of each message storage operation. It cycles from 0 (H) to 33FE (H).

### 8.12.15 Last Block Register

**Address: 3FF2 – 3FF3 (H)**

**Look-up Table mode only**

Read the Last Block register to determine the (Look-up Table) block number of the current 1553 message. This register is used to identify the location of the current 1553 message. The Last Block register is updated at the end of each message reception.

### 8.12.16 Mode Code Control Register

**Address: 3FE (H)**

Set the Mode Code Control register to specify which 1553 Subaddress value indicates the reception of a 1553 Mode command.

The Mode Code Control register must be set before issuing a Start command to the channel. To modify the Mode Code Control register, issue a Stop command, modify the register, then issue a Start command. (See **8.12.7 Start Register** on page 8-21.)

| Bit | Description | | |
|---|---|---|---|
| **02 – 07** | 0 | | |
| **00 – 01** | **Bit 01** | **Bit 00** | **Subaddresses Recognized as Mode Code** |
| | 0 | 0 | 31 and 0 |
| | 0 | 1 | 0 |
| | 1 | 0 | 31 |
| | 1 | 1 | 0 and 31 |

**Mode Code Control Register**

### 8.12.17 Broadcast Control Register

**Address: 3FE8 – 3FE9 (H)**

Set the Broadcast Control register to specify whether RT address 11111 should be regarded as a valid RT number or as the Broadcast address.

| Bit | Description |
|---|---|
| **01 – 07** | 0 |
| **00** | 1 = RT #31 is Broadcast Address<br>0 = RT #31 is Regular RT |

**Broadcast Control Register**

### 8.12.18 1760 Header Value Transmit Table

**Address: 3F40 – 3F7F H**

**1760 Option only**

Write to the 1760 Header Value Transmit table to set the expected value of the first Data Word in a RT-to-BC message. The monitor checks that the specified header value was received. If the wrong data was sent, the **1760 Header error** bit is set in the Message Status Word, see **8.7 Message Status Word** on page 8-12.

The 1760 option provides predefined values, and these are preset on each channel. The user can change the preset values.

| Transmit Subaddress | Header Value | Address |
|---|---|---|
| **1** | **0421 H** | 3F42 H |
| **11** | **0420 H** | 3F56 H |
| **14** | **0423 H** | 3F5C H |

**Predefined 1760 Transmit Header Value**

### 8.12.19 1760 Header Value Receive Table

**Address: 3F00 – 3F3F (H)**

**1760 Option only**

Write to the 1760 Header Value Receive table to set the expected value of the first Data Word in a BC-to-RT message. The monitor checks that the specified header value was received. If the wrong data was sent, the **1760 Header error** bit is set in the Message Status Word, see **8.7 Message Status Word** on page 8-12.

The 1760 option provides predefined values, and these are preset on each channel. The user can change the preset values.

| Receive Subaddress | Header Value | Address |
|---|---|---|
| **11** | **0400 H** | 3F16 H |
| **14** | **0422 H** | 3F1C H |

**Predefined 1760 Receive Header Values**

### 8.12.20 1760 Header Exist Table

**Address: 3EC0 – 3EFF (H)**

**1760 Option only**

The 1760 Header Exist table contains 32 entries corresponding to 32 RT subaddresses. Each entry may be set to indicate whether, or not, the channel should expect a header word for messages directed to that subaddress. In Bus Monitor mode, there is a separate bit to select Header Words for transmit and receive messages.

For those Header Exist table entries for which MIL-STD 1760 provides predefined values, the corresponding Header Exist table entries are preset on each channel. To set other values, enable the Header Exist table entry for this Subaddress (set it to 1) and write the value to the Header Value (Transmit/ Receive) table.

| Bit | Description |
|---|---|
| **09 – 15** | Reserved |
| **08** | 1 = Channel should expect a Header word in a transmit message (RT-to-BC or RT-to-RT)<br>0 = Channel should not expect a Header word in a transmit message |
| **01 – 07** | Reserved |
| **00** | 1 = Channel should expect a Header word in a receive message (BC-to-RT)<br>0 = Channel should not expect a Header word in a receive message |

**1760 Header Exist Table**

| Associated Subaddress | Header Value | Address |
|---|---|---|
| 11 | 0100 H | **3EC2 H** |
| 14 | 0101 H | **3ED6 H** |
| 1 | 0101 H | **3EDC H** |

**Predefined 1760 Header Values**

### 8.12.21 Expanded Current Message Block Register

**Address: 3EBE – 3EBF (H)**

**Sequential Fixed-Block mode only**

When Expanded Monitor mode or Enhanced Monitor mode is enabled, read the Expanded Current Message Block register to determine the Current Message Block number (0 – 799 when Expanded Monitor mode is enabled; 0 – 399 when Enhanced Monitor mode is enabled). (For more information on these modes, see **8.4.1 Message Block Fixed-Block Operation** on page 8-5.)

The value of this register is incremented by the channel as each message is received. The first counter increment (to 1), which indicates that the first message has been received and stored, occurs at the beginning of the *second* 1553 message transfer operation. To determine the arrival of the first 1553 message, check the Message Status Word of the *first* Message block. The End of Message bit (Bit 15) in the Message Status Word will be set.

**Note:** When Expanded Monitor mode or Enhanced Monitor mode is enabled, this register is used instead of the **8.12.11 Current Message Block Register**.

### 8.12.22 Pretrigger Message Counter Lo & Hi

**Address: 3EAA – 3EAB (H)**
**3EA8 – 3EA9 (H)**

**Sequential Fixed-Block mode only**

The Pretrigger Message Counters Lo & Hi keep track of how many messages were received *until* the trigger kicked in. This enables the user to know that there is activity, until the trigger condition is fulfilled, causing the monitor to actually begin monitoring.

### 8.12.23 Channel Time Register Lo & Hi

**Address: 3EA4 – 3EA5 (H)**
**3EA2 – 3EA3 (H)**

This register holds the channel time value, which is stored in non-volatile flash memory and loaded at power-up. This value can be modified by calling the Set_ModuleTime_Px function. (See the *1553Px Family Software Tools Programmer's Reference*.) The factory default value is FFFF FFFF (H).

### 8.12.24 Serial Number Register

**Address: 3EA0 – 3EA1 (H)**

This register holds the board's serial number, which is stored in non-volatile flash memory and loaded at power-up. The value is binary coded. For example, a value of 1234 (H) represents the serial number 4660.

### 8.12.25 Error Counter Lo & Hi

**Address: 3E9E – 3E9F (H)**
**3E9C – 3E9D (H)**

Error Counter is a running 32-bit counter of message errors.

### 8.12.26 Message Counter Lo & Hi

**Address: 3E9A – 3E9B (H)**
**3E98 – 3E99 (H)**

Message Counter is a running 32-bit counter of all messages received.

### 8.12.27 Monitor Response Time Register

**Address: 3E8E – 3E8F (H)**

The Monitor Response Time register sets the maximum wait time until the Monitor considers an RT's Status Response valid.

The Monitor Response Time register is measured in microseconds. The default value of the register is 14 μsec, if not set otherwise by the user.

### 8.12.28 1553A Register

**Address: 3E8C – 3F8D H**

Set the 1553A register to simulate MIL-STD-1553A protocol. If set to 1553A protocol, Mode Codes are assumed not to have any data.

### 8.12.29 Channel Function Register

**Address: 3E8A – 3E8B (H)**

**Sequential Fixed-Block mode only**

The Channel Function register is a 16-bit register that specifies whether the module is using Expanded or Enhanced Monitor. For more information, see **8.4.1 Message Block Fixed-Block Operation** on page 8-5.

| Bit | Description |
|---|---|
| **02 – 15** | Reserved |
| **01** | 1 = Enhanced Monitor is in use<br>0 = Enhanced Monitor is not in use |
| **00** | 1 = Expanded Monitor is in use<br>0 = Expanded Monitor is not in use |

**Module Options Register**

**Note:** Do not set both Bit 00 and Bit 01 to 1. Enhanced Monitor and Expanded Monitor cannot be enabled simultaneously.

### 8.12.30 Clear Time Tag on Sync Register

**Address: 3E88 – 3E89 (H)**

Write 1 to the lower byte (3E88 H) of the Clear Time Tag on Sync register to indicate that the Channel should clear the Time Tag counter (7008 – 700B H) (resets to 0) upon receipt of a Mode Code 1 message (synchronize). A value of 0 disables this function.

Write 1 to the higher byte (3E89 H) of the Clear Time Tag on Sync register to indicate that the channel should clear the Time Tag counter (7008 – 700B H) (resets to 0) upon receipt of a Mode Code 17 message (synchronize with data). A value of 0 disables this function.

**Note:** This register setting does not take effect until the channel is restarted.

### 8.12.31 More Channel Options Register

**Address: 3E86 – 3E87 (H)**

**Read only**

The More Channel Options register is a 16-bit register that provides additional channel information.

| Bit | Description |
|---|---|
| **06 – 15** | Reserved |
| **05** | 1 = Expanded Block mode is available in BC mode<br>0 = Expanded Block mode is not available in BC mode |
| **04** | 1 = Enhanced Monitor mode is available in Sequential Fixed-Block Monitor mode<br>0 = Enhanced Monitor mode is not available in Sequential Fixed-Block Monitor mode |
| **03** | 1 = Expanded Block mode is available in Sequential Fixed-Block Monitor mode<br>0 = Expanded Block mode is not available in Sequential Fixed-Block Monitor mode |
| **02** | 1 = Channel is single function (*PxS*)<br>0 = Channel is multifunction (*Px*) |
| **01** | 1 = Onboard Loopback option is available<br>0 = Onboard Loopback option is not available |
| **00** | 1 = Channel is only available in Monitor mode<br>0 = Channel is available in all modes |

**More Channel Options Register**

### 8.12.32 Channel Options Register

**Address: 3E84 – 3E85 (H)**

**Read only**

The Channel Options register is a 16-bit register that provides information about the internal processor and firmware.

| Bit | Description |
|---|---|
| **15** | 1 = PxIII |
| **14** | Reserved; set to 1 |
| **13** | 1 = Expanded Block mode is in use in RT mode |
| **12** | 1 = Channel is on a removable card (PCMCIA or ExpressCard)<br>0 = Channel is on an add-in board |
| **11** | 1 = Replay mode is in use (BC mode only) |
| **10** | 1 = PxII |
| **09** | 1 = 1760 |
| **08** | 1 = 1553 |
| **00 – 07** | 4D H Always set; indicates Internal Concurrent Monitor |

**Channel Options Register**

### 8.12.33 Firmware Revision Register

**Address: 3E8 (H)**

The Firmware Revision register indicates the revision level of the channel firmware. The value 18 (H) would read as revision 1.8.

# 9 MIL-STD-1553 Channel Internal Concurrent Monitor (Advanced)

Chapter 9 describes how to operate the MIL-STD-1553 Internal Concurrent Monitor via the memory registers, which enables you to develop your own software tools. When using Excalibur's software tools, this chapter is not required.

The topics covered are:



An Internal Concurrent Monitor operates automatically on each channel when the channel is started in either RT or BC/RT modes. It operates in Sequential Fixed-Block mode, the 1553 Message blocks are stored in sequential locations in memory. The storing of messages starts at the first block.

## 9.1 Internal Concurrent Monitor Memory Map

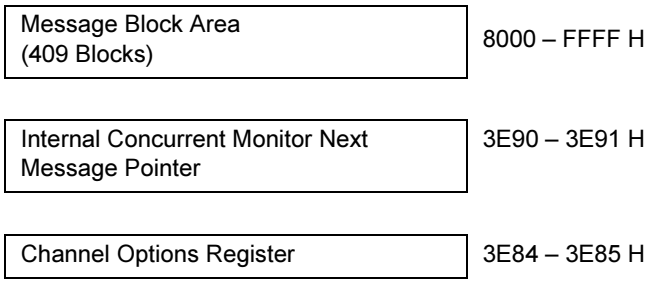



**Figure 9-1 Internal Concurrent Monitor Memory Map**

When Expanded Block mode is set in RT mode, the Internal Concurrent Monitor Message Block area is reduced.

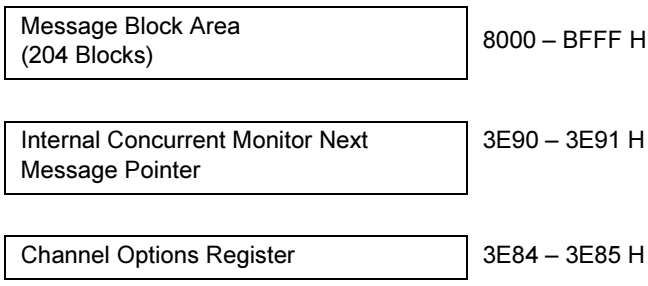



**Figure 9-2 Internal Concurrent Monitor Memory Map with Expanded Block Mode**

## 9.2 Message Block Area

The message block area is divided into 409 blocks of 80 bytes each (or 204 blocks when using Expanded Block mode in RT mode). The first block starts at address 8000 (H), the second at 8050 (H), the third at 80A0 (H), etc.

| | |
|---|---|
| Block #408 | FFFF H |
| • | |
| Block #203 | BFFF H<br>(End of Internal Concurrent Monitor when using Expanded Block mode in RT mode) |
| • | |
| Block #2 | 80A0 H |
| Block #1 | 8050 H |
| Block #0 | 8000 H |

**Message Block Area**

### 9.2.1 Message Block Structure

Each message block occupies 40 words. These 40 words include a Message Status Word, 2 consecutive Time Tag words and all the 1553 message words. (See **Appendix B MIL-STD-1553 Message Formats** on page B-1.)

Word 40 is a serial counter. The first message will have a serial counter value of 1; the second message will have a value of 2, etc.

| |
|---|
| Serial Counter Word |
| 1553 Message Word 36 |
| • |
| • |
| 1553 Message Word 2 |
| 1553 Message Word 1 |
| Time Tag Word #2 (MSB) |
| Time Tag Word #1 (LSB) |
| Message Status Word |

**Internal Concurrent Monitor Message Block Structure**

### 9.2.2 Message Status Word

The Message Status Word indicates the status of the message transfer. The channel creates this word. Do not confuse it with the 1553 Status Word. (See **6.10.43 1553 RT Status Word Table** on page 6-37.) The contents of the Message Status Word are shown below.

**Note:** The Message Status Word is different in RT/Concurrent Monitor mode and BC-RT/Internal Concurrent Monitor mode.

A logic 1 indicates the occurrence of a status flag.

**Message Status Word: RT/Internal Concurrent Monitor**

| Bit | Bit Name | Description |
|---|---|---|
| **15** | **End of Message** | Message transfer completed |
| **14** | **Bus A / B** | Bus on which the message was transferred (1 = BUS A) |
| **13** | **1760 Checksum Error** (1760 option only) | The calculated checksum (on an incoming message) does not match the last Data Word received |
| **12** | **Message Error** | Message Error bit (Bit 10) in the RT Status Word was set |
| **11** | **RT Status** | A bit other than the Message Error bit in the RT Status Word was set. The Error bit is not set in conjunction with this bit. |
| **10** | **TX Time Out** | The channel, acting as receiver in RT-to-RT message, did not sense a transmitter Status Word. |
| **09** | **Response Error** | Response time error occurred in the message, even if no RT is active on the channel. |
| **08** | **Invalid Message RT/Internal Concurrent Monitor** | 1553 message-level error occurred (e.g., Word Count, Sync Error). See other bits set for the exact error. For example: an RT-to-RT message which contains two receive messages. |
| **07** | **Invalid Word Received** | At least one invalid 1553 Word received (i.e. bit count, Manchester code, parity) |
| **06** | **1760 Header Word Error** (1760 option only) | Header Word received does not match the value set in the Header Value Table (1760 option only) See **6.9.1 Header Word** on page 6-15 |
| **05** | **Word Count Error** | Incorrect number of words received in the message |
| **04** | **Incorrect RT Address** | Received 1553 Status Word did not contain the correct RT address |
| **03** | **Sync Error** | Sync of either the Status or the Data Word(s) is incorrect |
| **02** | **Non-Contiguous Data** | Invalid gap between received 1553 Words |
| **01** | **RT2RT Message** | RT-to-RT message was received |
| **00** | **Error** | Error occurred. The error type is defined in one of the other message status bit locations. |

**Message Status Word: BC-RT/Internal Concurrent Monitor**

| Bit | Bit Name | Description |
|---|---|---|
| 15 | **End of Message** | Message transfer completed. |
| 14 | **Bus A / B** | Bus on which the message was transferred (1 = BUS A) |
| 13 | **1760 Checksum Error** (1760 option only) | The calculated checksum (on an incoming message) does not match the last Data Word received. See **7.11.2 Checksum** on page 7-16 |
| 12 | **Message Error** | Message Error bit (Bit 10) in the RT Status Word was set. |
| 11 | **RT Status** | A bit other than the Message Error bit in the RT Status Word was set. The Error bit is not set in conjunction with this bit. |
| 10 | **Invalid Message** | 1553 message-level error occurred (e.g., Word Count, Sync Error). See other bits set for the exact error. For example: an RT- to-RT message which contains two receive messages. |
| 09 | **Response Error** | Response time error occurred in the message, even if no RT is active on the channel. |
| 08 | **1760 Header Word Error** (1760 option only) | Header Word received does not match the value set in the Header Value Table (1760 option only) See **7.11.1 Header Word** on page 7-16 |
| 07 | **Invalid Word Received** | At least one invalid 1553 Word received (i.e. bit count, Manchester code, parity). |
| 06 | **Word Count High** | RT transmitted too many words. |
| 05 | **Word Count Low** | RT transmitted too few words. |
| 04 | **Incorrect RT Address** | Received 1553 Status Word did not contain the correct RT address. |
| 03 | **Sync Error** | Sync of either the Status or the Data Word(s) is incorrect. |
| 02 | **Non-Contiguous Data** | Invalid gap between received 1553 Words. |
| 01 | **RT2RT Message** | RT-to-RT message was received. |
| 00 | **Error** | Error occurred. (The error type is defined in one of the other message status bit locations.) |

**Note:** The message contents are valid only after the Message Status Word has been written, which is indicated by the End of Message bit being turned on.

### 9.2.3 1553 Message Words

The 1553 message words are stored in the sequence they appear on the bus, i.e., 1553 Command Words, 1553 Status Words, 1553 Data Words - all according to the order of the specific type of message.

## 9.3 Control Register Definitions

### 9.3.1 Internal Concurrent Monitor Next Message Pointer

**Address: 3E90 – 3E91 (H)**

The Internal Concurrent Monitor Next Message pointer is a 16-bit pointer that indicates the address of the 1553 message about to be written. The register is updated at the end of each message storage operation. It cycles from 8000 (H) to FFFF (H).

### 9.3.2 Channel Options Register

**Address: 3E84 – 3E85 (H)**

**Read only**

The Channel Options register is a 16-bit register that shows information about the internal processor and firmware.

| Bit | Description |
|---|---|
| **15** | 1 = PxIII |
| **14** | Reserved – set to 1 |
| **13** | 1 = Expanded Block mode is in use in RT mode |
| **12** | 1 = Channel is on a removable card (PCMCIA or ExpressCard)<br>0 = Channel is on an add-in board |
| **11** | 1 = Replay mode is in use (BC mode only) |
| **10** | 1 = PxII |
| **09** | 1 = 1760 |
| **08** | 1 = 1553 |
| **00 – 07** | 4D H Always set; indicates Internal Concurrent Monitor |

**Channel Options Register**

# 10 Switching Modes of Operation of a MIL-STD-1553 Channel (Advanced)

Chapter 10 describes how to switch the channel's operating mode via the memory registers, which enables you to develop your own software tools. When using Excalibur's software tools, this chapter is not required.

Many test applications simulate only one mode, for example, Remote Terminal mode. For these applications, this chapter is irrelevant.

If your application requires simulation of more than one mode, you can switch from one mode of operation to another, for example, between the Bus Controller/ Concurrent-RT mode and Remote Terminal modes.

**To switch between modes of operation:**

1. Halt the operation of the channel (via the Start register).
2. Modify the Channel Configuration register to the desired mode.

| Hex Value | Operating Mode |
|---|---|
| **02** | Remote Terminal |
| **04** | BC/Concurrent-RT |
| **08** | Bus Monitor Sequential Fixed-Block |
| **10** | Bus Monitor Sequential Linked-List |
| **20** | Bus Monitor Look-Up Table |

**Channel Configuration Register Values**

3. Set up the memory as required.
4. Set the Start bit in the Start register.

# 11 Discrete Module Control Registers (Advanced)

Chapter 11 describes how to run the Discrete I/Os via the memory registers, which enables you to develop your own software tools. When using Excalibur's software tools, this chapter is not required.

The following topics are covered:

## 11.1 Discretes Module Memory Map

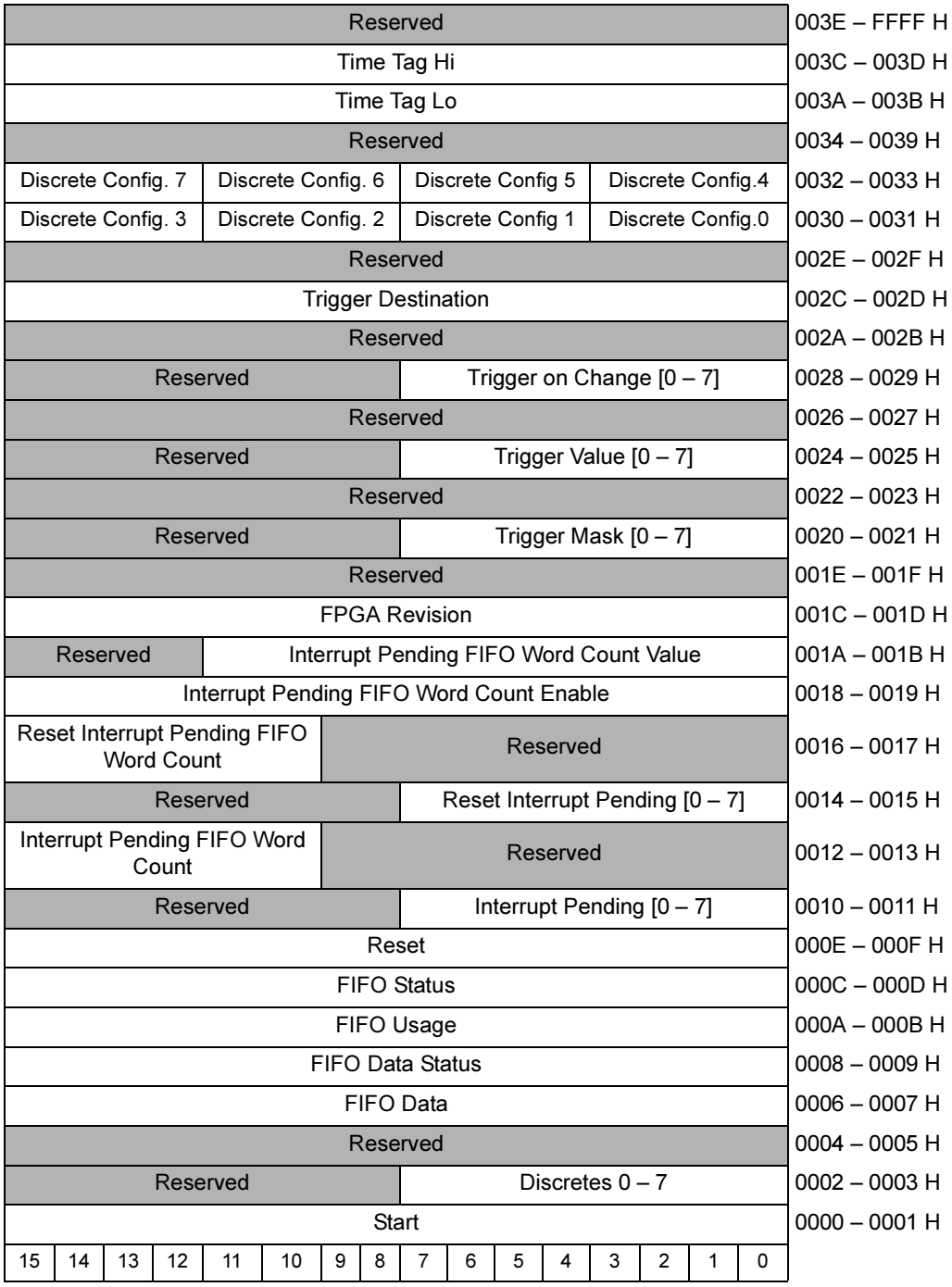



**Figure 11-1 Discretes Control Registers Map**

## 11.2 Discrete Control Register Definitions

### 11.2.1 Start Register

**Address: 00000 – 00001 (H)**

**Read/Write** The Start Register controls the start/stop operation of the module.

| Bit | Description |
|---|---|
| **01-15** | Reserved |
| **00** | Start/Stop operation<br>0 = Stop module - **Default**<br>All bits in the Discrete Input Register remain constant and are not effected by changes in the voltage levels on the external Discrete lines. The bits in the Discrete Output registers can not be altered and the value on the output external Discrete lines will be dependent on the last value stored in the Output register before the module was stopped.<br>Configuration of the Discretes is allowed<br>1 = Start module<br>All bits in the Input and Output registers can be updated and/or changed.<br>Configuration of the Discretes is *not* allowed |

**Start Register**

### 11.2.2 Discrete Registers

**Address: 00002 – 00003 (H) Discretes 0 – 7**

When the module is stopped, each Discrete can be programmed by the user either as an Input Discrete or as an Output Discrete. See **Discretes Configuration Registers,** on page 11-12. The bits in the Discrete register control and represent the values associated with these Discretes.

**Read only** **Discretes Configured as Input [Default]**

When a Discrete has been configured as an input, the associated bit of this register represents the value stored in the input Discrete. Either an Avionics or TTL Discrete logic-level will be stored, depending on the voltage threshold configured for the Discrete. (See **Discretes Configuration Registers,** on page 11-12)

When the module is *stopped,* all bits in this register remain constant and are not affected by changes in the logic-level voltages on the Input Discretes. When the module is *started,* all the bits in this register will be updated/changed according to the current logic-level voltages on the input Discretes. (See **Start Register,** page 11-3.)

**Note** When an input Discrete has been configured with External Debounce *enabled* (see **Discretes Configuration Registers,** on page 11-12) a change in the input Discrete will only be stored in the associated bit of this register if the input Discrete is stable for a minimum of 60ms.

| Bit | Signal | Description |
|---|---|---|
| 08-15 | Reserved | |
| 07 | IN7 | Logic-level voltage on input Discrete 7 |
| 06 | IN6 | Logic-level voltage on input Discrete 6 |
| 05 | IN5 | Logic-level voltage on input Discrete 5 |
| 04 | IN4 | Logic-level voltage on input Discrete 4 |
| 03 | IN3 | Logic-level voltage on input Discrete 3 |
| 02 | IN2 | Logic-level voltage on input Discrete 2 |
| 01 | IN1 | Logic-level voltage on input Discrete 1 |
| 00 | IN0 | Logic-level voltage on input Discrete 0 |

**Read only: Input Discrete Register - Discretes 0 – 9: Address: 00002 – 00003 (H)**

**Note** All Input Discretes are sampled on either power-on, software reset or a start of the module. Thus for any of these actions, the voltage level on the external Discrete lines will be stored to the bits of this register.

**Write only** **Discretes Configured as Output**

When an I/O Discrete has been configured as an output, the associated bit of this register controls the value set on the output Discrete. The threshold voltage level of the output Discrete will depend on what voltage the line is pulled up to (since the output Discretes are open collector). See **Discrete Channel Information,** on page 1-11.

When the module is *stopped,* all bits in this register cannot be changed and the output Discretes are kept at the voltage level that was set before the board was stopped. When the module is *started,* all the bits in this register can be changed and will update the voltage levels set to the associated Discrete output accordingly. (See **Start Register,** page 11-3)

All bits of the register have a default value of 1, which will tri-state the external Discrete lines.

| Bit | Signal | Description |
|---|---|---|
| 08-15 | Reserved | |
| 07 | OUT7 | Logic-level voltage on output Discrete 7 |
| 06 | OUT6 | Logic-level voltage on output Discrete 6 |
| 05 | OUT5 | Logic-level voltage on output Discrete 5 |
| 04 | OUT4 | Logic-level voltage on output Discrete 4 |
| 03 | OUT3 | Logic-level voltage on output Discrete 3 |
| 02 | OUT2 | Logic-level voltage on output Discrete 2 |
| 01 | OUT1 | Logic-level voltage on output Discrete 1 |
| 00 | OUT0 | Logic-level voltage on output Discrete 0 |

**Read/Write: Output Discrete Register - Discretes 0 – 9: Address: 00002 – 00003 (H)**

### 11.2.3 FIFO Data Register

**Address: 0006 – 0007(H)**

**Read only**

The *EXC-1553UNET/Px* contains a FIFO which stores changes that occur on the input Discretes. The Fifo data register displays the value of the oldest data stored in the FIFO. When any of the Discretes are triggered on a Rising Edge, Falling Edge or Change, the new value stored in the Discrete registers will be stored in the FIFO along with an associated Time tag.

See **Discretes 0 – 7 Trigger Mask Register,** page 11-9,
**Discretes 0 – 7 Trigger Value Register,** page 11-10 and
**Discretes 0 – 7 Trigger on Change Register,** page 11-11.

After every trigger, four 16-bit words will be stored in the FIFO. The words are stored in the following order:

1. Discrete Inputs 0 – 7
2. 16 zeros (reserved)
3. The Low word of the TTAG associated with the stored Discretes
4. The High word of the TTAG associated with the stored Discretes

**Note** The unused upper bits of the Discrete Input words mentioned above (1& 2) are set to zero.

When any word is stored in the FIFO, the value in the FIFO Usage register will be incremented by one. When any word is read from the FIFO, the value in the FIFO Usage register will be decremented by one and that word will be deleted from the memory of the FIFO.

The Time tag that is stored in the FIFO with the Discrete data has a size of 32-bits.

Time Tag is reset to 0 upon a power up, a channel reset, or a global software reset, and starts counting. (See **Reset Register**, on page 11-7 and **Software Reset Register**, on page 5-3.) When it reaches the value FFFF FFFF (H), the counter wraps around to 0 and continues counting. To re-initialize to 0 without doing a software reset, write to the Time Tag Reset register.

On the *EXC-1553UNET/Px* only, the Time Tag can also be reset via the External Time Tag Reset signal from the External Signals Connector, allowing synchronization to other systems. See the **External Signals Connector Pinouts**, on page 4-10.

The Time Tag resolution is 4 μs. So the Time Tag will wrap around after approximately 4.8 hours. The 4 μs resolution is based on the *UNET*'s 1 μs global clock for Time Tags. On the *EXC-1553UNET/Px*, you can change the Discrete channel's Time Tag resolution by using an external Time Tag clock source via the EXTTCLKI pin of the External Signals Connector. (See the **External Signals Connector Pinouts**, on page 4-10.) When using an external clock, the Discrete channel will multiply the clock resolution by 4. For example, if the clock is taken from an external source that has a 10 μs resolution, the Discrete channel Time Tag will have a 40 μs resolution.

The reset value of all the bits in the FIFO is 0, i.e., all data in the FIFO is cleared on a reset.

### 11.2.4 FIFO Data Status Register

**Address: 0008 – 0009 (H)**

**Read only**

The FIFO Data Status Register displays the *type* of data that is displayed in the FIFO Data Register.

| Bit | Description | Value | |
|---|---|---|---|
| **02-15** | Reserved - Set to 0 | | |
| **00-01** | FIFO Data Register type | 00 | DIscrete inputs 0 – 9 [**Default**] |
| | | 01 | Discrete inputs 10 – 19 |
| | | 10 | Time tag - Low word |
| | | 11 | Time tag - High word |

**FIFO Data Status register**

### 11.2.5 FIFO Usage Register

**Address: 000A – 000B (H)**

**Read only**

The FIFO Usage Register defines how much of the FIFO's memory has been used.

**Note**

1. The FIFO has a maximum capacity of 4092, 16 bit words.
2. For each trigger, 4 words will be automatically stored in the FIFO

| Bit | Description |
|---|---|
| **12-15** | Reserved - Set to 0 |
| **00-11** | Defines the number of words stored in the FIFO<br>**Default: 0000** |

**FIFO Usage register**

### 11.2.6 FIFO Status Register

**Address: 000C – 000D (H)**

**Read only**

The FIFO Status Register defines the status of the FIFO memory

| Bit | Description |
|---|---|
| **04-15** | Reserved - Set to 0 |
| **03** | 0 FIFO memory is not full **Default**<br>1 FIFO memory is full [4092 words] |
| **02** | 0 FIFO has sufficient memory left - there are less than 4072 words stored in FIFO **Default**<br>1 FIFO memory is nearly full: there are 4072 or more words stored in FIFO |
| **01** | 0 FIFO memory is not empty<br>1 FIFO memory is empty **Default** |
| **00** | 0 FIFO memory is nowhere near empty: there are 20 or more words stored in FIFO<br>1 FIFO memory is nearly empty - there are less than 20 words stored in FIFO **Default** |

**FIFO Status register**

### 11.2.7 Reset Register Address: 000E – 000F (H)

**Write only** Writing any value to the Reset register will cause a software reset to the module. A reset will cause all registers to restore to the default values all clear all data from the FIFO.

### 11.2.8 Discrete Inputs [0 – 7] Interrupt Pending Register Address: 0010 – 0011 (H)

Use the Discrete Inputs [0 – 7] Interrupt Pending Register to identify which Discrete or Discretes (0 – 7) produced a trigger and may have requested an interrupt/external trigger.

The default value for all these bits is 0.

| Bit | Description |
|---|---|
| **08-15** | Reserved - set to 0 |
| **07** | 0 = No trigger generated by Discrete 7<br>1 = Trigger generated by Discrete 7 |
| **06** | 0 = No trigger generated by Discrete 6<br>1 = Trigger generated by Discrete 6 |
| **05** | 0 = No trigger generated by Discrete 5<br>1 = Trigger generated by Discrete 5 |
| **04** | 0 = No trigger generated by Discrete 4<br>1 = Trigger generated by Discrete 4 |
| **03** | 0 = No trigger generated by Discrete 3<br>1 = Trigger generated by Discrete 3 |
| **02** | 0 = No trigger generated by Discrete 2<br>1 = Trigger generated by Discrete 2 |
| **01** | 0 = No trigger generated by Discrete 1<br>1 = Trigger generated by Discrete 1 |
| **00** | 0 = No trigger generated by Discrete 0<br>1 = Trigger generated by Discrete 0 |

**Discrete Inputs [0 – 7] Interrupt Pending Register**

### 11.2.9 Reset Discrete Inputs [0 – 7] Interrupt Pending Register Address: 0014 – 0015 (H)

**Write only** Writing to the Reset Discrete Inputs [0 – 7] Interrupt Pending Register will reset the corresponding bits of the Discrete Inputs [0 – 7] Interrupt Pending Register.

| Bit | Description |
|---|---|
| **10-15** | Reserved - set to 0 |
| **00-09** | Reset Discrete Inputs [0 – 7] Interrupt Pending<br>0 = No effect<br>1 = Reset the corresponding bit in the Discrete Inputs [0 – 7] Interrupt Pending Register |

**Reset Discrete Inputs [0 – 7] Interrupt Pending Register**

**Note** The Discrete Inputs [0 – 7] Interrupt Pending Register bits are not cleared automatically, they must be reset with this register.

### 11.2.10 Interrupt on FIFO Word Count Enable Register Address: 0018 – 0019 (H)

**Read/Write** When the bit of this register is set, then the Interrupt on FIFO Word Count register will be enabled.

| Bit | Description |
|---|---|
| **01-15** | Reserved set to 0 |
| **00** | 0 Interrupt on FIFO Word Count will be disabled - **Default**<br>1 Interrupt on FIFO Word Count will be enabled |

**Interrupt on FIFO Word Count Enable Register**

### 11.2.11 Interrupt on FIFO Word Count Value Register Address: 001A – 001B (H)

**Read/Write** When the number of words stored in the FIFO is equal to or greater than user defined value stored in this register, then an interrupt will be generated (if this feature has been enabled see **Interrupt on FIFO Word Count Enable Register,** page 11-8).

| Bit | Description |
|---|---|
| **12-15** | Reserved set to 0 |
| **00-11** | Interrupt on FIFO Word Count Value:<br>**Default 0FFF** |

### 11.2.12 FPGA Revision Register Address: 001C – 001D (H)

**Read only** The FPGA Revision register contains the FPGA revision of the module.

### 11.2.13 Discretes 0 – 7 Trigger Mask Register

**Address: 0020 – 0021 (H)**

**Read/Write**

The Discrete 0 – 7 Trigger Mask Register defines which input Discretes are to be monitored for activity. When any of the masked inputs change, a trigger will be produced depending on the value of the Trigger Value Register. (See **Discretes 0 – 7 Trigger Value Register**, on page 11-10 or **Discretes 0 – 7 Trigger on Change Register**, on page 11-11) The output of the trigger is dependent on the **Trigger Destination Register**, on page 11-11. The Discrete that produced the trigger is indicated in the **Discrete Inputs [0 – 7] Interrupt Pending Register**, page 11-7.

The default value for all the bits of the register is 0.

| Bit | Description |
|---|---|
| **08-15** | Reserved - set to 0 |
| **07** | 0 = Do not monitor Discrete 7<br>1 = Monitor Discrete 7 |
| **06** | 0 = Do not monitor Discrete 6<br>1 = Monitor Discrete 6 |
| **05** | 0 = Do not monitor Discrete 5<br>1 = Monitor Discrete 5 |
| **04** | 0 = Do not monitor Discrete 4<br>1 = Monitor channel 4 |
| **03** | 0 = Do not monitor Discrete 3<br>1 = Monitor Discrete 3 |
| **02** | 0 = Do not monitor Discrete 2<br>1 = Monitor Discrete 2 |
| **01** | 0 = Do not monitor Discrete 1<br>1 = Monitor Discrete 1 |
| **00** | 0 = Do not monitor Discrete 0<br>1 = Monitor Discrete 0 |

**Discrete 0 – 7 Trigger Mask Register**

### 11.2.14 Discretes 0 – 7 Trigger Value Register

**Address: 0024 – 0025 (H)**

**Read/Write**

The Discrete 0 – 7 Trigger Value Register defines if a trigger will be generated by a rising edge or a falling edge on the specific input Discretes 0 – 7.

The default value for all the bits of the register is 0.

| Bit | Description |
|---|---|
| **08-15** | Reserved |
| **07** | 0 = Rising edge on input Discrete 7<br>1 = Falling edge on input Discrete 7 |
| **06** | 0 = Rising edge on input Discrete 6<br>1 = Falling edge on input Discrete 6 |
| **05** | 0 = Rising edge on input Discrete 5<br>1 = Falling edge on input Discrete 5 |
| **04** | 0 = Rising edge on input Discrete 4<br>1 = Falling edge on input Discrete 4 |
| **03** | 0 = Rising edge on input Discrete 3<br>1 = Falling edge on input Discrete 3 |
| **02** | 0 = Rising edge on input Discrete 2<br>1 = Falling edge on input Discrete 2 |
| **01** | 0 = Rising edge on input Discrete 1<br>1 = Falling edge on input Discrete 1 |
| **00** | 0 = Rising edge on input Discrete 0<br>1 = Falling edge on input Discrete 0 |

**Discretes 0 – 7 Trigger Value Register**

### 11.2.15 Discretes 0 – 7 Trigger on Change Register Address: 0028 – 0029 (H)

**Read/Write** The Discretes 0 – 7 Trigger On Change Register defines if a trigger will be generated by a change (rising edge or falling edge) on the specific input Discretes 0 – 7.

The default value for all the bits of this register is 0.

| Bit | Description |
|---|---|
| **08-15** | Reserved – Set to 0 |
| **07** | 0 = Do not Trigger on change of input Discrete 7<br>1 = Trigger on change of input Discrete 7 |
| **06** | 0 = Do not Trigger on change of input Discrete 6<br>1 = Trigger on change of input Discrete 6 |
| **05** | 0 = Do not Trigger on change of input Discrete 5<br>1 = Trigger on change of input Discrete 5 |
| **04** | 0 = Do not Trigger on change of input Discrete 4<br>1 = Trigger on change of input Discrete 4 |
| **03** | 0 = Do not Trigger on change of input Discrete 3<br>1 = Trigger on change of input Discrete 3 |
| **02** | 0 = Do not Trigger on change of input Discrete 2<br>1 = Trigger on change of input Discrete 2 |
| **01** | 0 = Do not Trigger on change of input Discrete 1<br>1 = Trigger on change of input Discrete 1 |
| **00** | 0 = Do not Trigger on change of input Discrete 0<br>1 = Trigger on change of input Discrete 0 |

**Discretes 0 – 7 Trigger on Change Register**

### 11.2.16 Trigger Destination Register Address: 002C – 002D (H)

**Read/Write** Whenever a trigger occurs, the value of the Discrete registers will be added to the Discrete FIFO along with a Time Tag. The trigger may also be used to generate an interrupt or external signal.

Use the Trigger Destination Register to control where the trigger on the input or inputs is directed.

| Bit | Description | | |
|---|---|---|---|
| **02-15** | Reserved - set to 0 | | |
| **00-01** | Trigger Output | | |
| | **Bit 01** | **Bit 00** | |
| | 0 | 0 | Not Connected |
| | 0 | 1 | Routed to interrupt line |
| | 1 | 0 | Reserved |
| | 1 | 1 | Reserved |

**Trigger Destination Register**

For a description of the External Signal and its pinout, see the **External Signals**

**Connector Pinouts,** page 4-10.

### 11.2.17 Discretes Configuration Registers

**Address: 0030 – 0031 (H) Discretes 0 – 3**
**0032 – 0033 (H) Discretes 4 – 7**

**Read/Write** Each of these registers controls the configuration of 4 separate Discretes. There are 4 bits associated with each Discrete, each with the functionality outlined below. Note these registers can only be changed while the board has been stopped.

**Note**

1. For information concerning Discrete set as Inputs or Outputs, see **Discrete Registers,** page 11-3.
2. For information about Input Threshold Designation see **Discrete Channel Information,** page 1-11.
3. If External Debounce has been enabled, then a change on any of the input Discretes will only be stored once the input is stable for a minimum of 60 ms.

### 11.2.18 Discretes 0 – 3 Configuration Registers

**Address: 0030 – 0031 (H)**

The default value for all the bits of the register is 0.

| Bit | Discrete # | Name | Description |
|---|---|---|---|
| **15** | | Reserved | |
| **14** | **3** | Input Debounce Designation | 0 = External Debounce disabled<br>1 = External Debounce enabled |
| **13** | **3** | Input Threshold Designation | 0 = TTL Voltage Threshold<br>1 = Avionics Voltage Threshold |
| **12** | **3** | I/O Designation | 0 = Set as Input<br>1 = Set as Output |
| **11** | | Reserved | |
| **10** | **2** | Input Debounce Designation | 0 = External Debounce disabled<br>1 = External Debounce enabled |
| **09** | **2** | Input Threshold Designation | 0 = TTL Voltage Threshold<br>1 = Avionics Voltage Threshold |
| **08** | **2** | I/O Designation | 0 = Set as Input<br>1 = Set as Output |
| **07** | | Reserved | |
| **06** | **1** | Input Debounce Designation | 0 = External Debounce disabled<br>1 = External Debounce enabled |
| **05** | **1** | Input Threshold Designation | 0 = TTL Voltage Threshold<br>1 = Avionics Voltage Threshold |
| **04** | **1** | I/O Designation | 0 = Set as Input<br>1 = Set as Output |
| **03** | | Reserved | |
| **02** | **0** | Input Debounce Designation | 0 = External Debounce disabled<br>1 = External Debounce enabled |
| **01** | **0** | Input Threshold Designation | 0 = TTL Voltage Threshold<br>1 = Avionics Voltage Threshold |
| **00** | **0** | I/O Designation | 0 = Set as Input<br>1 = Set as Output |

**Discretes 0 – 3 Configuration Register**

### 11.2.19 Discretes 4 – 7 Configuration Registers

**Address: 0032 – 0033 (H)**

The default value for all the bits of the register is 0.

| Bit | Discrete # | Name | Description |
|---|---|---|---|
| **15** | | Reserved | |
| **14** | **7** | Input Debounce Designation | 0 = External Debounce disabled<br>1 = External Debounce enabled |
| **13** | **7** | Input Threshold Designation | 0 = TTL Voltage Threshold<br>1 = Avionics Voltage Threshold |
| **12** | **7** | I/O Designation | 0 = Set as Input<br>1 = Set as Output |
| **11** | | Reserved | |
| **10** | **6** | Input Debounce Designation | 0 = External Debounce disabled<br>1 = External Debounce enabled |
| **09** | **6** | Input Threshold Designation | 0 = TTL Voltage Threshold<br>1 = Avionics Voltage Threshold |
| **08** | **6** | I/O Designation | 0 = Set as Input<br>1 = Set as Output |
| **07** | | Reserved | |
| **06** | **5** | Input Debounce Designation | 0 = External Debounce disabled<br>1 = External Debounce enabled |
| **05** | **5** | Input Threshold Designation | 0 = TTL Voltage Threshold<br>1 = Avionics Voltage Threshold |
| **04** | **5** | I/O Designation | 0 = Set as Input<br>1 = Set as Output |
| **03** | | Reserved | |
| **02** | **4** | Input Debounce Designation | 0 = External Debounce disabled<br>1 = External Debounce enabled |
| **01** | **4** | Input Threshold Designation | 0 = TTL Voltage Threshold<br>1 = Avionics Voltage Threshold |
| **00** | **4** | I/O Designation | 0 = Set as Input<br>1 = Set as Output |

**Discretes 4 – 7 Configuration Register**

### 11.2.20 Time Tag Hi & Lo

**Address:** **003A – 003B (H)**
**003C – 003D (H)**

**Read only**

This Time Tag is a free-running 32-bit counter on the Discrete channel. Time Tag is reset to 0 upon a power up, a channel reset, or a global software reset, and starts counting. (See **Reset Register**, on page 11-7 and **Software Reset Register**, on page 5-3.) When it reaches the value FFFF FFFF (H), the counter wraps around to 0 and continues counting. To re-initialize to 0 without doing a software reset, write to the Time Tag Reset register.

On the *EXC-1553UNET/Px* only, the Time Tag can also be reset via the External Time Tag Reset signal from the External Signals Connector, allowing synchronization to other systems. See the **External Signals Connector Pinouts**, on page 4-10.

The Time Tag resolution is 4 μs. So the Time Tag will wrap around after approximately 4.8 hours. The 4 μs resolution is based on the *UNET*'s 1 μs global clock for Time Tags. On the *EXC-1553UNET/Px*, you can change the Discrete channel's Time Tag resolution by using an external Time Tag clock source via the EXTTCLKI pin of the External Signals Connector. (See the **External Signals Connector Pinouts**, on page 4-10.) When using an external clock, the Discrete channel will multiply the clock resolution by 4. For example, if the clock is taken from an external source that has a 10 μs resolution, the Discrete channel Time Tag will have a 40 μs resolution.

The user may read the Time Tag counter at any time. Read the two 16-bit words of the Time Tag counter value sequentially, first Lo word, then Hi word.

**The counter must be read in the following sequence:**

1. Read 003A H – Lo word (16 bit, read only)
2. Read 003C H – Hi word (16 bit, read only)

# 12 Ordering Information

Chapter 12 describes the part numbers to indicate when ordering a *UNET*. Replace '***x***' in the part numbers with the required number of MIL-STD-1553 channels (1 or 2).

**Note:** The *EXC-1553UNET/Px* is supplied with two Micro-B to Standard-A USB cables, and a USB power supply.

## 12.1 Ordering Information of *EXC-1553UNET/Px*

Use the following part numbers when ordering an *EXC-1553UNET/Px*.

| Part Number | Option | Description |
|---|---|---|
| **EXC-1553UNET/P*x*-M**<br>(Replace '*x*' with 1 or 2) | | Multifunction MIL-STD-1553 USB/Ethernet device; panel mounted with Trompeter BJ157 connectors (or equivalent). Supports multiple RTs, BC/Concurrent-RT or Triggerable Bus Monitor modes, with an Internal Concurrent Monitor in RT and BC/Concurrent-RT modes.<br>In addition, there are 8 Discrete channels.<br>For this configuration, Excalibur supplies 30 cm (11.8 in.) adapter cables that have Trompeter PL155 (or equivalent) connectors on one side and Trompeter CJ70 (or equivalent) on the other side<br>(P/N: X1BA1AB2-0.3). A 15-pin micro DB15 mating connector for the External Signals Connector is also provided. |
| **EXC-1553UNET/P*x*S-M**[1]<br>(Replace '*x*' with 1 or 2) | | Single function MIL-STD-1553 USB/Ethernet device; panel mounted with Trompeter BJ157 connectors (or equivalent). Supports single RT, BC, or Triggerable Bus Monitor modes with an Internal Concurrent Monitor in RT and BC modes, without error injection.<br>In addition, there are 8 Discrete channels.<br>For this configuration, Excalibur supplies 30 cm (11.8 in.) adapter cables that have Trompeter PL155 (or equivalent) connectors on one side and Trompeter CJ70 (or equivalent) on the other side<br>(P/N: X1BA1AB2-0.3). A 15-pin micro DB15 mating connector for the External Signals Connector is also provided. |
| **EXC-1553UNET/P*x*-C**<br>(Replace '*x*' with 1 or 2) | | Multifunction MIL-STD-1553 USB/Ethernet device with hard-wired MIL-STD-1553 cable with Trompeter CJ70 connectors (or equivalent). Supports multiple RTs, BC/Concurrent-RT or Triggerable Bus Monitor modes, with an Internal Concurrent Monitor in RT and BC/Concurrent-RT modes.<br>In addition, there are 8 Discrete channels.<br>A 15-pin micro DB15 mating connector for the External Signals Connector is provided. |
| **EXC-1553UNET/P*x*S-C**[1]<br>(Replace '*x*' with 1 or 2) | | Single function MIL-STD-1553 USB/Ethernet device with hard-wired MIL-STD-1553 cable with Trompeter CJ70 connectors (or equivalent). Supports single RT, BC, or Triggerable Bus Monitor modes with an Internal Concurrent Monitor in RT and BC modes, without error injection.<br>In addition, there are 8 Discrete channels. A 15-pin micro DB15 mating connector for the External Signals Connector is provided. |
| | **-1760** | Add this suffix for 1760 protocol support. |
| | **-D** | Add this suffix for 1553 Direct Bus coupling. |
| | **-P** | Add this suffix for a mounting plate. |
| | **-B** | Add this suffix for an internal rechargeable battery. |

1. A single function *UNET* can be ordered with the jumpers set to a specific RT address and parity. Contact Excalibur Sales for more information. See **1.8 Technical Support,** on page 1-12.

## 12.2 Ordering Information of *ES-1553RUNET/Px*

Use the following part numbers when ordering an *ES-1553RUNET/Px*.

| Part Number | Option | Description |
|---|---|---|
| **ES-1553RUNET/P1** | | Rugged, single-channel, multifunction MIL-STD-1553 Ethernet device with military standard connectors. Supports multiple RTs, BC/Concurrent-RT or Triggerable Bus Monitor modes, with an Internal Concurrent Monitor in RT and BC/Concurrent-RT modes. In addition, there are 8 Discrete channels. |
| **ES-1553RUNET/P1S** | | Rugged, single-channel, single function MIL-STD-1553 Ethernet device with military standard connectors. Supports single RT, BC, or Triggerable Bus Monitor modes with an Internal Concurrent Monitor in RT and BC modes, without error injection. In addition, there are 8 Discrete channels. |
| | **-1760** | Add this suffix for 1760 protocol support. |
| | **-D** | Add this suffix for 1553 Direct Bus coupling. |

# Appendix A MIL-STD-1553 Word Formats

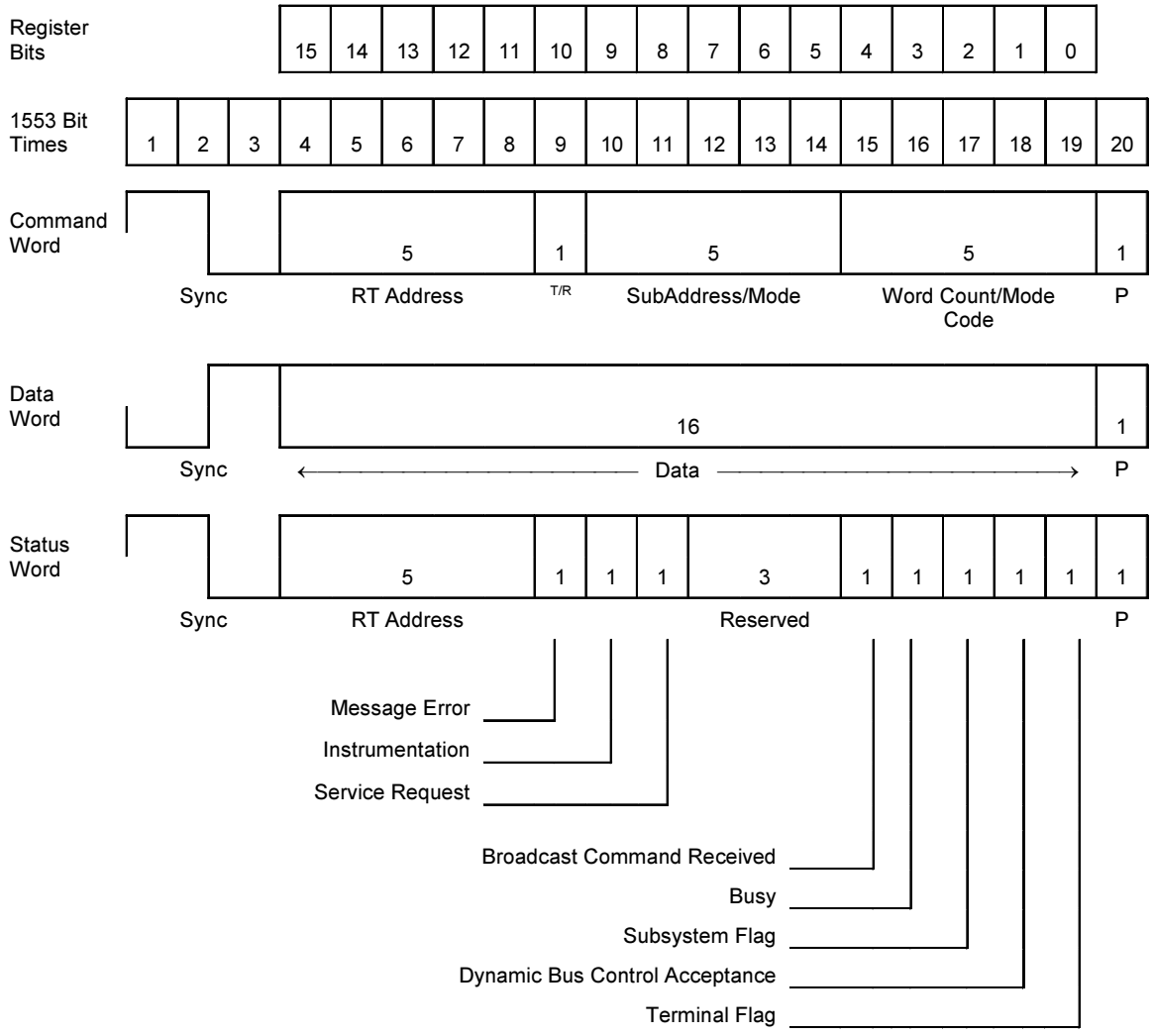



**Figure A-1 MIL-STD-1553 Word Formats**

**Note:** T/R = Transmit/Receive
P = Parity

# Appendix B MIL-STD-1553 Message Formats

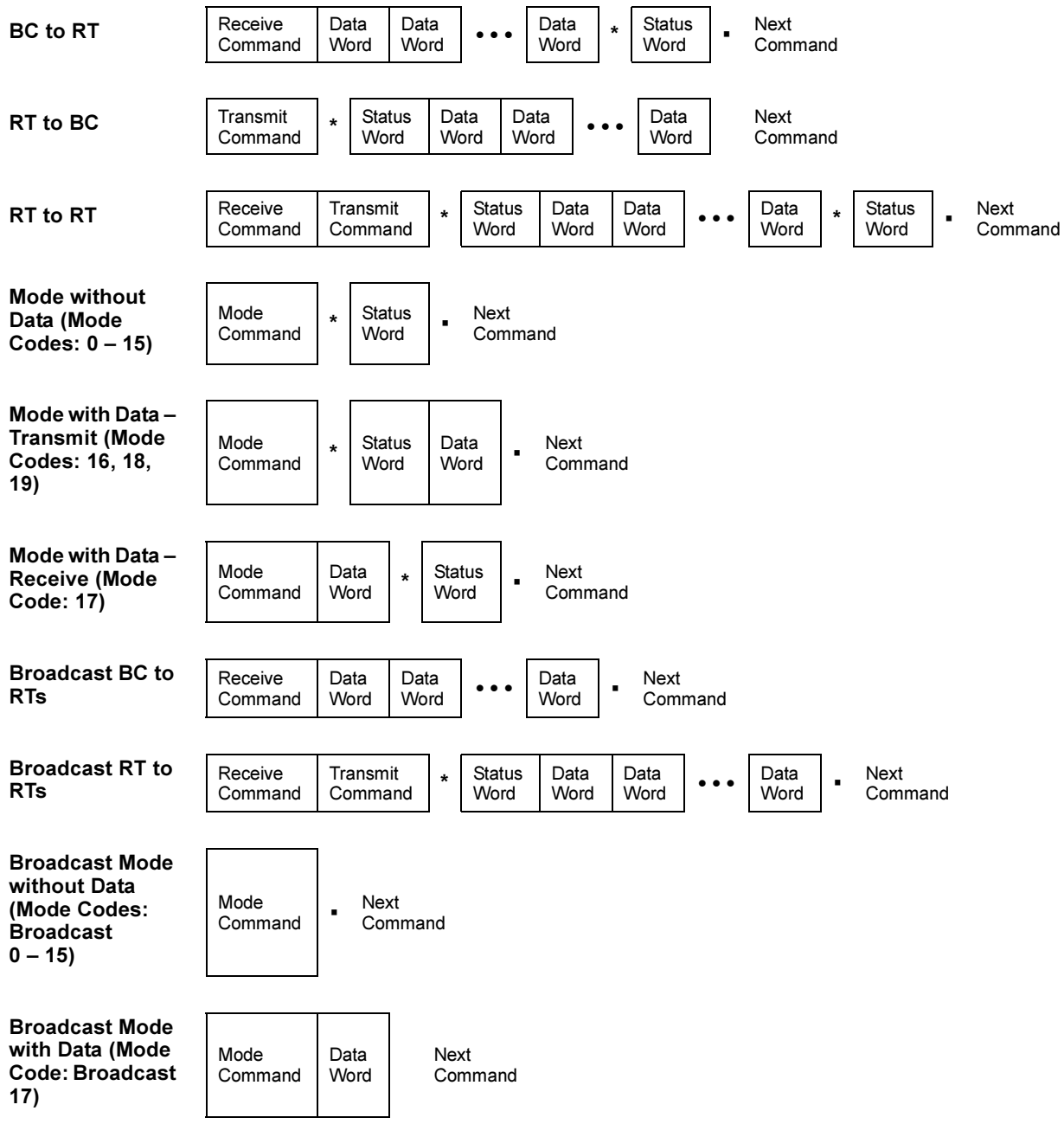



**Figure B-1 MIL-STD-1553B Message Formats**

**Note:** * = Response time
▪ = Intermessage Gap time

# Appendix C Internal Loopback Test

The Internal Loopback Test is used to check the 1553 front-end logic, excluding transceivers and coupling transformers.

**Note:** When running a loopback, all values previously written to registers and memory addresses are erased.

**To initiate the Internal Loopback Test:**

1. Write ED (H) into the Card Configuration Register.
2. Write 1 into the Start Register.
3. Wait for 0 in the Start Register.

The results of this test are returned to the host in dual-port RAM using the following structure beginning at address 0:

| | Definition | Address in Dual-Port RAM | Status Value |
|---|---|---|---|
| {struct I_LOOP-BACK | | | |
| usint frame_val; | | 0 | X (not for user) |
| usint frame_status; | frame time counter status | 2 | 8000H passed, 8001H failed |
| usint resp_status; | response time counter status | 4 | 8000H passed, 8001H failed |
| usint early_val; | | 6 | 6 LSB must be 15H |
| usint receive_data1; | first looped word test, using com-mand sync | 8 | 5555H |
| usint status_1; | | A | 8000H passed, else failed |
| usint receive_data2; | second looped word test, using data sync | C | AAAAH |
| usint status_2; | | E | 8000H passed, else failed |
| usint mc_status; | mode code func-tion test | 10 | 8000H passed, else failed |
| usint ttag_val_lo; | | 12 | 30D4H ± 2 |
| usint ttag_val_hi; | | 14 | 0 |
| usint ttag_status; | time tag status | 16 | 8000H passed, 8001H failed |
| usint prl; | | 18 | 8 LSB contain the CPU version |
| } *I_loopback; | | | |

# Appendix D External Loopback Test

The External Loopback Test is used to check the 1553 transceivers, transformers and associated bus cables.

**Note:** **T**he External Loopback Test requires a loopback cable to connect bus A to bus B.

**To initiate the External Loopback test:**

1. Write FF (H) into the Card Configuration Register.
2. Write 1 into the Start Register.
3. Wait for 0 in the Start Register.

**Note:** When running a loopback, all values previously written to registers and memory addresses are erased.

The results of this test are returned to the host in dual-port RAM using the following structure beginning at address 0:

| | Definition/conditions for passing E_loopback test (TX-bus, RX-bus, command or data sync) | Address in Dual-Port RAM | Status Value |
|---|---|---|---|
| {struct E_LOOPBACK | | | |
| usint frame_val; | | 0 | X (not for user) |
| usint frame_status; | frame time counter status | 2 | 8000H passed, 8001H failed |
| usint cmd_send[8]; | | 4 | cmd_send[0]: 5555H |
| | TX-A, command sync | 6 | cmd_send[1]:8000H passed, else failed |
| | | 8 | cmd_send[2]: 1234H |
| | TX-A, data sync | A | cmd_send[3]:8000H passed, else failed |
| | | C | cmd_send[4]: 5555H |
| | TX-B, command sync | E | cmd_send[5]:8000H passed, else failed |
| | | 10 | cmd_send[6]: 1234H |
| | TX-B, data sync | 12 | cmd_send[7]:8000H passed, else failed |
| usint ttag_val_lo | | 14 | 30D4H ± 2 |
| usint ttag_val_hi | | 16 | 0 |
| usint ttag_status; | time tag status | 18 | 8000H passed, 8001H failed |
| } *E_loopback; | | | |

June 2015, Rev A-2