SCRAMNet+ Network

# SC150 PCI Bus Hardware Reference

Document No. D-T-MR-PCI#####-A-0-A8

Systran®

# FOREWORD



Revised: April 25, 2003

**Systran Corporation**
4126 Linden Avenue
Dayton, OH 45432-3068 USA
(800) 252-5601 (U.S. only)
(937) 252-5601

**FCC**

This product is intended for use in industrial, laboratory or military environments. This product uses and emits electromagnetic radiation, which may interfere with other radio and communication devices. The user may be in violation of FCC regulations if this device is used in other than the intended market environments.

**CE**

As a component part of another system, this product has no intrinsic function and is therefore not subject to the European Union CE EMC directive 89/336/EEC.

# TABLE OF CONTENTS

# APPENDICES

# FIGURES

# TABLES

*This page intentionally left blank*

# 1. INTRODUCTION

## 1.1 How To Use This Manual

### 1.1.1 Purpose

This document is a reference manual for the SCRAMNet+ SC150 PCI host interface board. It provides a physical and functional description of the SCRAMNet+ SC150 PCI board. The manual describes how to unpack, set up, install and operate the hardware.

### 1.1.2 Scope

This information is intended for systems designers, engineers and network installation personnel. You need at least a systems level understanding of general computer processing, of memory and hardware operation, and of the specific host processor to effectively use this manual.

### 1.1.3 Style Conventions

- Hexadecimal values are written with a "0x" prefix. For example, 0x03FF
- Switch, signal and jumper abbreviations are in capital letters. For example, RSW1, J5, etc.
- Register bits and bit ranges are specified by the register identification followed by the bit or range of bits in brackets [ ]. For example, CSR6[4], CSR3[15:0], ACR[1,2]
- Bit values are shown in single-quotes. For example, set bit 15 to '1'

## 1.2 Related Information

- *SCRAMNet Network Programmer's Reference Guide* (Doc. Nr. D-T-MR-PROGREF) - A collection of routines to assist SCRAMNet users with application development.
- *SCRAMNet Network Utilities User Manual* (Doc. Nr. C-T-MU-UTIL) - A user's manual for the SCRAMNet Classic, SCRAMNet-LX, and SCRAMNet+ SC150e hardware diagnostic software, SCRAMNet+ SC150e EEPROM initialization software, and the SCRAMNet Network Monitor.
- *SCRAMNet Network Media User's Guide* (Doc. Nr. D-T-MU-MEDIA)—A description of network cabling hardware accessories for the SCRAMNet+ SC150e Network.
- All documentation related to the V360EPC PCI Bridge chip, including register specification can be obtained from the *V3 Semiconductor Inc.* web site at http://www.vcubed.com under Products, V3xxEPC, Documentation.

## 1.3 Quality Assurance

Systran Corporate policy is to provide our customers with the highest quality products and services. In addition to the physical product, the company provides documentation, sales and marketing support, hardware and software technical support, and timely product delivery. Our quality commitment begins with product concept, and continues after receipt of the purchased product.

Systran's Quality System conforms to the ISO 9001 international standard for quality systems. ISO 9001 is the model for quality assurance in design, development, production, installation and servicing. The ISO 9001 standard addresses all 20 clauses of the ISO quality system, and is the most comprehensive of the conformance standards.

Our Quality System addresses the following basic objectives:

- Achieve, maintain and continually improve the quality of our products through established design, test, and production procedures.
- Improve the quality of our operations to meet the needs of our customers, suppliers, and other stakeholders.
- Provide our employees with the tools and overall work environment to fulfill, maintain, and improve product and service quality.
- Ensure our customer and other stakeholders that only the highest quality product or service will be delivered.

The British Standards Institution (BSI), the world's largest and most respected standardization authority, assessed Systran's Quality System. BSI's Quality Assurance division certified we meet or exceed all applicable international standards, and issued Certificate of Registration, number FM 31468, on May 16, 1995. The scope of Systran's registration is: "Design, manufacture and service of high technology hardware and software computer communications products." The registration is maintained under BSI QA's bi-annual quality audit program.

Customer feedback is integral to our quality and reliability program. We encourage customers to contact us with questions, suggestions, or comments regarding any of our products or services. We guarantee professional and quick responses to your questions, comments, or problems.

# 1.4 Technical Support

Technical documentation is provided with all of our products. This documentation describes the technology, its performance characteristics, and includes some typical applications. It also includes comprehensive support information, designed to answer any technical questions that might arise concerning the use of this product. We also publish and distribute technical briefs and application notes that cover a wide assortment of topics. Although we try to tailor the applications to real scenarios, not all possible circumstances are covered.

Although we have attempted to make this document comprehensive, you may have specific problems or issues this document does not satisfactorily cover. Our goal is to offer a combination of products and services that provide complete, easy-to-use solutions for your application.

If you have any technical or non-technical questions or comments, contact us. Hours of operation are from 8:00 a.m. to 5:00 p.m. Eastern Standard/Daylight Time.

- Phone: (937) 252-5601 or (800) 252-5601
- E-mail: support@systran.com
- Fax: (937) 252-1349
- World Wide Web address: www.systran.com

# 1.5 Ordering Process

To learn more about Systran products or to place an order, please use the following contact information. Hours of operation are from 8:00 a.m. to 5:00 p.m. Eastern Standard/Daylight Time.

- Phone: (937) 252-5601 or (800) 252-5601
- E-mail: info@systran.com
- World Wide Web address: www.systran.com

*This page intentionally left blank*

# 2. SCRAMNET NETWORK

## 2.1 Overview

The SCRAMNet+ Network is a real-time communications network, based on a replicated, shared-memory concept. Each host processor on the network has access to its own local copy of shared memory that is updated over a high-speed, serial-ring network. The network is optimized for the high-speed transfer of data among multiple, real-time computers that are all solving portions of the same real-time problem. The SCRAMNet+ node board can automatically filter out redundant data.

## 2.2 Shared Memory

In its simplest form, the SCRAMNet+ Network system is designed to appear as general-purpose memory. The use of this memory depends only on the conventions and limitations imposed by the specific host computer system and operating system. On most processors, this means that the application program can use this memory in basically the same way as any other data-storage area of memory. The memory cannot be used as instruction space.

The major difference between SCRAMNet+ memory and system memory is that any data written into SCRAMNet+ memory is automatically sent to the same SCRAMNet+ memory location in all nodes on the network. This is why it is also referred to as replicated shared memory. A good analogy is the COMMON AREA used by the FORTRAN programming language. Where the COMMON AREA makes variables available to subroutines of a program, SCRAMNet+ makes variables available to processors of a network.

When a host computer writes to the shared memory, the proper handshaking logic is supplied by the SCRAMNet+ node host adapter. The shared memory behaves somewhat like resident or local memory.

A software driver is usually not required except for interrupt handling.

### 2.2.1 Dual Port Memory Controller

The Dual Port Memory Controller (see Figure 2-1) allows the host to read from or write to shared memory with a simultaneous network write to shared memory. Unless an interrupt has been authorized for that memory address, the host is not aware the network is writing to shared memory. This is why caching must be disabled for SCRAMNet memory. If an interrupt has been authorized, the interrupt will then be sent to the host processor.

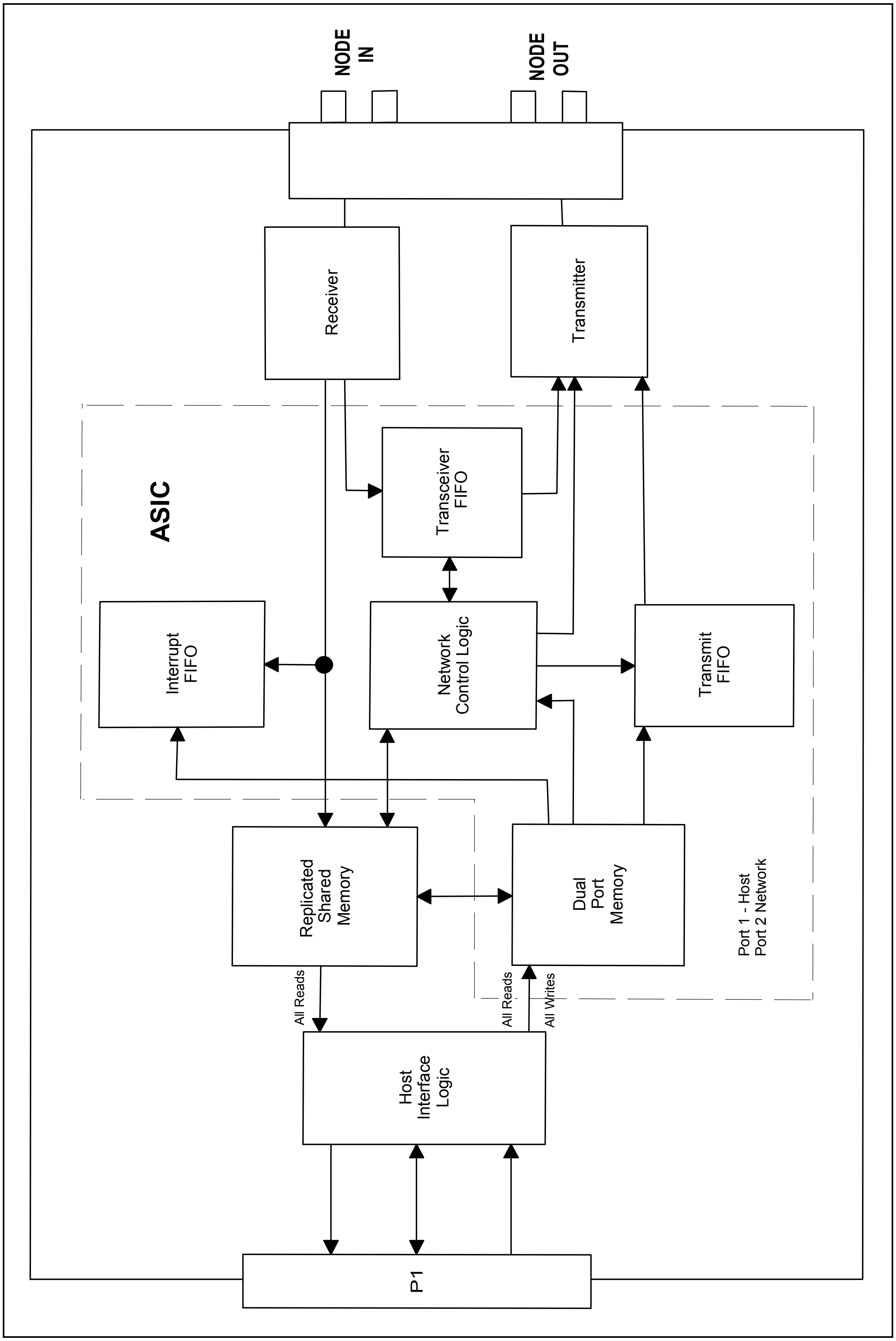



**Figure 2-1 Functional Diagram**

### 2.2.2 Control/Status Registers (CSRs)

The operation of the SCRAMNet+ board is controlled by Input/Output (I/O) CSRs. The location of the CSRs in the computer's address space is determined by the plug-and play BIOS in the host system. Address offsets for the CSRs can be found in Chapter 4, INSTALLATION.

Most modes of operation are set during initialization in registers embedded in the PCI target controller, and remain unchanged during run time.

### 2.2.3 Virtual Paging

The SCRAMNet+ network may include a variety of SCRAMNet+ nodes having varying amounts of shared memory. All SCRAMNet+ nodes use the same 8 MB shared memory map. This feature permits different SCRAMNet+ boards with 4 MB of shared memory or less to be paged into different sections of the 8 MB memory map. A board with a 4 MB or smaller memory may be located on any shared-memory address boundary that is an even multiple of itself (for example, 2 MB can page to 0, 2, 4 or 6 MB address).

## 2.3 FIFO Buffers

The SCRAMNet+ board contains various FIFO buffers used for temporarily storing information during normal send and receive operation of the node. Refer to Figure 2-1.

### 2.3.1 Transmit FIFO

The Transmit FIFO is a message holding area for native messages waiting to be transmitted. Each host write to SCRAMNet+ memory may constitute a write to the Transmit FIFO. (Data Filtering and HIPRO features may interfere with this.) Each write to the Transmit FIFO contains 21 bits of address (A22-A2), 32 bits of data, and one bit of interrupt information. The Transmit FIFO can hold up to 1024 writes before becoming full.

When the Transmit FIFO reaches a FULL condition (CSR1[0] ON), one more host write could cause a message to be lost. To prevent this, the CSR-controllable, built-in SCRAMNet+ feature called HOLDOFF mode extends the computer write cycle until the Transmit FIFO is able to empty at least one message.

### 2.3.2 Transceiver FIFO

This buffer is used to receive foreign messages from the network, and send them on, or to hold received foreign messages while inserting a native message from the host onto the network.

Each node is responsible for receiving foreign messages, writing them to its copy of shared memory, and re-transmitting the message to the next node.

### 2.3.3 Interrupt FIFO

The Interrupt FIFO contains a 21-bit address (A22 - A2) and a retry-status bit for each shared-memory-based interrupt received. The Interrupt FIFO can hold 1024 interrupt addresses. This FIFO can be read using CSR4 and CSR5.

### 2.3.4 Receiver FIFO

The Receiver FIFO is designed as a temporary holding place for incoming foreign messages while the shared memory is busy servicing a host request. This FIFO is three messages deep, and is designed so it can never be overrun. Each item in the Receiver

FIFO contains 21 bits of address (A22 - A2), 32 bits of data, and one incoming interrupt bit. When the messages are 1024 bytes, the initial header information data stays in the FIFO, the subsequent 4 bytes of data are loaded in, and the address is incremented by four.

# 2.4 Network Ring

The SCRAMNet+ Network is a ring topology network. Data is transmitted at a rate of 150 Mbits/s over dual fiber-optic cables. The two lines together produce the incoming data clock. Due to the network speed and message packet size, the network can accommodate over 1,800,000 message packets passing by each node every second. There is an approximate 247 ns (minimum) delay at each node as the message packet works its way around the ring. The maximum delay depends on the selection of fixed-length or variable-length message packets. A fixed-length message packet has a maximum delay of 800 ns, a 256-byte variable-length message packet is 16.2 µs, and a 1024-byte variable-length message packet is 62 µs. Delay can be imposed when a node must complete the transmission of a native message packet before retransmitting a foreign message packet. A SCRAMNet+ Network can accommodate up to 256 nodes per network ring.

## 2.4.1 Protocol

The protocol is a register-insertion methodology and is <u>NOT</u> a token ring. Depending on the protocol selected, all message packets are the same size or are variable (as in the PLUS modes), and multiple nodes can transmit data simultaneously. There is no master node, and all nodes have equal priority for network bandwidth. The message protocol is designed specifically for real-time applications where data must be passed very rapidly. When the node operates in BURST or BURST PLUS mode, the node will never re-transmit its own messages for error correction. When operating in PLATINUM or PLATINUM PLUS mode, error detection is enabled, and re-transmission can occur.

### BURST MODE

BURST mode is an open loop, non-error-corrected communication mode. This mode allows multiple 82-bit messages (46-bit header plus 32-bite data and four parity bits) per node on the ring at a time. The limited-message-packet length enhances the data latency characteristics of the network by providing the shortest possible media access delay. The messages are transmitted as fast as the system will allow.

### PLATINUM MODE

PLATINUM mode is BURST mode with error correction enabled. The messages are transmitted as fast as the system will allow, but error checking is used to detect and re-transmit corrupted message packets.

### PLUS MODES

The PLUS mode protocol enhancement can increase the maximum network throughput from 6.5 MB/sec to approximately 15.2 to 16.7 MB/sec by the use of variable-length message packets. Each SCRAMNet+ message packet has a 46-bit header plus the data. The user-selectable maximum message packet size increases the data size from the normal 32 bits to either 256 or 1024 bytes of data. Data must be written to sequential longword addresses.

## 2.5 Auxiliary Control RAM (ACR)

The Auxiliary Control RAM (ACR) provides a method of external triggering and interrupt control by offering a choice of four actions to occur when a particular SCRAMNet+ shared-memory address is written into. Each shared-memory location has its own action or set of actions associated with it.

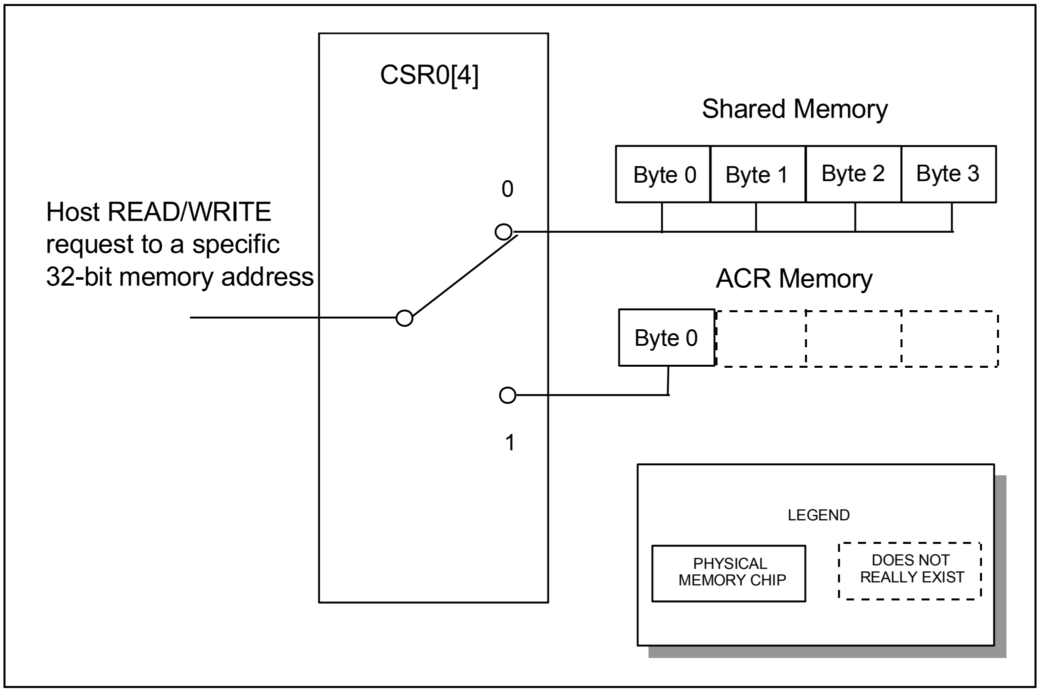



**Figure 2-2 ACR/Memory Access**

In Figure 2-2, host CPU read/write operations are channeled to either SCRAMNet+ memory or to the ACR. The ACR is a physically separate memory from the shared memory. Channeling is based on a user-controlled switch setting and may be toggled to the desired position by writing to a bit in the SCRAMNet+ CSR. When access to the ACR is enabled, shared memory is not accessible by the host and the ACR byte is viewed as the least significant byte (LSB) of every shared-memory four-byte address. The ACR bits define what external trigger and/or interrupt action(s) are to be taken whenever writing to any byte of the SCRAMNet+ shared memory 4-byte word.

Only five bits of the ACR are associated with every four-byte word of shared memory (on even four-byte boundaries). The other 27 bits of the ACR are phantom bits and do not physically exist.

## 2.6 Interrupts

SCRAMNet+ allows a node processor to receive interrupts from and transmit interrupts to any node on the network, including the originating node, provided the receiving node is set up to receive an interrupt message. Interrupts can be generated under two different conditions:

- SCRAMNet+ Network data writes to shared memory; and
- SCRAMNet+ network errors detected on the local node.

SCRAMNet+ interrupts usually require a device driver to interface with the node processor. The driver is required primarily to permit the host processor to handle interrupts from the SCRAMNet device.

## 2.6.1 Network Interrupt Writes

### FOREIGN MESSAGE

The node can receive a message from another node with the interrupt bit set. If Receive Interrupt Enable ACR[0] and Interrupt Mask Match Enable CSR0[5] are enabled, the data is written to shared memory and the address is placed on the Interrupt FIFO.

### NATIVE MESSAGE

If the message received was originated by the node, and Write Own Slot Enable CSR2[9] and Enable Interrupt on Own Slot CSR2[10] are enabled, the host has authorized a Self-Interrupt. The data is written to shared memory and the address is placed on the Interrupt FIFO.

Network Interrupt writes can be accomplished by two methods:

- **Selected**. Data writes to selected shared memory locations from the network.
- **Forced**. Any data writes to any shared memory from the network.

In either case, the node can be configured to write to itself. This condition is called "Self Interrupt".

## 2.6.2 Selected Interrupt

The selected-interrupt method requires choosing SCRAMNet+ shared-memory locations on each node to receive and/or to transmit interrupts. These shared-memory locations may also be used to generate signals to external triggers. The procedure for selecting shared-memory locations for interrupts and/or external triggers is explained in the paragraph on the Auxiliary Control RAM, paragraph 2.5.

### OUTGOING INTERRUPT

The Outgoing Interrupt is described in Figure 2-3. If both Transmit Interrupt Enable ACR[1] and Network Interrupt Enable CSR0[8] are set, and a data item is transmitted to any of the selected-interrupt memory locations, then an interrupt message is sent out on the network. This message will generate interrupts to any processors on the network that have that same shared-memory location selected to receive interrupts.

### INCOMING INTERRUPT

Figure 2-4 demonstrates the process of receiving a message with the interrupt bit set. The data is written to shared memory and the address is placed in CSR5 and CSR4 to await being sent to the host. If the Receive Interrupt Enable ACR[0], Host Interrupt Enable CSR0[3], and the Interrupt Memory Mask Match Enable CSR0[5] are set, and network interrupt data is received for any one of the selected interrupt memory locations the following occurs:

- The data is stored in that location
- The SCRAMNet+ address of the memory location is placed on the Interrupt FIFO queue, and
- An interrupt is sent to the processor.

### NETWORK ERRORS

The Interrupt on (Network) Errors mode is enabled by setting CSR0[7] ON. Network errors are defined in CSR1 according to an interrupt mask set in CSR9. When an incoming foreign message generates an interrupt, there is no way to mask the interrupt according to the content of the message. However, specific error conditions may be identified.

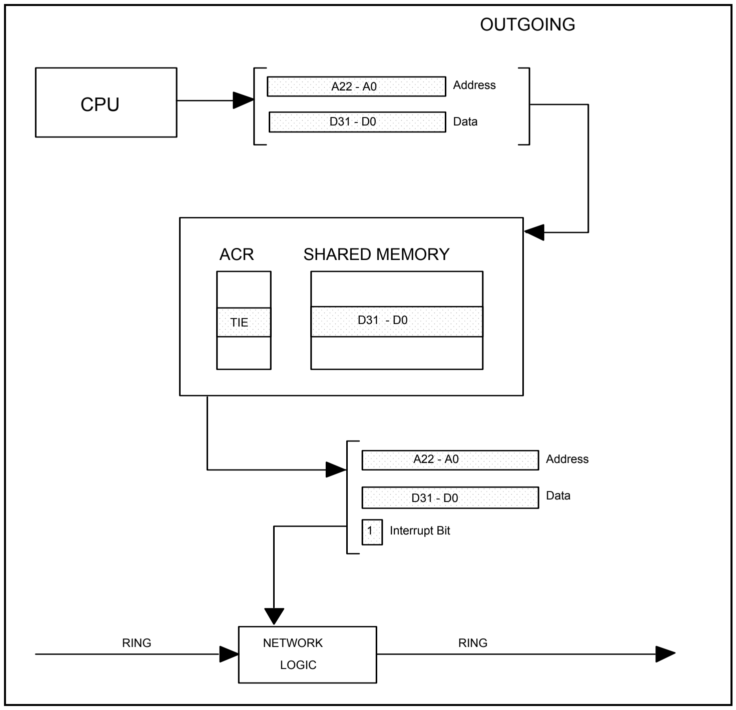



**Figure 2-3 Outgoing Interrupt**

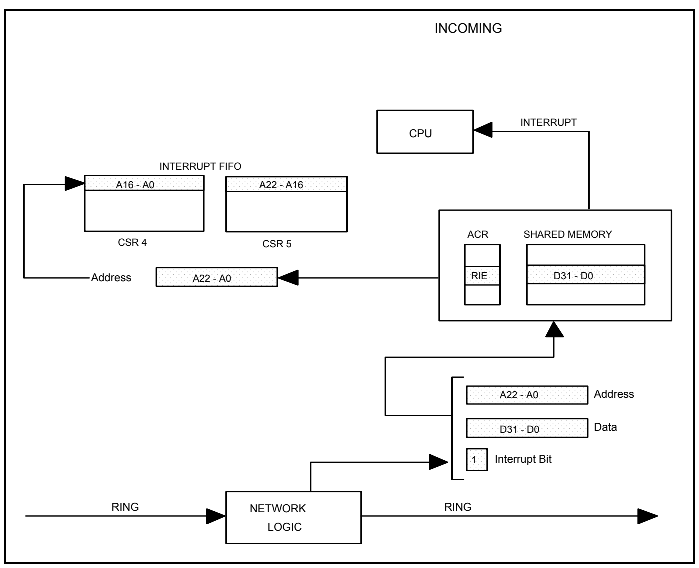



**Figure 2-4 Incoming Interrupt**

Error conditions are listed in CSR1 and may be masked by setting the corresponding bit in CSR9. If the Mask bits in CSR9 are all set to ‘1’, any error will generate an interrupt. Otherwise, only errors with a ‘1’ in the appropriate Mask bit will generate an interrupt.

## 2.6.3 Forced Interrupt

The forced-interrupt method works the same as for selected-interrupt except for the choice of interrupt locations. All shared-memory locations are automatically set up to receive and/or transmit interrupts depending upon the ACR override conditions set in CSR0[6] and/or CSR0[9].

When Override Receive Interrupt Enable CSR0[6] is set, an interrupt will be sent to the host by any network-interrupt-data message , regardless of the status of the ACR Receive Interrupt bit.

When Override Transmit Interrupt Enable CSR0[9] is set, an interrupt will be sent out on the network regardless of the status of the ACR Transmit Interrupt bit.

A third condition, Receive Interrupt Override CSR8[10], is used to designate all incoming network traffic as interrupt messages. The network message interrupt bit does not need to be set.

## 2.7 External Triggers

Two external triggers are supported by SCRAMNet+ SC150 PCI. The external triggers will occur only if the ACR has been configured to enable them. Triggers 1 and 2 are generated by SCRAMNet+ shared-memory access. Both triggers generate a 26.64 ns TTL level compatible, non-terminated, output.

- **Trigger 1 -** Host Read/Write (ACR[2] enables)
- **Trigger 2 -** Network Write (ACR[3] enables)

## 2.8 General Purpose Counter/Global Timer

The General Purpose Counter/Timer has six modes of operation controlled by CSR8 and CSR9, the output from the General Purpose Counter/Timer is stored in CSR13. Counter modes can count errors, external trigger events, or network messages. A high-resolution timer mode can run free or measure the ring time with a 26.66 ns resolution.

The global timer mode clocks with a resolution of 1.706 µs and resets on an external trigger event. (See 2.7: External Triggers). A specific shared-memory location may be identified with External Trigger 2 ACR[3] so that a memory write from a single node on the network can simultaneously reset all the global timers in the ring.

If the Trigger 2 event is the frame counter, the timers in the ring effectively become synchronized sub-frame timers, which can then be used to tag time-critical data or to measure and compare the completion time of various tasks within a distributed real-time system.

# 2.9 LED Status Indicators

## 2.9.1 Network Access

### INSERT LED

The green Insert LED is ON when the node is Inserted into the SCRAMNet+ Network ring.

### CARRIER DETECT LED

The green Carrier Detect LED is ON when there is a valid pair of transmit lights from the previous SCRAMNet+ node into this node’s receiver pair. Assuming at least one node is inserted in the ring, if the fiber-optic cables are connected and the Carrier Detect LED is OFF, then the ring integrity is NOT valid. This condition indicates improper fiber-optic cabling or problems with the down-line node's transmitter(s).

## 2.9.2 Internal Access

A set of eight bi-color LED indicators are available for troubleshooting and fine tuning of the node. Since there are no remote sensors, the chassis must be open to observe the LEDs.

Four “HOST” LEDs indicate read and write transactions to shared memory or to control/status registers, an indicator to show receipt of a host acknowledgment, and indication of a memory or error interrupt.

Four “NETWORK” LEDs show message waiting; and receipt of an error, a foreign message, or a native message.

# 2.10 Modes of Operation

## 2.10.1 Data Filter Mode

When SCRAMNet+ Data Filtering is enabled, only those writes to SCRAMNet+ memory that produce a data change are transmitted to the network.

### EXAMPLE:

If location 1000 in SCRAMNet+ memory contains the value ‘20’ and the host processor writes the value ‘20’ to location 1000, then no network traffic will be generated. However, if any other value is written to location 1000, then the new value will be passed around the network to update the other SCRAMNet+ node memories.

When a write is received from the host, a comparison is made to the old data at that address to see if there was a change before writing to shared memory. If the data has changed, then it is written to shared memory, and is also transmitted onto the network. This entire process is completed within the host memory standard bus write cycle.

Data filtering is a powerful communications compression technique for cyclical applications. This technique has been shown to significantly reduce the network traffic and therefore increase the effective throughput on the network.

## 2.10.2 High Performance (HIPRO) Mode

HIPRO provides an efficient means to transmit 8-bit and 16-bit data transactions as one 32-bit network write. It also provides a means of keeping 32-bit data from becoming fractured.

### EXAMPLE #1:

A floating-point length numeric sent in 8-bit or 16-bit pieces may not be accurately re-assembled at the destination.

### EXAMPLE #2:

The receiving node may otherwise try to use part or half of such a value before the entire 32 bits is received.

### HIPRO WRITE

The SCRAMNet+ network message is based on 32-bit longword data. This means if any 8-bit field of the 32-bit buffer is changed, the entire 32-bit message is transmitted. If a host is limited to only 8-bit or 16-bit databus transactions the network throughput is quartered or halved, respectively.

HIPRO mode permits a 32-bit location to be set up in shared memory such that any initial write smaller than 32 bits to that location will not automatically go onto the network. The 32-bit write to the network will only occur when all four bytes within the 32-bit location have been written through subsequent writes by the host CPU. This can be accomplished by four consecutive 8-bit or two consecutive 16-bit writes to the SCRAMNet memory.

**NOTE**: HIPRO WRITE will not work if Disable Host to Memory Write CSR2[8] is set, or when writing two separate shortwords while using interrupts.

## 2.10.3 Holdoff Mode

It is possible that the Transmit FIFO can become full when the host is writing to the SCRAMNet+ interface faster than the network can absorb the data.

In Holdoff mode, the host write cycle is automatically extended until the SCRAMNet+ Transmit FIFO buffer transmits at least one message. This prevents the loss of data and is transparent to the user.

In some system designs, and on some computer buses, it is not desirable or effective to have the write cycle lengthened to match network throughput—even at the expense of possible data loss across the network. In this case this option may be disabled by setting CSR8[1] ON. Transmit FIFO 7/8 Full CSR1[2] can then be used to control the data flow via software control.

## 2.10.4 Loopback Modes

Loopback mode is used for testing, and for routing data, which would normally be transmitted onto the network back into the node. This mode is used to check performance internally (Wire Loopback) at the Media Card (Mechanical Switch Loopback) and Transmit/ Receive (Fiber-optic Loopback).

### WIRE LOOPBACK MODE

The Wire Loopback mode needs no manual external modifications to work. Wire Loopback is enabled by setting CSR2[7] ON. This mode checks the on-board circuitry for continuity.

**NOTE**: If a node is inserted into the network while in wire loopback mode, it will create a break in the network ring, making all nodes down-line unreachable.

### MECHANICAL SWITCH (MEDIA CARD) LOOPBACK MODE

Mechanical Switch (Media Card) Loopback mode is enabled by setting Mechanical Switch Override CSR8[11] to OFF. This test is used to check the circuitry onto the Media Card but excludes the fiber-optic circuitry. In this test the signal does not leave the Media Card.

### FIBER-OPTIC LOOPBACK MODE

The Fiber-optic Loopback mode must have the optional Fiber Optic Bypass Switch connected, Disable Fiber-optics Loopback CSR2[6] set to OFF (power up default), and Insert Node CSR0[15] enabled to be valid. When the Fiber-optic Loopback mode is in effect, the output of the transmitter is connected by fiber optics directly to the input of the receiver, and the receiver is disconnected from the network.

The optional Fiber Optic Bypass Switch must be installed for this loopback to work. However, in the absence of the Fiber Optic Bypass Switch, fiber-optic cables could be run from the node's transmitter output connectors to the receiver input connectors. This configuration, with Insert Node enabled, would constitute a Fiber-optic Loopback mode for stand-alone testing. Set CSR2[6] ON to disable the Fiber-optic Loopback mode when the node is in use as a part of the network. This configuration is not a substitute for the Fiber Optic Bypass Switch for network operation.

## 2.10.5 Write-Me-Last Mode

The Write-Me-Last mode of operation allows the originating node to be the last node in the ring to have the data deposited to its memory. This can be useful for synchronization. This means that when the host performs a write to the SCRAMNet+ shared memory, this data is not immediately written to the host node's memory, but is first sent to the other nodes on the network. When the message returns to the originating node it is written to shared memory, and is then removed from the network ring.

Therefore, host-originated data written to shared memory travels the ring updating the SCRAMNet+ node memories on the ring and, upon returning to the originating node, that node writes the data to its own shared memory as the last node on the ring. This guarantees that the data is available on all other nodes.

*This page intentionally left blank*

# 3. PRODUCT OVERVIEW

## 3.1 Overview

SCRAMNet+ (Shared Common Random Access Memory Network) is a communications network geared toward real-time applications, and based on a replicated, shared-memory concept.

The SCRAMNet+ SC150 interface node board is backwards compatible with the original SCRAMNet Classic product with the exception of the GOLD Ring communication protocol.

The SCRAMNet+ SC150 PCI board requires one a single slot in the computer chassis.

**CAUTION**: Check with the host computer manufacturer to find out which slots are available for third party PCI memory cards before installing the SCRAMNet+ Network board into any system. Installing any type of PCI card in a non-standard PCI slot may result in serious damage to the host machine.

The board is available with 4 KB or 128 KB on-board shared memory. Shared memory can be upgraded to 512 KB, 1 MB, 2 MB, 4 MB or 8 MB random access memory (RAM). Installing any memory upgrade overrides the on-board 4 KB (or optional 128 KB) memory.

## 3.2 Network Features

- A ring topology with 150 Mbit/s line transmission rate.
- A "Data-Filter" that allows only data stored in shared memory that has changed to be passed to the network for communications to the other nodes.
- Field Upgrade Memory Options up to 8 MB of replicated, shared memory for each node processor.
- BURST Mode protocol (Error Correction Disabled) with fixed-length message packets of 82-bits.
- BURST PLUS Mode communication based on variable packet size to a maximum of either 256 bytes or 1024 bytes.
- PLATINUM Mode protocol (error correction enabled) with fixed-length message packets of 82-bits.
- PLATINUM PLUS Mode communication based on variable-length message packet size to a maximum of either 256 bytes or 1024 bytes.
- 256-node capacity on each ring.
- No operating or system software required to support network protocol.
- No network-dependent application software required.

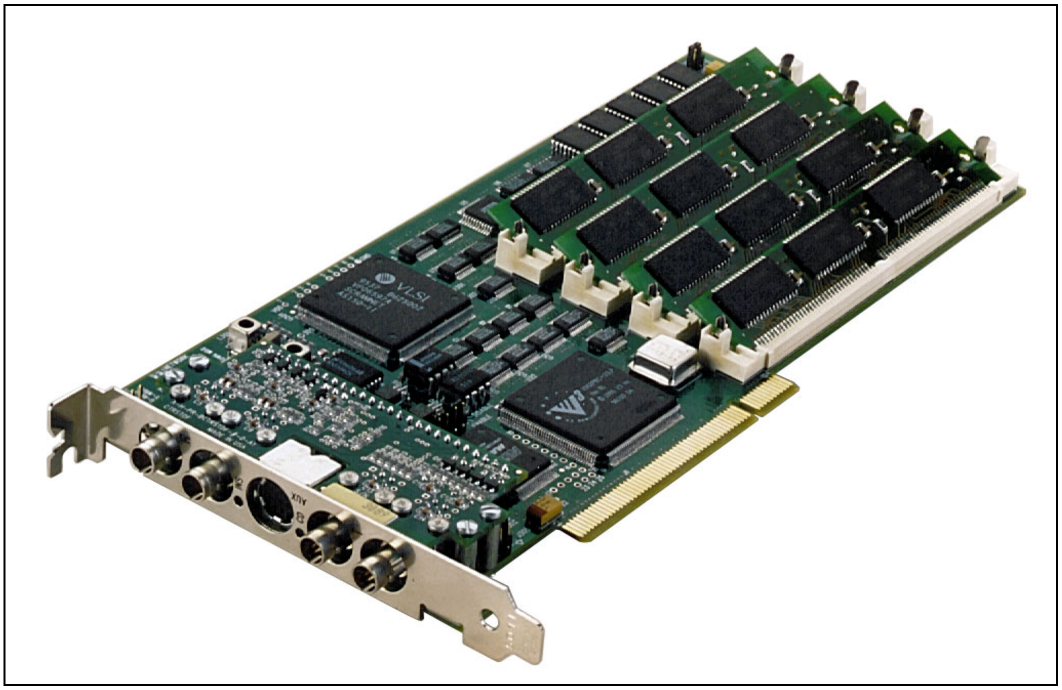


**Figure 3-1 SC150 PCI Board, Version C1**

## 3.3 Options

- Optional paired fiber-optic or coaxial transmission media
- Fiber Optic Bypass Switch for ring continuity when node power is off.
- Quad Switch—A switching control device that controls up to four nodes or sub-rings, eliminates the need for a separate Fiber Optic Bypass Switch, and functions as a repeater.

## 3.4 PCI Board Features

- Custom SIMM memory upgrade option.
- General-purpose Counter.
- Error Interrupt Mask.
- Dynamic shared-memory addressing.
- Virtual paging for Shared Memory (CSR selectable).
- Variable-length message packet capability.
- Dual-port memory.
- Dual memory and error interrupt.
- Single-Slot Solution.

## 3.5 PCI Specification Level

The SCRAMNet+ SC150 PCI board was designed in accordance with the PCI specification 2.1.

## 3.6 Hardware

- 32-bit PCI compliant.
- Supports bus-level interrupt INTA.

- Maximum bus clock speed is 33 MHz.
- Less than 1.5 amps @ 5 volts only.
- Single-slot connection.

# 3.7 PCI Controller

The SCRAMNet+ SC150 PCI card uses the V96xPBC PCI Bridge chip from V3 Semiconductor. All documentation related to this device, including register specification, can be obtained from the *V3 Semiconductor Inc.* web site at http://www.vcubed.com/.

# 3.8 PC Software

The objective of the software effort for the PC-Clone platforms, is to provide one software package that is adaptable to all of the environment combinations described below:

- PCI Bus
- SCRAMNet Classic/SCRAMNet-LX/SCRAMNet+ SC150
- 32 bit Address Bus, 4 GB Max Address

# 3.9 Utility Software

## 3.9.1 SCRAMNet Diagnostics

The SCRAMNet Network Hardware Diagnostics are designed to test the functionality of the hardware. This suite of tests will detect whether it is testing a SCRAMNet Classic board or a SCRAMNet-LX/SCRAMNet+ board and adjust the test menus accordingly.

## 3.9.2 EEPROM Initialization (EPI)

The EEPROM Initialization program is a SCRAMNet+ utility used to simplify configuration of the network node. The EPI program will store a start-up configuration in the serial EEPROM, which can initialize the node on power up. This initialization program can be run when the board is installed to set the desired power-up state of the SCRAMNet+ node. EPI is completely menu driven and contains a context-sensitive help feature.

## 3.9.3 SCRAMNet Monitor

The SCRAMNet Monitor allows viewing and editing of memory and CSR locations on the SCRAMNet node. This utility is useful during software development to verify that the correct values are being written to SCRAMNet memory and CSRs.

# 3.10 Options

## 3.10.1 Electronic Bypass Switch

The Electronic Bypass Switch exists on some Media Cards. This switch allows for fast bypass on power-fail conditions. The electronic switch operates in the low nanosecond range compared to a 20-millisecond time for a typical mechanical switch.

In case of node power failure, the electronic switch restores the network so quickly that only one or two messages will have to be retransmitted, whereas a mechanical switch could cost an excessive amount of transmission time re-sending perhaps thousands of messages.

## 3.10.2 Quad Switch

The SCRAMNet Quad Switch is designed to provide configuration control over the network topology and computing resources. The Quad Switch allows local clusters of up to four SCRAMNet nodes to be switched in or out of a primary SCRAMNet ring, independently and dynamically (Figure 3-2). It also allows sharing of a critical real-time resource between multiple systems.

The Quad Switch performs other useful functions such as optical bypassing, fiber-optic repeating to gain transmission length beyond the SCRAMNet node's transmission power limit, and to act as a media converter.

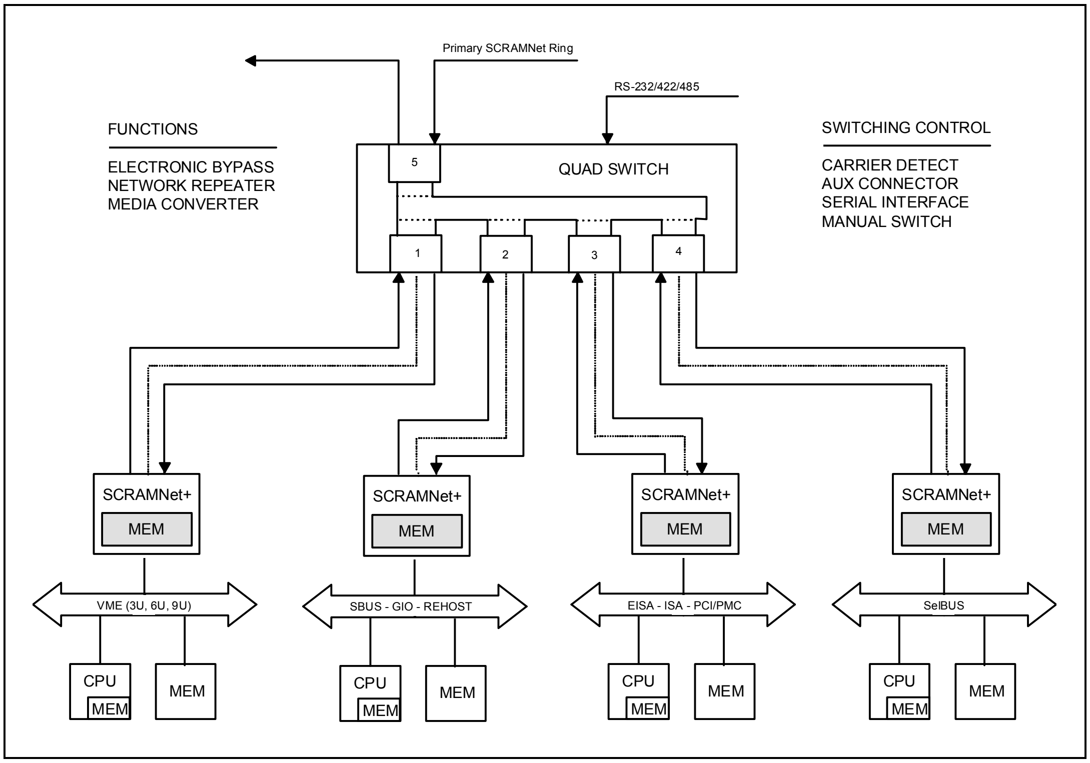



**Figure 3-2 Node Inclusion and Isolation**

The electronic bypass switching action is very fast, introducing a total network disruption of about one microsecond. This is over 10,000 times faster than mechanical optical bypass switches, and permits ring re-configuration to be performed in real-time with minimal impact on the system.

As a repeater, each Quad Switch port converts optical signals to electrical signals. These signals are re-synchronized and re-transmitted. This allows each connection to the Quad Switch to be the maximum length for the type of media selected.

The Quad Switch can also perform media conversion. Since each port has a Media Card just like a SCRAMNet node, each port can be configured to handle coaxial, standard link or long link fiber. This allows a signal to arrive on one media type, and go out on another.

Visually, LED's signify the state of node inclusion in the ring and if carrier is detected. If carrier is not detected, the port is put into Isolate state and the port is bypassed thereby retaining ring integrity. The auxiliary connector and the associated control cable links the port to the node to allow the application running the node to switch the Quad Switch in and out of Include or Isolate state. A manual Include/Isolate switch is used to guarantee that a node is isolated or that it can be included. A serial-port interface is used to send message packets to the Quad Switch to perform control functions or to obtain switch status remotely via the RS-232 or RS-422/485 connection. Two mechanical rotary switches are used to set the serial-interface address.

*This page intentionally left blank*

# 4. INSTALLATION

## 4.1 Installation Procedures

Installation of the PCI board includes the following:

- Unpack the board
- Visually inspect the board
    - Check SIMM connection
    - Check Media Card connection
- External configuration
    - Set/verify EEPROM WRITE jumper (J303)
    - Set/verify EEPROM READ jumper (J304)
    - Set/verify SIMM jumper (J2)
    - Set/verify Ground jumper (J305)
- Install the board
- Select Cabling Options
- Install Fiber Optic Bypass Switch (optional)
- Internally Configure the board
    - Set up SCRAMNet+ SC150 PCI Registers
    - Set up SCRAMNet+ Control/Status Registers
    - Set Node Identification (CSR3[15:8])
    - Set Network Time-out (CSR5)
    - Enable WRITE POSTING (CSR10[0])

# 4.2 Unpack the Board

Perform the following steps:

1. The PCI host card is wrapped in an anti-static bag and encased in anti-static foam.

**CAUTION**: Exercise care regarding the static environment. Use an anti-static mat connected to a wristband when handling or installing the SCRAMNet+ board

2. Remove the PCI anti-static bag from the carton.
3. Open the anti-static bag and remove the PCI host card.

Save the shipping material in case the SCRAMNet+ board needs to be returned.

The optional fiber-optic cables and Fiber Optic Bypass Switch are shipped in separate cartons.

# 4.3 Visually Inspect the Board

Check the board for any damage that may have occurred during shipping. In the event that any shipping damage has occurred, call Systran Customer Service at (937) 252-5601.

## 4.3.1 DEC-specific Board

If the board is a DEC-specific board:

- It will have the part number H-AS-DPCIDXXX-X0 (where "X" varies with memory and media options.)
- The U21 chip will be labeled "PCIDEC".
- EEPROM U20 will be labeled: (Where # stands for the EEPROM revision.)

V962
A#

## 4.3.2 Non-specific Board

If the board is a non-specific board:

- It will have the part number H-AS-DPCINXXX-X0 (where "X" varies with memory and media options.)
- The U21 chip will be labeled "PCICTRL4".
- EEPROM U20 will be labeled: (Where # stands for the EEPROM revision.)

SN V3
A#

Figure 4-1 represents a diagram of the SCRAMNet+ SC150 PCI Host Card layout.

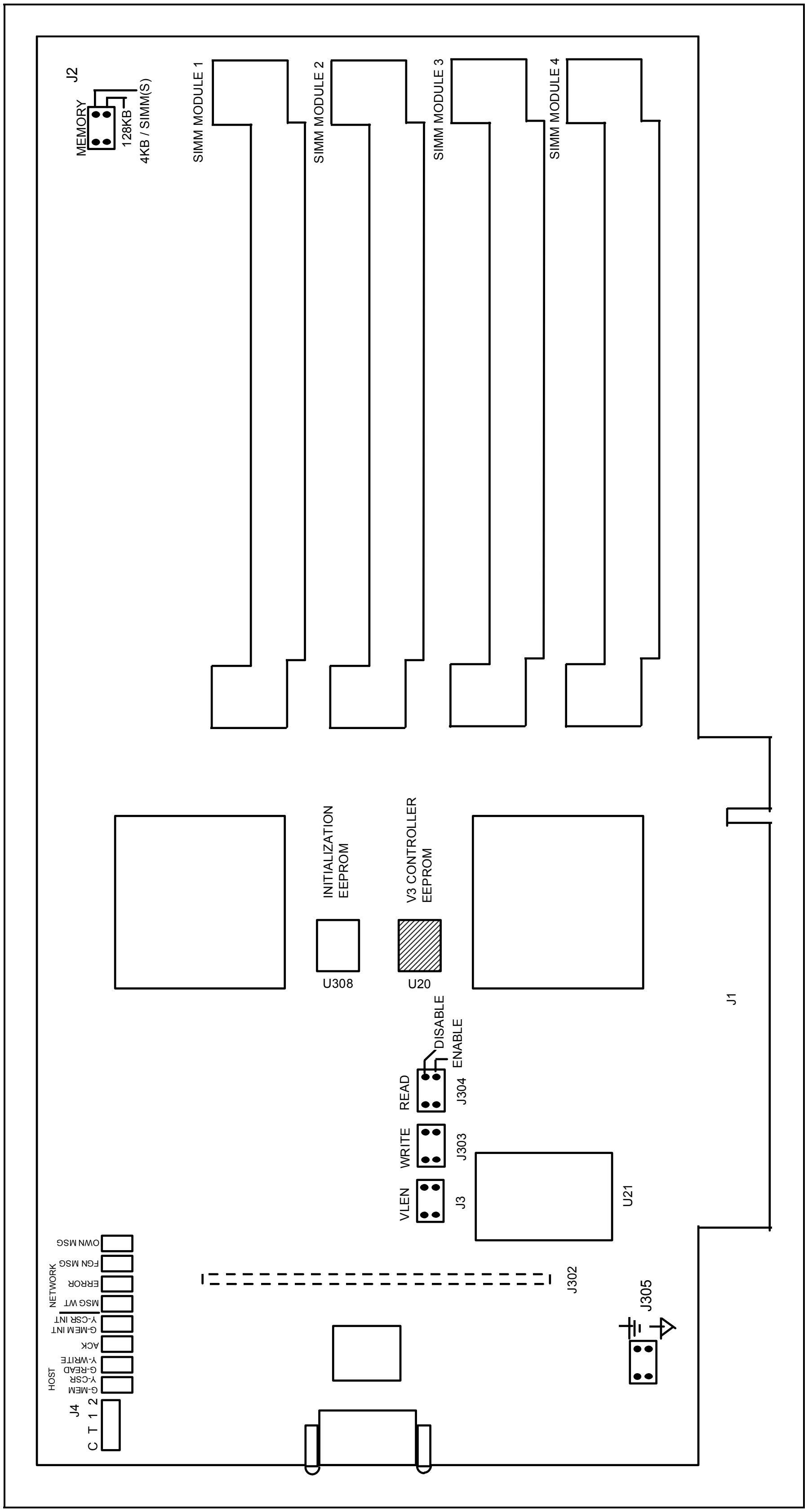



**Figure 4-1 SC150 PCI Layout**

## 4.3.3 Check SIMM Connections

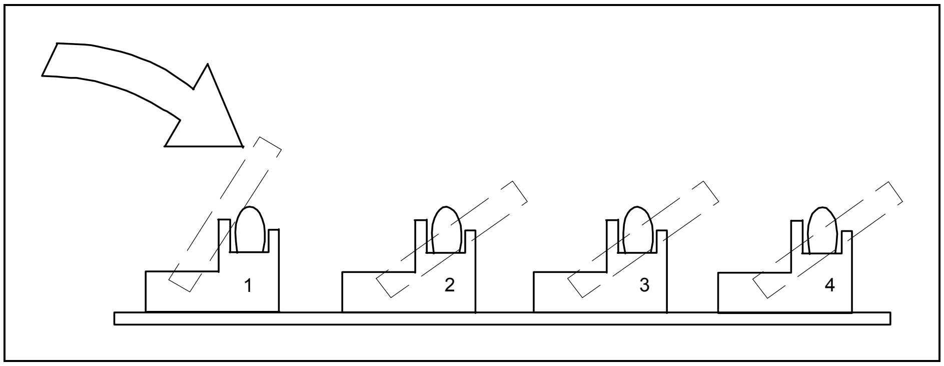



**Figure 4-2 SIMM Installation**

To install SIMMs, set the SIMM in the slot and gently press back and down until the clips snap into place as shown in Figure 4-2.

To remove the SIMMs, push the clips gently to the outside with each thumb while gently pulling the SIMM toward you with your index fingers.

In the case where no SIMMs are installed, shared memory defaults to the on-board 4 KB or optional 128 KB (see 4.4.2 Set/Verify Memory Jumper (J2)). If SIMMs are to be installed, the options are:

- Low density, 512 KB SIMMs
  1 = 512 KB (SIMM 1)
  2 = 1 MB (SIMM 1 and 2)
  4 = 2 MB (SIMMs 1 through 4)

- High density, 2 MB SIMMs
  1 = 2 MB (SIMM 1)
  2 = 4 MB (SIMM 1 and 2)
  4 = 8 MB (SIMMs 1 through 4)

SIMMs must all be either low density or high density; they cannot be mixed. The SIMMs are proprietary and must be ordered from Systran.

## 4.3.4 Media Card

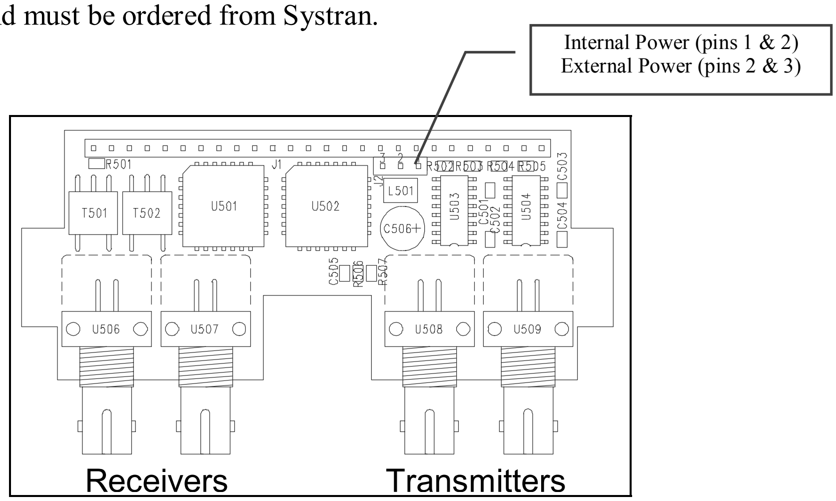



**Figure 4-3 Fiber-optic Media Card (Bottom view)**

The media card can have either coaxial or fiber-optic connectors. Figure 4-3 shows the fiber-optic media card. The SCRAMNet+ SC150 PCI board will support either option.

The type of media card installed will be dictated by the network configuration. There are two receive connections (Rx1 and Rx2) and two transmit connections (Tx1 and Tx2).

# 4.4 External Configuration

## 4.4.1 Set/Verify VLEN and EEPROM Jumpers

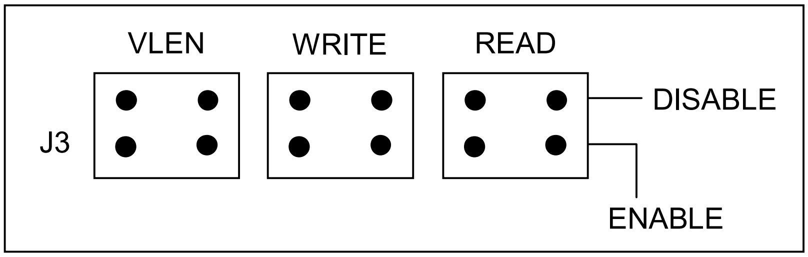



**Figure 4-4 VLEN and EEPROM Jumpers (J3)**

### SET/VERIFY VLEN JUMPER

If variable-length message packets will be used on the network, enable VLEN by installing a 2-pin header on the bottom row of the VLEN jumper.

Factory default setting: ENABLED

### SET/VERIFY EEPROM WRITE

To enable EEPROM write, install a 2-pin header on the bottom row of the write jumper.

Factory default: ENABLED

### SET/VERIFY EEPROM READ

To enable EEPROM READ install a 2-pin header on the bottom row of the READ jumper.

Factory default: ENABLED

## 4.4.2 Set/Verify Memory Jumper (J2)

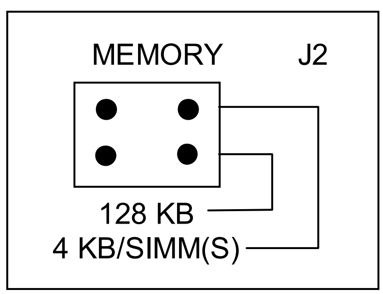



**Figure 4-5 Memory Jumper (J2)**

If no SIMMs are installed, two options are available

- 4 KB memory—install a 2-pin header on the top row.
- 128 KB memory—install a 2-pin header on the bottom row.

If SIMMs are installed, install a 2-pin header on the top row.

## 4.4.3 Set/Verify Ground Jumper (J305)

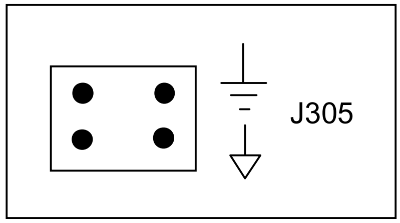



**Figure 4-6 Ground Jumper (J305)**

To select CHASSIS (Earth) ground, install a 2-pin header on the top row. To select SIGNAL ground, install a 2-pin header on the bottom row

Factory default setting: CHASSIS ground.

## 4.4.4 External Triggers

The SCRAMNet+ board generates two external triggers. Activating the triggers for any SCRAMNet memory location will cause an external trigger to be generated when the SCRAMNet memory location is accessed (Figure 4-7, Table 4-1 and Table 4-2).

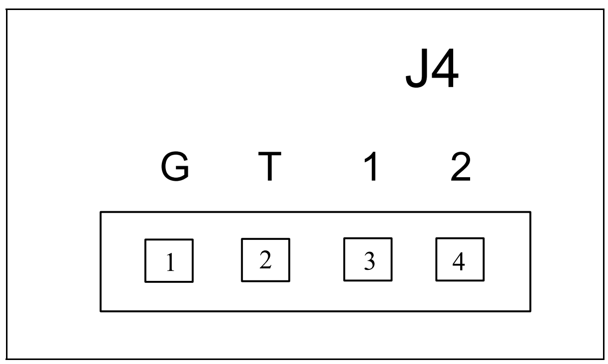



**Figure 4-7 External Trigger Connections (J2)**

**Table 4-1 Trigger Pin Connections (J2)**

| Pins | Label | Output |
|---|---|---|
| 1 | G | GROUND |
| 2 | T | TRIG1 or TRIG2 |
| 3 | 1 | TRIG1 |
| 4 | 2 | TRIG2 |

**Table 4-2 External Trigger Actions**

| Trigger | ACR | Action |
|---|---|---|
| 1 | Bit 2 | Host Read/Write |
| 2 | Bit 3 | Network Write |

# 4.5 Install the Board

Node configuration is pre-set or defined in the PCI Configuration File. Therefore, after the board has been inspected, and the SIMM and Ground jumpers have been verified, it is ready to be installed.

**CAUTION**: Make certain that the power to the host computer is OFF.

1. Remove the cover from the host machine.
2. Remove the bulkhead cover plate where the SCRAMNet+ host card will be.
3. Install the host card into the PCI bus slot.
4. Connect the Transmitter and Receiver fiber-optic cables to the SCRAMNet+ card.

# 4.6 Cabling Options

Options include installation of coaxial or fiber-optic cable. Media cards can be ordered for either one.

## 4.6.1 Coaxial Cable Configuration

SCRAMNet+ coaxial cable is composed of paired, shielded conductors terminated with SMA connectors. Maximum node separation using coaxial is 30 meters. The recommended coaxial cable is RG-58.

## 4.6.2 Fiber-optic Configuration

The basic SCRAMNet+ Network communication architecture consists of SCRAMNet+ boards tied together by paired sets of fiber-optic cable in a ring configuration. The maximum recommended distance between each node of the network using this configuration is approximately 300 meters. Maximum node separation using long link fiber is 3,500 meters. The recommended fiber-optic cable is 62.5/125 micron core multi-mode fiber cable with ST connectors.

**NOTE**: On a freshly powered system, a message from any node on the ring may be necessary to establish the carrier.

## 4.6.3 Fiber-optic Cables

The optional paired fiber-optic cables are shipped in a separate carton. The fiber-optic cables are to be attached to the connectors on the SCRAMNet+ board or on the Cabinet Kit, as appropriate. Remove the rubber boots on the fiber-optic transmitters and receivers as well as the ones on the fiber-optic cables. These rubber boots should be replaced when cables are not in use or in the event the node must be returned to the factory.

It is important that the ends of the fiber-optic cable be kept clean. If there is an exceptional amount of light-power loss experienced, the cable ends should be inspected for cleanliness. Alcohol-based fiber-optic cleaning pads are available to remove minor contaminants such as dust and dirt.

Figure 4-8 is a representation of a fiber-optic connector.

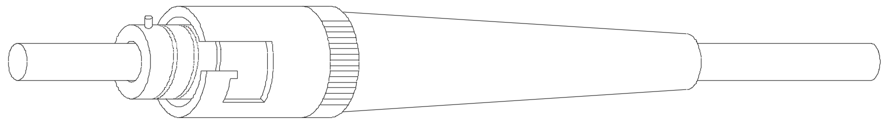


**Figure 4-8 Fiber-optic ST Connector**

**FIBER-OPTIC CABLE PRECAUTIONS**

Fiber-optic cables are made of glass and may break if crushed or bent in a loop with less than a 2-inch radius.

Perform a visual check of the cable ends before inserting into the Media Card connector. If debris is inserted into the transmitter/receiver connector it may not be possible to clean it out or could result in damage to the transmitter or receiver lens. Hair, dirt and dust can interfere with the light signal transmission.

Use an alcohol-base wipe to clean cable ends.

## 4.6.4 Fiber-optic Connection

The fiber-optic cable pairs should be connected between the transmitter pair of the "down-stream" node and the receiver pair of the "up-stream" node. Continuing this type of connection to all nodes in the network will result in a daisy-chain network ring as indicated in Figure 4-9.

**NOTE**: It does not matter if $Tx_1$ or $Tx_2$ is connected to the next node's $Rx_1$ or $Rx_2$ as long as both Tx cables are connected to both of the next node's Rx connectors

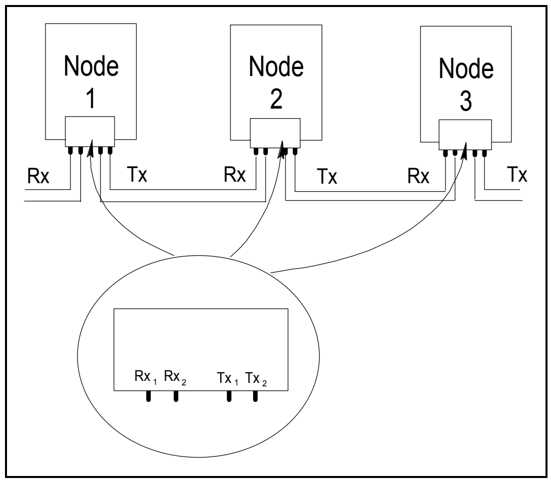



**Figure 4-9 Fiber-Optic Connections**

Make Fiber Optic Bypass Switch connections as shown in Figure 4-10 and Figure 4-11.

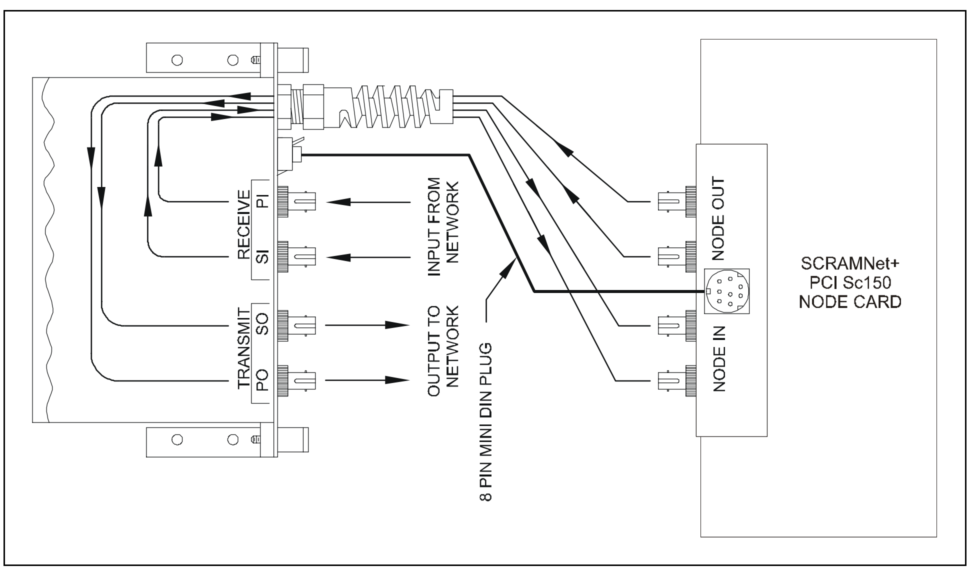



**Figure 4-10 Inserted State (Power On)**

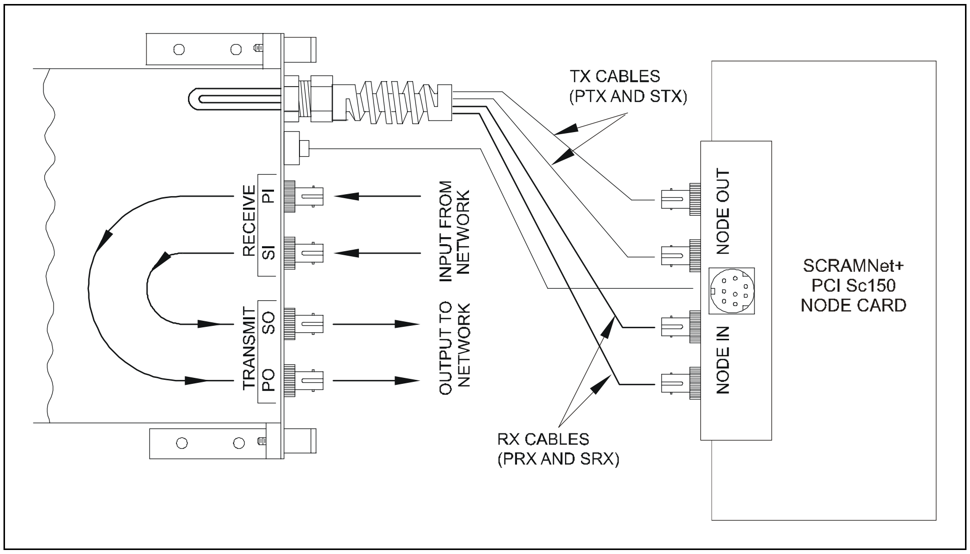



**Figure 4-11 Bypass State (Power Off)**

## 4.7 Auxiliary Connection

The Auxiliary Connection at the Media Card is used for communication with the Fiber Optic Bypass Switch. The 8-pin modular in-line plug male-pin connection described in Figure 4-12 is defined in Table 4-3. (The view is looking into the connector).

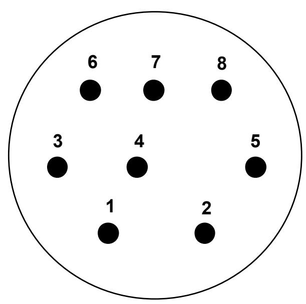



**Figure 4-12 Auxiliary Connection**

**Table 4-3 Auxiliary Connection Pinout**

| Pins | Name | Definition |
|---|---|---|
| 1 | GND | Logic Ground |
| 2 | S_CLK | Serial Clock |
| 3 | F_RELAY | Fiber-optic Relay Drive and Sense |
| 4 | S_DATA | Serial Data |
| 5 | EXT_PWR | +5 Source to External Ground |
| 6 | S_DIR | Serial Data Direction |
| 7 | TRIGGER | Trigger Output (TRIG1) |
| 8 | BAK_PWR | Backup +5 Source from External Device |

# 4.8 Internal Configuration

## 4.8.1 SCRAMNet+ SC150 Control/Status Registers (CSR)

Table 4-4 is a listing of the SCRAMNet+ Control/Status Registers.

If using software other than that supplied by Systran, the offsets in the first two columns must be added to the 0x800000 CSR base address.

The Systran software automatically calculates the offsets.

**Table 4-4 SCRAMNet+ SC150 Control/Status Registers**

| Shortword Offset | Byte Offset | Alias | Size (Bytes) | Active Bits | Type | Register Description |
|---|---|---|---|---|---|---|
| 0x1 | 0x2 | CSR0 | 2 | 16 | R/W | General SCRAMNet+ Enable and Reset |
| 0x80 | 0x100 | CSR1 | 2 | 16 | R/W | SCRAMNet+ Error Indicators |
| 0x103 | 0x206 | CSR2 | 2 | 16 | R/W | General SCRAMNet+ Control |
| 0x182 | 0x304 | CSR3 | 1<br>1 | 8<br>8 | R/W<br>R/W | Number of nodes<br>Node ID |
| 0x205 | 0x40A | CSR4 | 2 | 16 | R | Interrupt FIFO Address (LSW) |
| 0x284 | 0x508 | CSR5 | 2 | 9<br>7 | R*<br>- | Interrupt FIFO Address (MSW)<br>Reserved<br>*Write Transmit Time-out to shadow memory |
| 0x307 | 0x60E | CSR6 | 2 | 0 | - | Reserved |
| 0x386 | 0x70C | CSR7 | 2 | 0 | - | Reserved |
| 0x409 | 0x812 | CSR8 | 2 | 16 | R/W | Extended Control Register |
| 0x488 | 0x910 | CSR9 | 2 | 16 | R/W | Interrupt On Error Mask |
| 0x50B | 0xA16 | CSR10 | 2 | 4<br>12 | R/W<br>- | Shared Memory Address (LSW)<br>Reserved |
| 0x58A | 0xB14 | CSR11 | 2 | 16 | R/W | Shared Memory Address (MSW) |
| 0x60D | 0xC1A | CSR12 | 2 | 16 | R/W | Virtual Paging Register |
| 0x68C | 0xD18 | CSR13 | 2 | 16 | R/W | General-purpose Counter/Timer |
| 0x70F | 0xE1E | CSR14 | 2 | 0 | - | Reserved |
| 0x78E | 0xF1C | CSR15 | 2 | 0 | - | Reserved |
|  |  | ACR | 1 | 5<br>3 | R/W<br>- | Auxiliary Control, RAM<br>Reserved |

### 4.8.2 EEPROM Initialization

The EEPROM is used to store the initial power-up register values. The EEPROM can be programmed either over the host backplane or by most PROM programmers. An EEPROM initialization (EPI) program is included in the Systran Software Utilities Package for most systems.

The board comes with power-up default values in each of the registers and default switch settings so the board can be used without making any changes. Defaults are shown in and where 000 = CSR0, 002 = CSR1, 004 = CSR2. Both location 006 and 008 write to CSR3, TX_ID, RX_ID. This is possible because CSR4 is READ Only. Then location 00A = CSR5, etc.

**Table 4-5 EEPROM Table**

| | 0 | 2 | 4 | 6 | 8 | A | C | E |
|---|---|---|---|---|---|---|---|---|
| **00** | 0000 | 0000 | C040 | XX00 | XX00 | 0010 | 0000 | 0000 |
| **10** | 0800 | 0020 | 0000 | 0080 | 0000 | 0000 | 0000 | 0020 |
| **...** | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |
| **70** | 00B1 | 5300 | 0600 | 5555 | 006E | 5555 | 5555 | 5555 |

**NOTE:** XX denotes TX_ID and RX_ID. Both values are the same.

These values are assigned to the Control/Status Registers on power-up. The values may be changed at any time using the appropriate software to access the CSRs. When the system is powered down and powered up again, the CSRs will be reinitialized to these EEPROM values. Only the first 16 or 17 values are used in the registers.

**Table 4-6 EEPROM Initialization**

| **SCRAMNet+ Registers** | |
|---|---|
| CSR0 - 0 | CSR1 - READ Only (Errors) |
| CSR2 - 0xC040 (BURST Mode) | CSR3 - Node ID (0 - 255) |
| CSR4 - 0 (READ Only) | CSR5 - 0x0010 READ Only (Write a non-zero Network Time-out to shadow register) |
| CSR6 - 0 (Reserved) | CSR7 - 0 (Reserved) |
| CSR8 - 0x0800 (Mech Switch Override)[$] | CSR9 - 0x0020 |
| CSR10 - 0x0000 (WRITE POSTING) | CSR11 - 0x0080 (Reserved) |
| CSR12 - 0 (Virtual Page) | CSR13 - 0 (GP Counter) |
| CSR14 - 0 (Reserved) | CSR15 - 0x0020 (Reserved) |

$ HOLDOFF is enabled

### 4.8.3 Node Identification

Each node on the ring must have a unique Node ID. To set the Node ID, write a unique value 8-bit number between 0 and 255 to CSR3[15:8].

### 4.8.4 Network Time-out

The recommended value for the Network Time-out is:

**# NODES IN RING + (TOTAL LENGTH OF CABLE IN METERS÷50) + 1**

To set the network time-out value, write the result as a 16-bit hexadecimal number to CSR5. This register has a “shadow register” which holds the network time-out value. Each increment in the “shadow register” is worth approximately 240 ns. The time-out will be 240 ns multiplied by the value written.

The time-out register is Write Only. If a read is performed, it will result in a read to the Interrupt FIFO.

**CAUTION**: Ensure a non-zero value is written to CSR5. A value of ‘0’ will keep host-generated data from leaving the Transmit FIFO.

### 4.8.5 Memory Addressing

The SCRAMNet+ memory address must begin on an even boundary beyond extended memory pool, where the boundary is a multiple of the installed board memory. There must be a non-contiguous “gap” between system memory and the SCRAMNet+ SC150 PCI board shared-memory address. For example, if the system has 12 MB of RAM, and the SCRAMNet+ SC150 PCI board has 8 MB of SIMMs, the address for the board would be at the 16 MB boundary.

### 4.8.6 Shared Memory

Set CSR10[0] ON to enable WRITE POSTING to SCRAMNet shared memory.

## 4.9 Byte Swapping

Some computer systems use different methods of byte ordering. Some have the byte order arranged from right to left (Little Endian) and others have the byte order going from left to right (Big Endian). Motorola is an example of a Big-Endian system. Intel is an example of a Little-Endian system.

Table 4-7 is a simplified summary for 8-bit, 16-bit, and 32-bit byte ordering for big endian and little endian.

**Table 4-7 Byte Ordering Comparisons**

| Size | Big Endian | Little Endian |
|---|---|---|
| byte (8-bit) | 12 34 56 78 | 78 56 34 12 |
| shortword (16-bit) | 1234 5678 | 5678 1234 |
| longword (32-bit | 12345678 | 12345678 |

The SCRAMNet-LX and SCRAMNet+ product line has adopted the big-endian ordering philosophy as the default for data passing. SCRAMNet-LX and SCRAMNet+ do not have a built-in byte-ordering conversion function. However, the SCRAMNet+ SC150 PCI permits byte swapping options via the V3 registers PCI_MAP0 and PCI_MAP1.

The PCI allows byte swapping on an aperture basis; PCI_MAP0 controls aperture 0, and PCI_MAP1 controls aperture 1. The options are shown in Table 4-8.

**Table 4-8 PCI_MAP0/PCI _MAP1 Swapping Options**

| Bits 9 | Bits 8 | Swapping Option |
|---|---|---|
| 0 | 0 | No Swapping |
| 0 | 1 | 16-bit Swapping |
| 1 | 0 | 8-bit Swapping |
| 1 | 1 | Reserved |

# 4.10 DMA Operation

The SCRAMNet+ SC150 PCI hardware provides two bi-directional DMA channels for transferring blocks up to 4 KB in length. Refer to Chapter 6 of the *VxxxPBC User's Manual* for details of the DMA operation. See section 3.7 PCI Controller in this manual for information on how to access the *VxxxPBC User's Manual*.

# 4.11 Maintenance

No routine maintenance is required for the SCRAMNet+ general-purpose nodes beyond that which is required for host computer systems. SCRAMNet+ network fiber-optic cabling connectors should be inspected periodically.

# 4.12 Troubleshooting

On a PC platform there is no power-up indication. If there is an incorrect address, the system will not boot up.

On UNIX-like systems, the driver will output a message on boot-up similar to the following:

```
SCRAMNet+ installed and on-line
```

If this message does not appear, the Register base address, memory base address and/or memory size may be incorrect.

All SCRAMNet+ nodes in the fiber-optic network ring must be powered on unless they have Fiber Optic Bypass Switches or Quad Switches installed.

## 4.12.1 LED Indicators

LED indicators on the board can be used to assist with troubleshooting or fine-tuning of the network node. There are no remote indicators, so the chassis must be left open in order to observe the LEDs.

The bi-color LEDs show either green (G) or yellow (Y)

As shown in Figure 4-13, the LEDs are divided into two groups—HOST and NETWORK.

### HOST

LED 1 and LED 2 are viewed together to indicate a memory/CSR read or write as shown in Table 4-9.

**Table 4-9 LED 1 and LED 2 Definitions**

| Action | Memory | | CSR | |
|---|---|---|---|---|
| | LED 1 | LED 2 | LED 1 | LED 2 |
| **READ** | G | G | Y | G |
| **WRITE** | G | Y | Y | Y |

LED 3 indicates a Host Acknowledge signal has been received.

LED 4 (G) indicates a memory interrupt has been received.
LED 4 (Y) indicates a CSR interrupt has been received.

### NETWORK

LED 5 indicates a message is waiting to be transmitted.

LED 6 indicates an error condition set in CSR1 has been detected.

LED 7 indicates a foreign message has been received.

LED 8 indicates a native (OWN) message has been received.

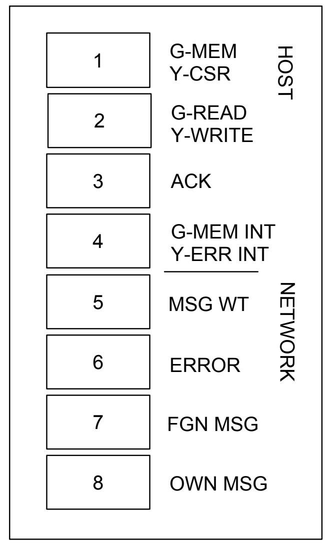



**Figure 4-13 LED Indicators**

## 4.12.2 Hardware

The following hardware conditions will cause the installation to fail:

- The SCRAMNet+ SC150 PCI host card is not installed into a standard PCI slot. (This check should have been performed before installing the SCRAMNet+ SC150 PCI host card into any slots).
- The memory jumper (J2) is set incorrectly.
- The READ/WRITE jumpers J303/J304 are not in the factory-default position.

## 4.12.3 Customer Support

If the system does not boot correctly, reseat the board and double-check cable connections. If problems persist, call Systran Customer Support at (937) 252-5601 for assistance.

Please be prepared to supply the following information:

Host machine: ______________________
OS Name: ______________________
OS Version: ______________________
Bus Interface: ______________________

*This page intentionally left blank*

# 5. OPERATION

## 5.1 Introduction

The SCRAMNet+ Network is a shared-memory system. Every computer on the network has a constantly updated local copy of all global data, which is passed to all the network computers. The network protocol is implemented in the SCRAMNet+ hardware and therefore no software overhead is required to retrieve this information from the network.

The protocol is transparent to the computer. This frees computer processor time for application algorithm execution and other real-time tasks. Since any computer on the network has access to data in the shared memory, any computer can read or modify the shared data and thereby communicate with the other computers on the network.

Very little special software is required for normal operation because of the SCRAMNet+ shared-memory configuration. Typically, SCRAMNet+ memory is installed and linked to a host global common block through the host operating system. Once the link is complete, any program can reference SCRAMNet+ memory as a standard common-block variable reference.

For interrupt driven applications, an interrupt service routine (ISR) is required to handle the interrupts triggered by the SCRAMNet+ node. An example of a generic ISR is included Figure 5-11, page 5-28 at the end of this section.

## 5.2 Shared Memory

Global variables are mapped directly onto the replicated shared memory. The application program typically contains a list of variables or arrays which are stored in a contiguous space and which are to be shared across processors. The analogy of a FORTRAN COMMON BLOCK is most fitting. For the purpose of identification, these variables are referred to as SCRAMNet+ variables.

The application program usually requires a short section of instructions to initialize the SCRAMNet+ hardware and to link the SCRAMNet+ memory to the SCRAMNet+ variable list. The shared memory cannot be used as instruction space.

### 5.2.1 Virtual Paging

CSR12 is the virtual-paging register. Set CSR12[0] to '1' to enable virtual paging.

The SCRAMNet+ network may include a variety of SCRAMNet+ nodes having varying amounts of shared memory. All SCRAMNet+ nodes use the same 8 MB network shared-memory map. Virtual paging allows a node with 4 MB or less shared memory to move its memory window throughout the SCRAMNet+ physical 8 MB network shared-memory map.

If a node has 4 MB of shared memory, it can be paged into the upper 4 MB or the lower 4 MB of the network shared-memory map. If it is paged into the lower 4 MB, it would operate the same as if Virtual Paging were disabled. The network address would be the same as the shared-memory address.

**NOTE**: Virtual paging does not affect host access to shared memory. Virtual Paging only changes the network address. The HOST-SPECIFIC logic always sees the base address of SCRAMNet+ shared memory as zero.

If the 4 MB block were paged into the upper 4 MB, a host-specific write to a shared-memory address of 2 MB would result in a network address of 6 MB. This translation is bi-directional. An incoming network message with a network address of 6 MB will be written to shared memory at 2 MB. Any write accesses to the lower 4 MB will be ignored since there is no memory addressed there.

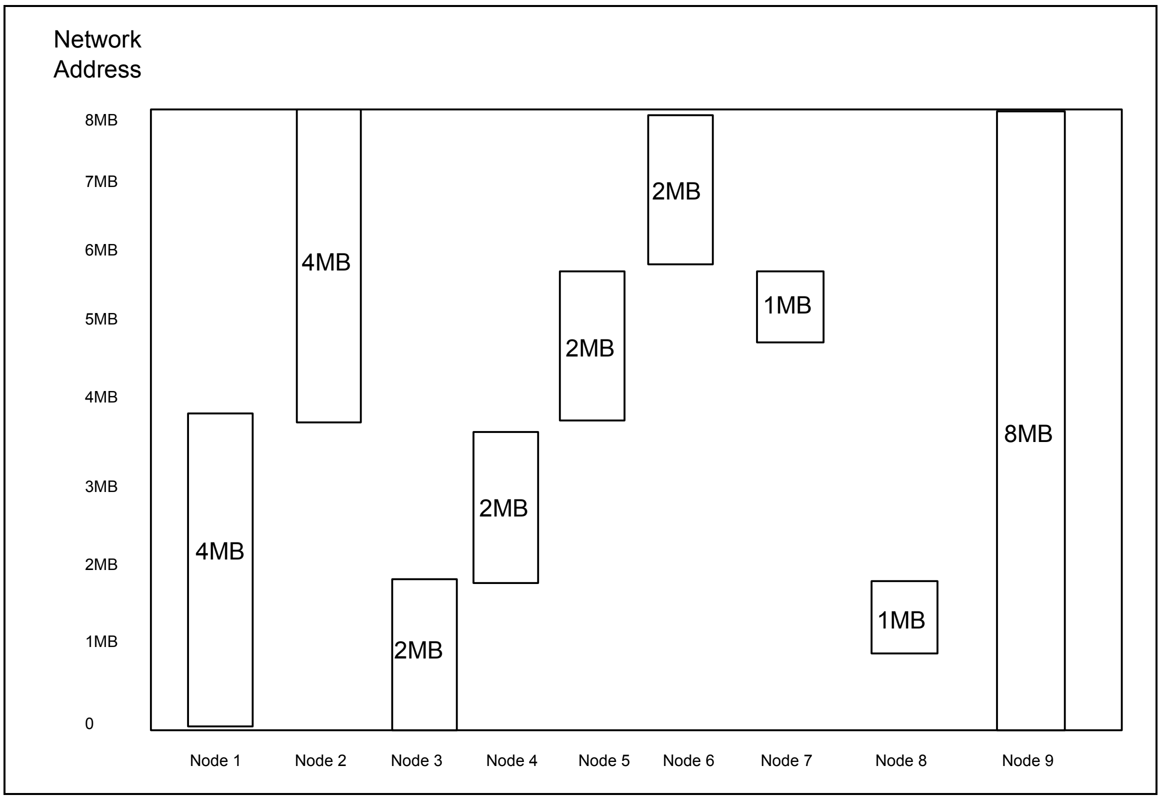



**Figure 5-1 Memory Sharing With Virtual Paging**

To produce a network address, the host write adds the relative SCRAMNet+ address and virtual page offset.

relative address + virtual page offset = network address

For example: 12340 + 400000 = 412340

This network address is then transmitted to all of the SCRAMNet+ nodes and is written to that address. In nodes where the address does not exist in SCRAMNet+ memory, the write is ignored.

The example in Figure 5-1 shows the memory-sharing relationships between various nodes in the virtual-paging mode. Node 1 shares data with nodes 3, 4, 8, and 9. Node 2 shares data with nodes 5, 6, 7, and 9. Node 8 shares data with nodes 1,3, and 9. Node 7 shares data with nodes 2, 5, and 9. Node 9 shares data with all nodes.

### 5.2.2 Memory Considerations

When using SCRAMNet+ shared-memory, consider the following:

#### PROGRAM AND DATA LIMITATIONS

Limitations on application program size and data variable size for a host computer system also apply to applications that use SCRAMNet+ memory because it becomes part of the host system.

#### DATA CACHING

The ability for a computer to write a copy of data to a local fast memory for quicker access later must be turned off during a SCRAMNet+ memory read. Since other nodes may be changing the data, it is critical that the processor read the data directly from SCRAMNet+ memory. This is processor-dependent and does not always apply.

#### MEMORY MAPPING

SCRAMNet+ memory is mapped by all operating systems in constant-length blocks called memory pages.

**NOTE**: To ensure that a compiler or operating system does not try to access unused portions of SCRAMNet+ memory to store other program segments, declare the SCRAMNet+ memory common blocks to be sized to an integer multiple of the processor memory page size. If this is not done, most compilers will try to optimize memory usage by filling out the SCRAMNet+ memory pages with other data. This can cause random results when this local data is transmitted around the network.

### 5.2.3 Control/Status Registers

The SCRAMNet+ boards are controlled through CSRs for node status, setting interrupt vectors, setting interrupt locations, receiving interrupt addresses, mode control and other functions. These registers may be accessed by linking to the I/O page and reading from or writing to the registers as if they were memory. The method used to access the registers depends on the particular computer and operating system being used.

These registers are set only during the SCRAMNet+ Network initialization. Once the control portion of the CSR is set up for the desired mode operation, the node functions as transparent shared memory and references to the CSRs are not required. However, the status portions of the registers will need to be accessed for interrupt servicing and for checking for error conditions. Appendix B, CSR DESCRIPTIONS discusses the definition and use of each bit in the CSRs. Appendix C contains a list of the CSRs and a brief identification of each bit.

## 5.3 Initialization

The initialization of the SCRAMNet+ node from a cold boot is determined by the settings of the EEPROM.

No fiber-optic cable connections are required to perform a read/write to the local host's SCRAMNet+ memory. The control registers CSR0 and CSR2 should both be zero at this point, and SCRAMNet+ memory is available for access. The memory address will remain at '0' and be disabled until programmed with the EEPROM Initialization (EPI) Program.

**NOTE**: All SCRAMNet+ nodes in the fiber-optic network ring must be powered on unless they have fiber-optic bypass switches or Quad Switches installed.

# 5.4 Basic Send/Receive Configuration

The minimum configuration which allows basic send and receive operation is accomplished without interrupts.

- Set CSR0 to '0xF000' to insert the node and initiate the reset of the FIFOs.
- Set CSR0 to '0x8003' to insert the node, toggle the reset of the FIFOs and enable network activity.
- Set CSR2 to '0xC040' to disable the Fiber-optic Loopback mode.
- Read CSR1 to read-out any latched error conditions.
- Read CSR1 again to check for any existing error conditions.
- Check for carrier-detect fail (this means there are fiber-optic cabling problems from the transmitter of the node downstream).
- Write a value to memory from at least one node. This will enable all powered node transmitters and check for fiber-optic ring integrity.
- Read CSR1 to check for any error/status conditions.

# 5.5 Network Ring

Data is passed from one node to the next by fiber-optic or coaxial cable. Given a three-node network configuration with nodes A, B and C, the following connections would be made:

- The transmitter pair from node A is connected by fiber-optic cable to the receiver pair of the next node B.
- The transmitter pair from node B is connected by fiber-optic cable to the receiver pair of node C.
- The transmitter pair of node C is then connected to the receiver pair of node A, thus completing a fiber-optic network ring.

## 5.5.1 Message Contents

The smallest SCRAMNet+ Network message packet consists of 82 bits. This basic message format contains five fields: Source ID, Age, Control, Data Address, and Data Value. The message can be described as follows:

**Table 5-1 SCRAMNet+ Message Contents**

| START | ID | AGE | CONTROL | DATA ADDRESS | DATA VALUE |
|---|---|---|---|---|---|
| 1 | 8+P | 8+P | 1 1 1<br>RES INT RTY | 5+P 8+P 8+P | 8+P 8+P 8+P 8+P |

For every 8 bits of data in the message there is a parity bit attached.

### SOURCE ID

This 8-bit field contains the node ID of the originating node. Value ranges from 0 to 255, so there can be 256 nodes on the network ring.

### AGE

This 8-bit field increments by one as a message passes through each network node. If the age ever exceeds 256 (the maximum number of nodes on the network), the message is removed from the network.

### CONTROL BITS

RES - Reserved.
INT - When this bit is set it signals an Interrupt Message.
RTY - Retry message used only in error correction mode (PLATINUM.)

### DATA ADDRESS

This 21-bit (A[22:2]) field contains the relative SCRAMNet+ memory address. Bits A0 and A1 are always zero for a longword boundary.

### DATA VALUE

This 32-bit field contains the data value in SCRAMNet+ memory that is currently being updated around the ring. When the PLUS mode is enabled, data size may vary up to 256 bytes or 1024 bytes depending on the option selected.

## 5.5.2 Protocol

### BURST MODE

BURST Mode is the normal protocol for SCRAMNet+. The BURST mode is enabled by setting CSR2[12] off, and CSR2[14:15] ON. The BURST Mode protocol allows each node to continuously transmit messages onto the network ring. This mode uses a 4-byte fixed-length message packet for data transfer.

### PLATINUM MODE

The PLATINUM mode is BURST mode with error correction enabled. PLATINUM mode is enabled by setting CSR2[12] OFF; CSR2[14] ON; and CSR2[15] OFF See the Protocol Mode Definition table on page B-7.

### PLUS MODES

The PLUS mode protocol is available as an option to the SCRAMNet+ BURST and PLATINUM mode network protocols. Selecting the 1024 byte maximum data length will result in the maximum bandwidth. However, it will also lengthen ring time, error correction time, and node latency.

Set up the PLUS mode protocol as follows:

- Set CSR2[12] to '1' to enable PLUS mode
- Set CSR2[11] to define the maximum data message size. CSR2[11] only has an effect when CSR2[12] is ON. CSR2[11] has the following definition:
  0 = 256 byte maximum data length
  1 = 1024 byte maximum data length
- Set up the writes to SCRAMNet+ shared memory to be in sequential 32-bit addresses. Data must be written to the SCRAMNet+ node with sequentially incrementing 32-bit addresses to take advantage of the PLUS mode protocol enhancement. It is not necessary to use the full 256 or 1024 byte data length—they are maximum values.

The PLUS mode allows variable-length message packets in which sequentially addressed data in the Transmit FIFO is transferred as a block in a single packet. Both BURST modes are open-loop, non-error-corrected modes of operation.

The node appends 4-byte data values with sequential addresses until the maximum of 256 or 1024 bytes is reached, a non-sequential address is detected, the Transmit FIFO is empty, or a transmit-interrupt event is detected. In both BURST and PLATINUM modes, the node is permitted to have multiple packets on the ring simultaneously.

The transmission of a PLUS mode message is an automatic function, and for the most part, cannot be controlled. If the appropriate PLUS mode bits are set in the control registers, then the following algorithm applies:

1. If Transmit FIFO is empty, end transmission.
2. If the address field is not equal to the address of the previous transmission + 4, end transmission.
3. If length limit overflow for PLUS mode operation occurs, end transmission.
4. ELSE transmit the four data bytes and when done GOTO step 1.

To maintain a PLUS mode transmission, step 1 requires that new data is written to the SCRAMNet+ board at a rate greater than or equal to 16.7 MB/sec (this is a 32-bit write every 240 ns). Any delay in the host data write will result in failure of step 1, and a premature end to the PLUS mode transmission.

While this method results in the reliable generation of a PLUS mode transmission, it increases the node latency. The SCRAMNet+ device automatically increases PLUS mode throughput (when blocking is not used) when needed due to high-throughput host, very busy network, etc.

#### ERROR CORRECTION

Error correction is the automatic retransmission of a SCRAMNet Network message when the original message is received in error by the originating node. The message will be retransmitted indefinitely until it is received correctly. During transmit retry, the same message is being sent. This prevents any new messages from being transmitted by this node. The transmit FIFO will hold these new messages until the retry message is received correctly.

If the original message is received by the originating node with some type of bit error, then this results in the transmit retry bit in CSR1 being set. If the original message is not received by the originating node in the time-out period specified in CSR5, then this results in the transmit retry time-out bit in CSR1 being set. The time-out period is based on the number of nodes in the network ring and the total length of cable used.

### 5.5.3 Performance

#### NODE LATENCY

Node latency is an important factor in networked application in real-time systems design. Data transfer around the network, while fast, does have a measurable delay.

Node latency can be defined as the time delay at a node before a foreign message can be retransmitted. This delay is a minimum of 247 ns; the time to transmit one byte. The maximum node latency depends on the maximum message size and could be from 800 ns to 61.8 µs, depending on the message length selection. To approximate the total maximum delay on the network, multiply the maximum node latency by the number of nodes in the system, and add a propagation delay of 5 ns/meter multiplied by the total message path of the ring in meters.

#### DATA TRANSFER

While the SCRAMNet+ Network appears as a shared-memory system, it is still a data network. The SCRAMNet+ Network includes a series of FIFO buffers to collect data changes until they are transmitted to the other nodes. The Transmit FIFO and the Interrupt FIFO are both 1024 messages in length. These numbers may become significant when performing data transfers of large blocks of data in a short period of time.

### HOLDOFF

If the Transmit FIFO becomes full, subsequent read or write cycles to SCRAMNet+ memory will be extended until the Transmit FIFO is no longer full (see paragraph 5.11.4 for more information).

### SHARED-MEMORY WRITE

SCRAMNet+ shared-memory is based upon a 32-bit word. If an 8- or 16-bit write occurs from the host system, then the 32-bit word that contains that 8- or 16-bit write is sent on the network. Therefore, it is important that other nodes do not simultaneously modify other 8- or 16-bit segments within that 32-bit word.

## 5.5.4 Throughput

A maximum throughput of 6.5 MB/sec could be achieved if only one node were transmitting data, assuming the host CPU could offer the data at that rate. When more than one node is transmitting in BURST mode, then the effective output per node is 6.5 MB/sec divided by the number of transmitting nodes. In BURST and BURST PLUS modes, the node never retransmits its own messages.

In the BURST PLUS mode, a 256-byte packet provides 16.2 MB/s of data throughput. A 1024-byte packet provides 16.7 MB/s maximum data throughput.

When multiple nodes are transmitting in the BURST mode, the network data passing through the other nodes can affect that node's output performance. If a node's receiver is so busy that the Transceiver FIFO is never empty, and the node has already sent a message, then the node will have to wait before it can send another message of its own until either one of its messages comes back or the timer runs out. When the node's own message is received, it is not placed in the Transceiver FIFO thereby creating an opportunity for the node to send a message from the Transmit FIFO.

In PLATINUM and PLATINUM PLUS modes, error detection is enabled. This will affect node latency in that some messages must be retransmitted.

### NETWORK TIME-OUT

Reset the transmit time-out according to the mode of operation selected by writing a 16-bit non-zero value to CSR5 as described in paragraph 4.8.4.

## 5.6 Auxiliary Control RAM

The ACR is a 5-bit register. When ACR Enable CSR0[4] is set, shared memory is not accessible by the host, and the ACR byte is viewed as the least significant byte of every shared-memory four-byte address. The ACR byte value controls the interrupt action(s) taken whenever a write occurs to any byte of the shared-memory 4-byte word. Table 5-2 describes the ACR functions.

**Table 5-2 ACR Functions**

| Bit | Function |
|---|---|
| 0 | Receive Interrupt Enable (RIE) |
| 1 | Transmit Interrupt Enable (TIE) |
| 2 | External Trigger 1 |
| 3 | External Trigger 2 |
| 4 | HIPRO Location Enable |

If these ACR actions are disabled, then no action will be taken when an interrupt condition exists unless override bits CSR0[6] or CSR8[10] are set.

The interrupt action and/or HIPRO mode for a particular shared-memory location is defined by setting these bits. Once the ACR has been defined, set the ACR Enable bit CSR0[4] back to zero so that shared-memory can again be accessed. The ACR actions are still in effect, but the ACR bytes can no longer be accessed while the ACR Enable bit is zero.

In order for the ACR values to take effect for interrupt action, the following SCRAMNet+ CSR actions should be considered for the type of interrupt operation desired:

- Host Interrupt Enable CSR0[3] to receive network interrupts
- Network Interrupt Enable CSR0[8] to transmit network interrupts
- Interrupt on Memory Mask Match Enable CSR0[5] for interrupts from memory writes

Receive and/or Transmit CSR0[1:0] must be enabled in order for the node to receive and/or transmit network data. There are other combinations of CSR settings to achieve varied interrupt results. Appendix B, CSR DESCRIPTIONS describes the SCRAMNet+ CSRs in detail.

In order for the HIPRO mode to become active, ACR[4] must be set for those selected memory addresses where this is to occur. Additionally, CSR2[13] must be set to enable the HIPRO mode. All five of the defined bits of the ACR can be used in any combination to achieve varied results for any shared-memory location.

# 5.7 Interrupt Controls

SCRAMNet+ allows a processor to receive interrupts from and/or transmit interrupts to any other processors on the network, including the originating processor. Table 5-3 indicates the various sources for interrupt control.

## 5.7.1 Interrupt Options

**Table 5-3 Interrupt Controls**

| Condition | Register | Description |
|---|---|---|
| Host Interrupt Enable | CSR0[3] | Must be set in order to receive any interrupts from the network. |
| Receive Interrupt Enable (RIE) | ACR[0] | Generates an interrupt to the host from network data received at the associated Shared Memory location. |
| Transmit Interrupt Enable (TIE) | ACR[1] | Generates an interrupt to the network for a host write to the associated Shared Memory location. |
| Interrupt on Memory Mask Match Enable | CSR0[5] | Permits a shared memory interrupt. Must be set in order to receive any interrupts from the network. |
| Override RIE | CSR0[6] | Generates an interrupt to the host regardless of the ACR RIE setting upon receipt of any network interrupt message. |
| Enable Interrupt on Error | CSR0[7] | Generates an interrupt request as specified in the CSR9 Mask register as the corresponding bit in CSR1 is set. |
| Network Interrupt Enable | CSR0[8] | Permits transmission of interrupt data to the network. |
| Override TIE | CSR0[ 9] | Transmits Interrupt message to the network regardless of the ACR TIE setting. |
| Reset Interrupt FIFO | CSR0[13] | Toggle from ‘0’ to ‘1’ to ‘0’ to reset Interrupt FIFO. |
| Interrupts Armed | CSR1[14] | During the interrupt operation, indicates conditions to receive interrupt are active. If ‘0’, no interrupts will be received by the host. Any write to CSR1 will reset to ‘1’. |
| Enable Interrupt on Own Slot | CSR2[10] | In conjunction with CSR2[9] enables host self-interrupt. |
| LSP of Interrupt Address | CSR4[15:0] | Interrupt Address A15 - A0. |
| MSP of Interrupt Address | CSR5[6:0] | Interrupt Address A22 - A16. Works in conjunction with CSR4[15:0]. |
| Interrupt FIFO Not Empty | CSR5[15] | When ‘0’, Interrupt FIFO is empty. If ‘1’, CSR5 and CSR4 contain legitimate interrupt address(es). |
| Receive Interrupt Override | CSR8[10] | When set, all incoming network messages are treated as interrupt messages. |
| Interrupt on Error Mask | CSR9[15:0] | Interrupts for specified error/status conditions. |
| PCI Interrupt Status | (See NOTE) | This register is used to report interrupt status and to clear the interrupt. |

**NOTE**: PCI configuration register offset 0x48 [32:0]

### SEND/RECEIVE WITH INTERRUPTS

- Set CSR0 to ‘0x0010’ to enable the Auxiliary Control RAM (ACR).
- Clear the SCRAMNet+ ACR by writing zeros to the entire address range.
- Set the SCRAMNet+ ACR memory locations designated to receive and/or transmit interrupts.
- Reset CSR0 to disable the ACR.
- Set CSR0 to ‘0xF000’ to insert the node and initiate the reset of the FIFOs.
- Set CSR0 to ‘0x8003’ to insert the node, toggle the reset of the FIFOs and enable network activity.
- Set CSR2 to ‘0xC040’ to use BURST mode and disable the Fiber-optic Loopback mode.
- Read CSR1 to read-out any latched error conditions.
- Read CSR1 again to check for any existing error conditions.
- Check for carrier-detect fail (this means there are fiber-optic cabling problems from the transmitter of the node downstream).
- Write a non-interrupt value to memory from at least one node. This will enable all powered node transmitters and check for fiber-optic ring integrity.
- Read CSR1 to check for any error conditions.
- Have the interrupt service routine in place.
- Set CSR0 to ‘0x812B’ to enable receive and transmit interrupts.
- Set CSR0 to ‘0x81AB’ to enable interrupt on errors.

# 5.8 Interrupt Conditions

Interrupts are generated under two different conditions:

- A SCRAMNet+ network data write to shared-memory
- A SCRAMNet+ network error detected on the local node

## 5.8.1 Network Data Write

As indicated in Figure 5-2, the Transmit Enable (CSR0[1] must be set before any message can be sent. Only those nodes which have the Transmit Interrupt Enable ACR[1] set for selected addresses send an interrupt bit out with the data packet on the network. Only those nodes which have the Receive Interrupt Enable ACR[0] bit set for that address will generate an interrupt signal to their host processor.

The host issues a write to SCRAMNet+ shared memory. If Override Transmit Interrupt Enable CSR0[9] or ACR Transmit Interrupt Enable ACR[1] is set and Network Interrupt Enable CSR0[8] is set, then the interrupt message is transmitted (INT = 1). Otherwise, the message is transmitted without the interrupt bit set (INT = 0). (See Table 5-1, page 5-4)

Network data write interrupts can be accomplished by two methods:

- **Forced**. Any data writes to any shared memory from the network will generate an interrupt.
- **Masked or Selected**. Data writes to selected shared-memory locations from the network. Under either of these two methods, an interrupt can be generated and received by the same host processor if desired. This condition is termed as “Self-Interrupt”

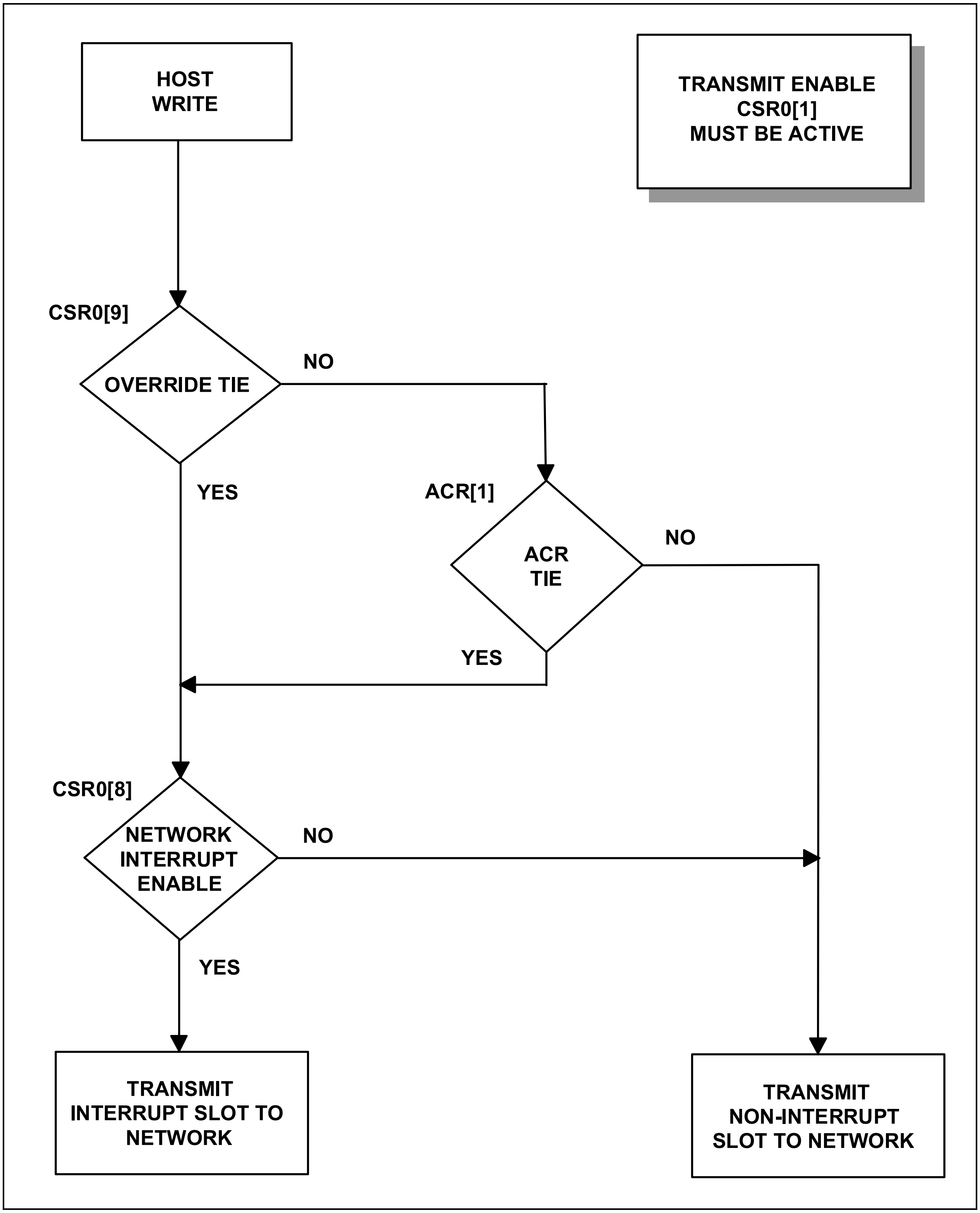



**Figure 5-2 Transmit Interrupt Logic**

## FORCED INTERRUPT

The forced-interrupt method works the same as the selected interrupt method with the exception of choice of interrupt locations. All shared-memory locations are automatically set up to receive and/or transmit interrupts depending upon the override bits set in CSR0 or CSR8.

## MASKED OR SELECTED INTERRUPT

The masked- or selected-interrupt method requires choosing SCRAMNet+ shared-memory locations on each node to receive and/or transmit interrupts. These shared-memory locations may also be used to generate signals to external triggers. The procedure for selecting shared-memory locations for interrupts and/or external triggers is explained in paragraph 5.6: Auxiliary Control RAM.

CSR5 contains the Interrupt FIFO Not Empty CSR5[15]. Set CSR2[9,10] to enable self-interrupts. This allows the message with the interrupt bit set to be processed as an incoming network interrupt. CSR2[9] enables the node's own message to be received as a network message. CSR2[10] allows the interrupt bit to generate an interrupt if it is set.

## SELF-INTERRUPTS

Set CSR2[10:9] to enable self-interrupts. This allows the message with the interrupt bit set to be processed as an incoming network interrupt. CSR2[9] enables the node's own message to be received as a network message. CSR2[10] allows the interrupt bit to generate an interrupt if it is set.

Receive Interrupt logic is described in Figure 5-3. If a native message is received and Write Own Slot CSR2[9] is enabled, and Enable Interrupt on Receipt in Own Slot CSR2[10] is set, the logic then checks for Receive Interrupt Enable. If Override Receive Interrupt Enable CSR0[6] is set or Receive Interrupt Enable ACR[0] is set, and if Interrupt on Memory Mask Match Enable CSR0[5] is set, the address is placed on the Interrupt FIFO.

**NOTE**: Interrupt data is not filtered when the data filter is enabled

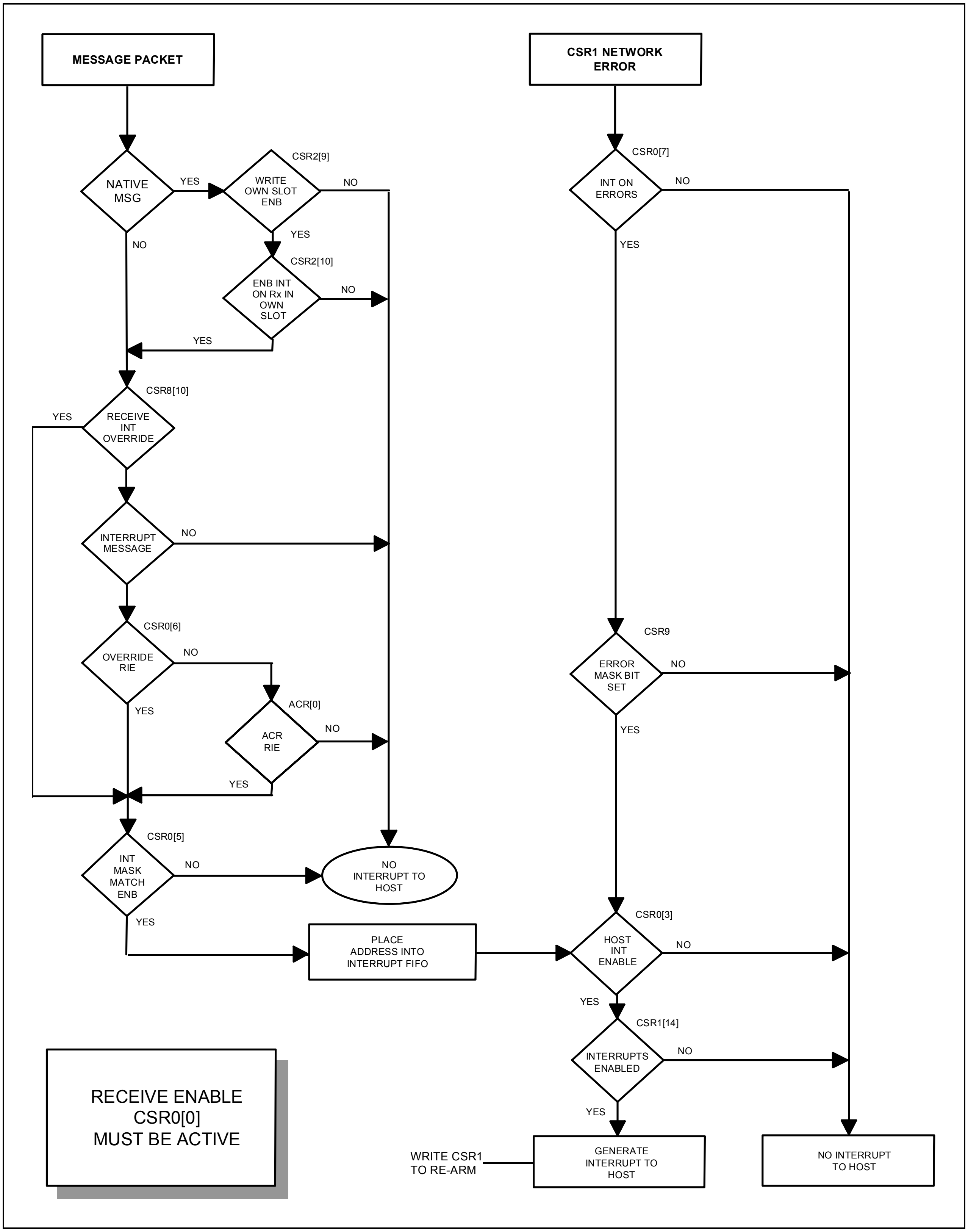



**Figure 5-3 Receive Interrupt Logic**

### 5.8.2 Network Error

The second interrupt condition is designed to intercept network errors. CSR1 contains the following error conditions that may be masked by CSR9:

**Table 5-4 Interrupt Error/Status Conditions**

| Bit | Interrupt |
|---|---|
| 0 | Transmit FIFO Full |
| 1 | Transmit FIFO Not Empty |
| 2 | Transmit FIFO 7/8 Full |
| 3 | (Not masked for errors) |
| 4 | Interrupt FIFO Full |
| 5 | Protocol Violation |
| 6 | Carrier Detect Failure |
| 7 | Bad Message |
| 8 | Receiver Overflow |
| 9 | Transmit Retry |
| 10 | Transmit Retry Time-out |
| 11 | Redundant Rx/Tx Fault |
| 12 | General Purpose Counter/Timer |
| 13 | (Not masked for errors) |
| 14 | (Not masked for errors) |
| 15 | Fiber Optic Bypass Not Connected |

The corresponding bit set (value ‘1’) in CSR1 identifies each of these conditions. If any of the preceding conditions are set and the Interrupt On Memory Mask Match Enable CSR0[5] is set, then an interrupt is generated to the host computer. Additional information about each error condition is contained in Appendix B, Table B-2: CSR1.

If a Network Error is received (Figure 5-3), and if Interrupt on Error CSR0[7] and Host Interrupt Enable CSR0[3] are set, and Interrupts are Enabled CSR1[14], then the message generates an interrupt to the host. If additional network data interrupts occur before the processor is able to service the interrupt, those shared-memory locations are updated and the addresses are added to the Interrupt FIFO queue. However, no additional interrupt signals are sent to the host until interrupts are armed by writing to CSR1. Details are included in the *Programmer’s Reference Guide* for the host computer interface.

### 5.8.3 Interrupt Handling

The Interrupt FIFO is accessed via CSR4 and CSR5. CSR5 contains the most significant seven bits of the 23-bit SCRAMNet+ interrupt address and CSR4 contains the remaining 16 bits of the interrupt address. (The 23-bit address allows for future expansion of memory.) CSR5 also contains Interrupt FIFO Not Empty (bit 15).

**NOTE**: The SCRAMNet+ Network is a longword (32-bit) oriented shared memory. External Triggers and Interrupts will occur when any of the four bytes associated with a longword are accessed. The Interrupt FIFO contains the longword address. If each of the four bytes of an interrupt location are written into as byte accesses, then four interrupts to the same longword address will be generated. Likewise, if each word of an interrupt location is written into as 16-bit shortwords, then two interrupts to the same longword address will be generated.

The two values of CSR5 and CSR4 make up the interrupt address. When an interrupt is received, the ISR should read CSR5 first in order to check the Interrupt FIFO Not Empty bit. If this bit is set (value is '1'), then read CSR4. If this bit is CLEAR (value is '0') then the Interrupt FIFO is empty. Therefore, the interrupt was due to an error, assuming that Enable Interrupt On Error is set.

Every read from CSR5 and CSR4 will contain the SCRAMNet+ memory address of the data received from the network interrupt. Every read of CSR5 and CSR4 will automatically increment the FIFO pointer to the next interrupt address for both registers. CSR4 should be read only if Interrupt FIFO Not Empty CSR5[15] is set. Continue to read CSR5 and CSR4 until the Interrupt FIFO Not Empty bit is zero. Writing any value to CSR1 will re-enable interrupts.

**NOTE**: See Page 5-28 for an example of a standard ISR algorithm for handling interrupts from the SCRAMNet+ boards.

**WARNING**: If HIPRO is enabled, an interrupt may affect the sequence of addresses on a read/write if SCRAMNet+ is manipulated in the ISR.

If an interrupt occurs before the interrupts have been armed, the interrupt will be placed in the Interrupt FIFO and it will occur when the interrupts are armed (CSR 1)

## 5.9 External Triggers

Two external triggers are provided by the SCRAMNet+ Network. The external triggers will occur only if the ACR has been configured to enable them. Triggers 1 and 2 are generated by SCRAMNet+ shared-memory access. Triggers generate a 26.64 ns TTL level compatible, non-terminated, output.

- Trigger 1 - Host read/write (ACR[2] enables)
- Trigger 2 - Network write (ACR[3] enables)

Trigger 1 will be generated for any host access to SCRAMNet+ memory.
Trigger 2 will be generated by a network write to the SCRAMNet+ memory.

The trigger output signals are available through the external trigger connection pins. Also, pin 7 of the Auxiliary Connector is connected to TRIG1.

The triggers can be used to measure time intervals or to start or stop an external event.

### EXAMPLE 1 - MEASURE TRANSACTION TIME

Select a shared-memory address in node A and enable trigger 1 by setting ACR[2]. Select the same memory address in node B and enable trigger 2 by setting ACR[3]. Connect a wire from TRIG1 to an oscilloscope. Connect a wire from TRIG2 to the oscilloscope. Initiate a host write to the node A memory address. When the corresponding network write occurs, the time difference can be measured.

### EXAMPLE 2 - MEASURE RING TIME

Enable WRITE-ME-LAST mode on node A. Select a shared-memory address and enable trigger 1 and trigger 2 for that address. Connect a wire from TRIG (pin 2 - trigger 1 OR 2) to an oscilloscope. Initiate a host write to that memory address. The time difference displayed on the oscilloscope will approximate the ring time—the time it takes from the host write on node A to the network write on node A.

### EXAMPLE 3 - SET OFF AN ALARM

Enable trigger 1 for the shared-memory address of a critical datum by setting ACR[2]. Connect a wire from TRIG1 to an alarm light. When the host write to that memory address occurs, the light will come on.

# 5.10 General Purpose Counter/Timer

This 16-bit counter/timer can be programmed by changing CSR9[13] and CSR9[14] to select the desired mode as described in Table 5-5. CSR8[9] can be set to override the counter/timer mode settings and allow the counter/timer to run free at 26.66 ns (37.5 MHz). CSR9[12] can be set to generate an interrupt upon overflow of the counter/timer. The output from the event counter/timer is stored in CSR13. See Appendix B, page B-4, B-10, and B-12 for more information.

## 5.10.1 Available Modes

The General Purpose Counter/Timer register (CSR13) can be used as a counter or a timer. The mode is selected via a combination of registers and bits, which are explained on page B-9. Table 5-5 describes the counter/timer modes available:

**Table 5-5 General Purpose Counter/Timer Modes**

| Mode | Description |
|---|---|
| Count Errors: | Each error detected in CSR1 will increment the counter by 1. |
| Count Trigger 1 and 2: | Each time a trigger event occurs the counter will increment. |
| Transit Time: | Set this mode and clear the counter. The counter will begin counting when the next message is transmitted, and stop counting when any message generated by this node is received. |
| Network Events: | Count incoming network messages. |
| Free Run @ 26.66 ns: | Increment counter using internal 37.5 MHz clock. Counter will roll over every 1.78 ms. |
| Free Run @ 1.706 µs with Trigger 2 to CLEAR: | Increment counter using the 585.9 KHz clock. Counter will roll over every 111.8 ms. Assertion of Trigger 2 will clear the counter. |

### 5.10.2 Rollover/Reset

A rollover/reset can generate an interrupt by setting Interrupt On General Purpose Counter/Timer Overflow Mask CSR9[12]. When this bit is set, an interrupt is generated to the host system whenever the counter register (CSR 13) rolls over or overflows. Interrupt On Errors CSR0[7] mode must be enabled in order for this to work properly. The counter/timer will roll over when it reaches 65,535 + 1.

Only one mode may be selected at a time since they use the same counter/timer register (CSR13) for output.

### 5.10.3 Presetting Values

The counter/timer register counts upward and may be preset with a value to arrive at the desired interrupt interval.

#### EXAMPLE

To set an interrupt to occur every 100 ms, the counter register is preloaded with '8717', so that when the counter reaches 65,536, only 100 ms would have passed instead of 111.8 ms.

The value of '8717' was determined by dividing the desired interrupt time of 100 ms (100,000 µs) by the increment frequency of 1.760 µs, which results in 56,818. This is the number that would be in the counter register after 100 ms. To obtain a starting value of '8717', subtract 56,818 from 65535. The counter/timer will not roll over until it reaches 65,535 + 1.

See Appendix B, pages B-4, B-10, and B-12 for additional information.

## 5.11 Modes of Operation

### 5.11.1 Data Filter

Many implementations of shared memory tend to rewrite data values to memory that have not actually changed. In order to reduce network traffic, the SCRAMNet+ board has the ability to compare the new value with the old value of data and avoid sending unchanged data values out on the network. This feature is a type of data filtering and can be enabled without affecting node latency while improving network throughput. See Figure 5-4 Data Filter Logic .

CSR0[10] and CSR0[11] control the operation of data filtering as shown in Table 5-6 (see Appendix B for details of CSR operation):

- CSR0[10] enables the data filtering during transmission to the SCRAMNet+ memory and only for the address space above the first 4 K bytes.
- CSR0[11] enables the address space of the first 4 K bytes to be data-filtered in conjunction with CSR0[10].

**Table 5-6 Data Filter Options**

| CSR0[11 | CSR0[10]* | Result |
|---|---|---|
| 0 | 1 | Only the address space above 4 K bytes of shared memory is data-filtered |
| 1 | 1 | All shared memory is data-filtered |

* CSR0[10] must be ON for any data filtering to take place on that node.

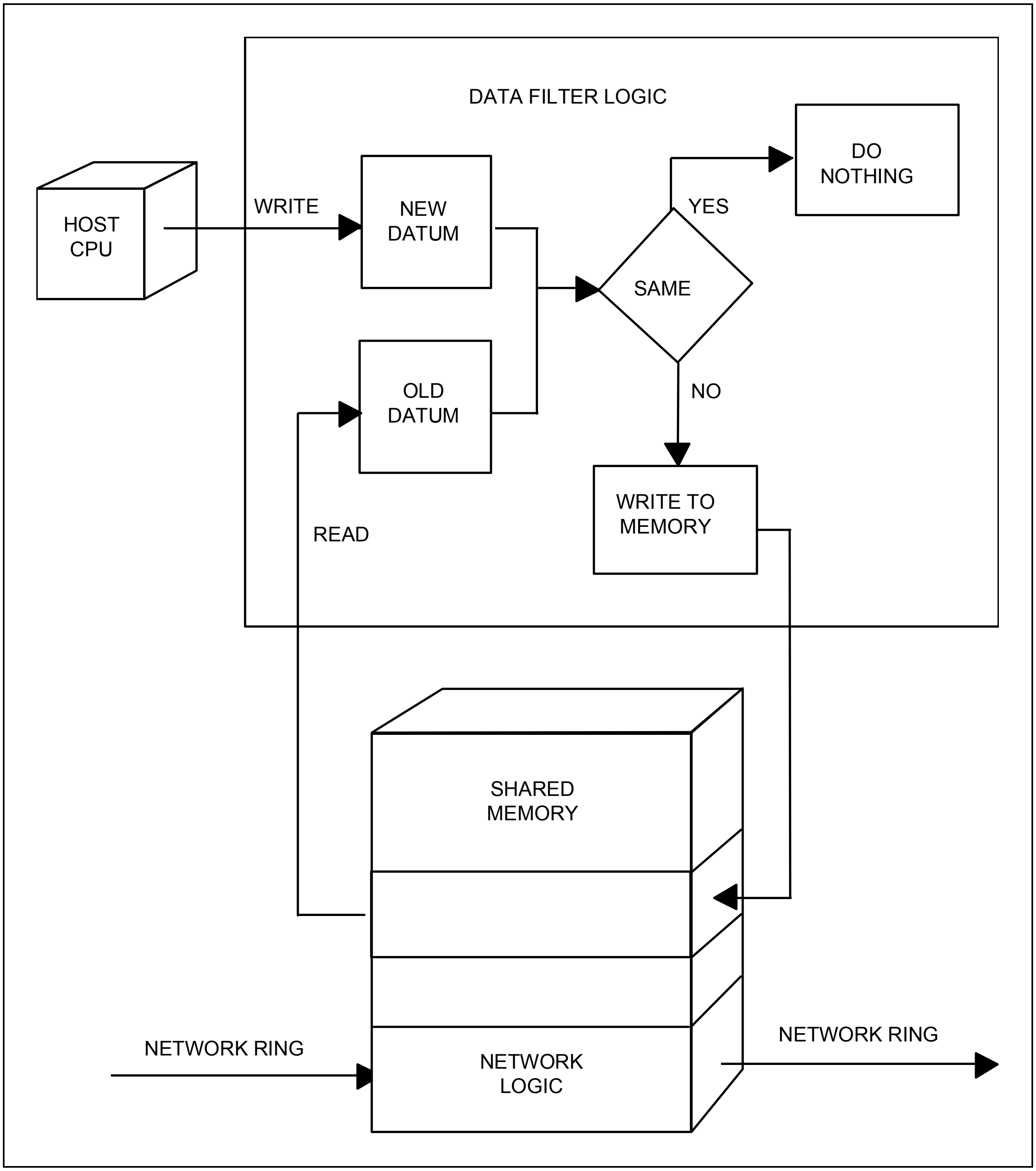



**Figure 5-4 Data Filter Logic**

## 5.11.2 HIPRO Mode

### WRITE

The SCRAMNet+ network message is based on 32-bit longword data. If a host processor is only capable of 8- or 16-bit data transactions, then the SCRAMNet+ bandwidth is quartered or halved, respectively. For each 32-bit data transaction from the host, two 16-bit data transactions, or four 8-bit transactions will occur on the bus each requiring a SCRAMNet+ network write.

The **HIPRO** mode was created to provide an efficient means to transmit two 16-bit data transactions as one 32-bit network write. The first of the two 16-bit writes is written to memory but is prevented from going onto the network. The second 16-bit write to memory triggers the write of the 32-bit location to the network. HIPRO WRITE will not work if Disable Host to Memory Write CSR2[8] is set, because the first 16-bit write must be to the SCRAMNet+ memory.

**NOTE**: The order of writing the shortwords or bytes into the longword boundary does not matter. However, it is important that a HIPRO location does receive a second shortword write if a first shortword write is initiated, or a total of 4-byte writes if a byte write is initiated, to a HIPRO location. Otherwise, it is possible to partially write a 32-bit location causing the data to be lost and never be transmitted.

The **HIPRO** mode is also effective for transmitting user-defined 16-bit data items. Two 16-bit data items may be sent as one 32-bit data item if they are consecutive and lie within the same 32-bit address boundary, and interrupts are not being used.

**HIPRO** mode is selected for those memory addresses which have ACR[4] set. **HIPRO** Enable CSR2[13] must also be set. Use a non-HIPRO location write to synchronize the HIPRO flags.

### 5.11.3 Loopback Modes

Each node has a Monitor and Bypass mode, Wire Loopback mode, Mechanical Switch Loopback mode, and a Fiber-optic Loopback mode. These modes are used to check the node's performance and to test transmit/receive circuitry. The loopback mode routes data which would normally be transmitted on to the network, directly back to the node from different points.

Table 5-7 depicts the data path for the Monitor and Bypass mode.

Table 5-8 depicts the data path for Wire Loopback Mode.

Table 5-9 depicts the data path for Mechanical Switch Loopback Mode.

Table 5-10 depicts the data path for the Fiber-optic Loopback Mode.

Table 5-11 depicts the data path for the Insert Mode.

## MONITOR AND BYPASS MODE

This mode permits the node to receive data only. Network data is not re-transmitted.

**Table 5-7 Monitor and Bypass Mode States**

| State | Register | Setting |
|---|---|---|
| Receive Enable | CSR0[0] | ON |
| Transmit Enable | CSR0[1] | DON'T CARE |
| Insert Enable | CSR0[15] | OFF |
| Enable Wire Loopback | CSR2[7] | OFF |

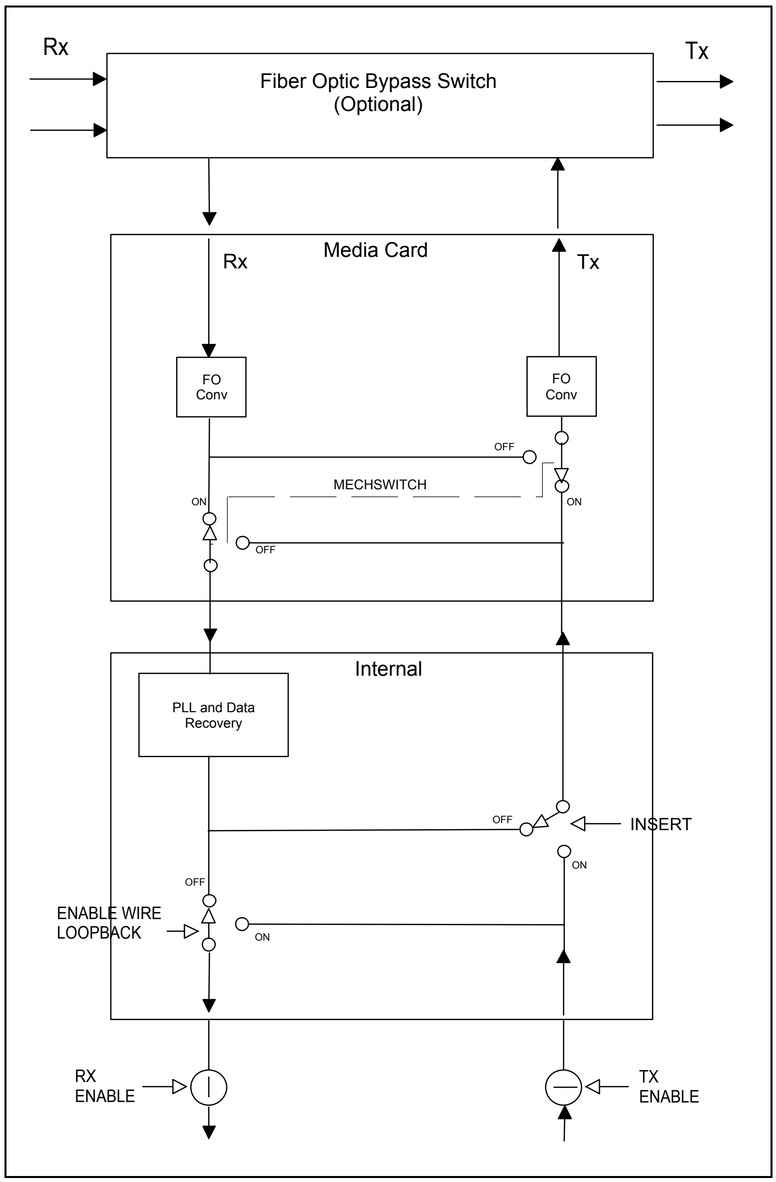



**Figure 5-5 Monitor and Bypass Mode**

## WIRE LOOPBACK MODE

The Wire loopback permits testing of the internal circuitry and needs no manual external modifications to work. In this mode, the transmitted signal does not leave the board.

**Table 5-8 Wire Loopback Mode States**

| State | Register | Setting |
|---|---|---|
| Receive Enable | CSR0[0] | ON |
| Transmit Enable | CSR0[1] | ON |
| Insert Enable | CSR0[15] | OFF |
| Enable Wire Loopback | CSR2[7] | ON |

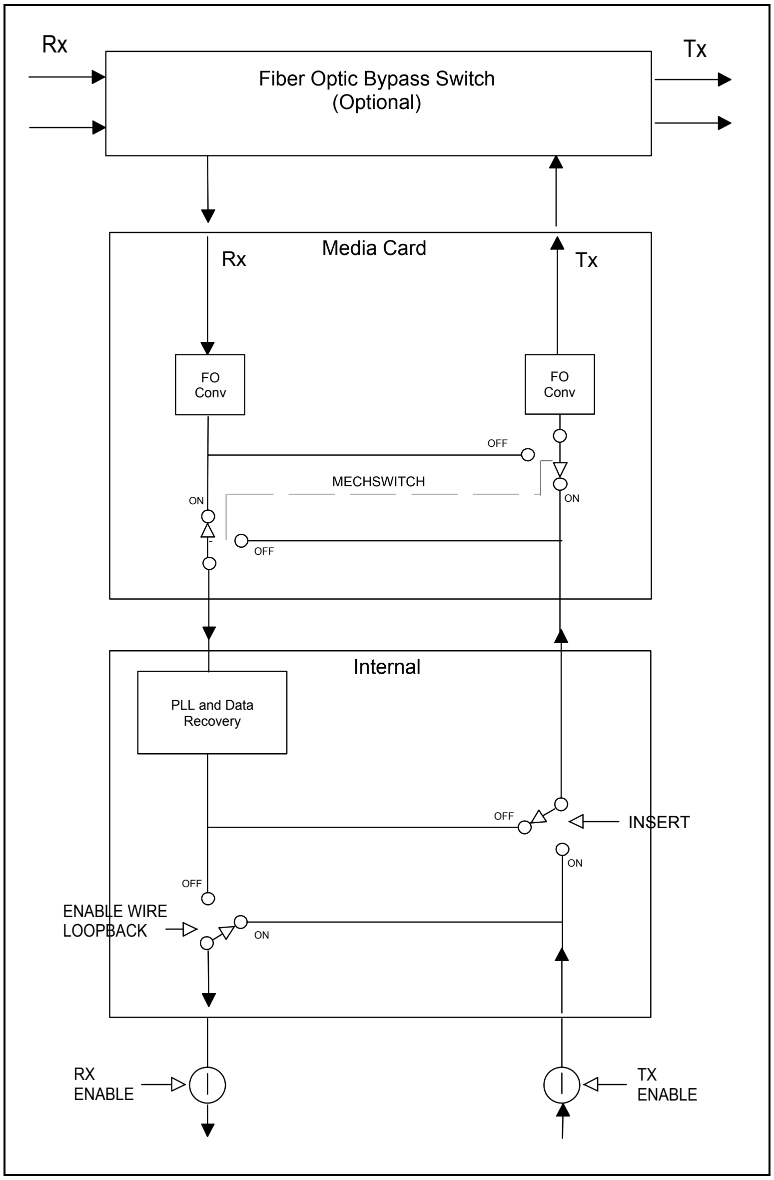



**Figure 5-6 Wire Loopback Mode**

## MECHANICAL SWITCH LOOPBACK MODE

This mode permits testing of all circuitry up to and including a major portion of the Media Card.

**Table 5-9 Mechanical Switch Loopback Mode States**

| State | Register | Setting |
|---|---|---|
| Receive Enable | CSR0[0] | ON |
| Transmit Enable | CSR0[1] | ON |
| Insert Enable | CSR0[15] | ON |
| Enable Wire Loopback | CSR2[7] | OFF |
| Mechanical Switch Override | CSR8[11] | OFF |

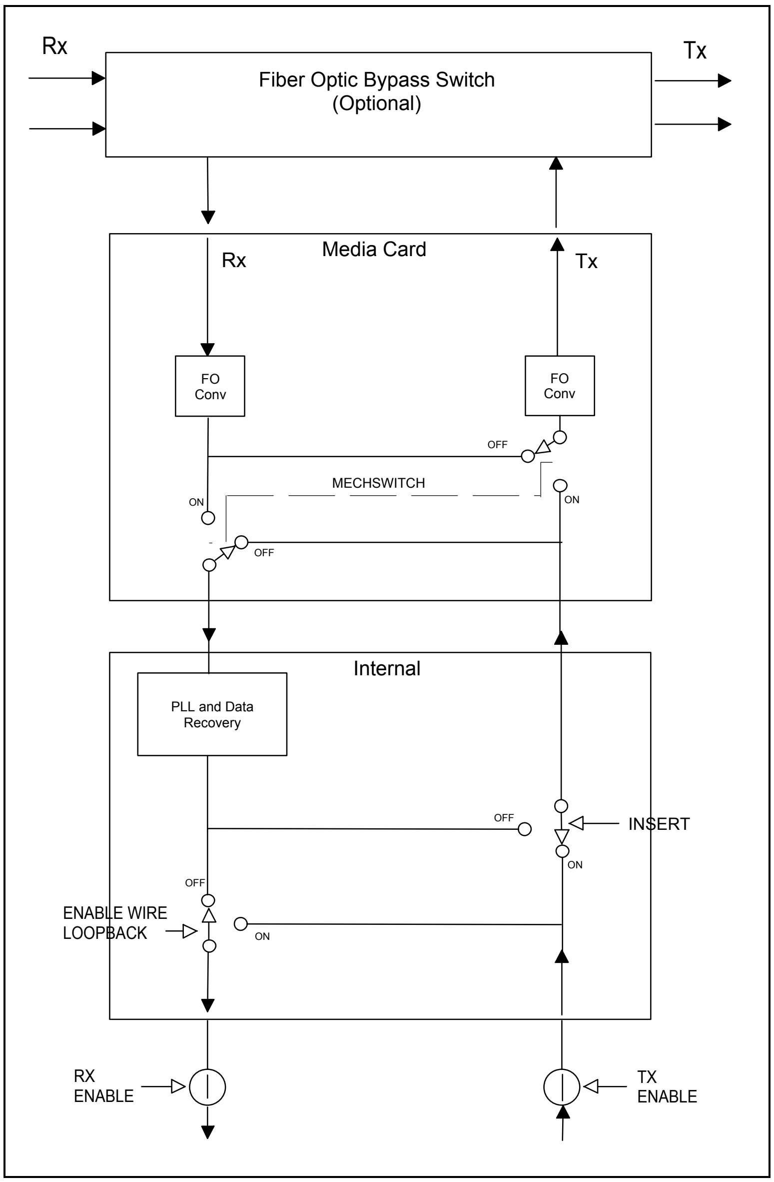



**Figure 5-7 Mechanical Switch Loopback Mode**

## FIBER-OPTIC LOOPBACK

When this mode is invoked, the output of the transmitter is connected by fiber optics directly to the input of the receiver, and the receiver is disconnected from the network.

**Table 5-10 Fiber-optic Loopback Mode States**

| State | Register | Setting |
|---|---|---|
| Receive Enable | CSR0[0] | ON |
| Transmit Enable | CSR0[1] | ON |
| Insert Enable | CSR0[15] | ON |
| Enable Wire Loopback | CSR2[7] | OFF |
| Disable Fiber-optic Loopback | CSR2[6] | OFF |
| Mechanical Switch Override | CSR8[11] | ON |

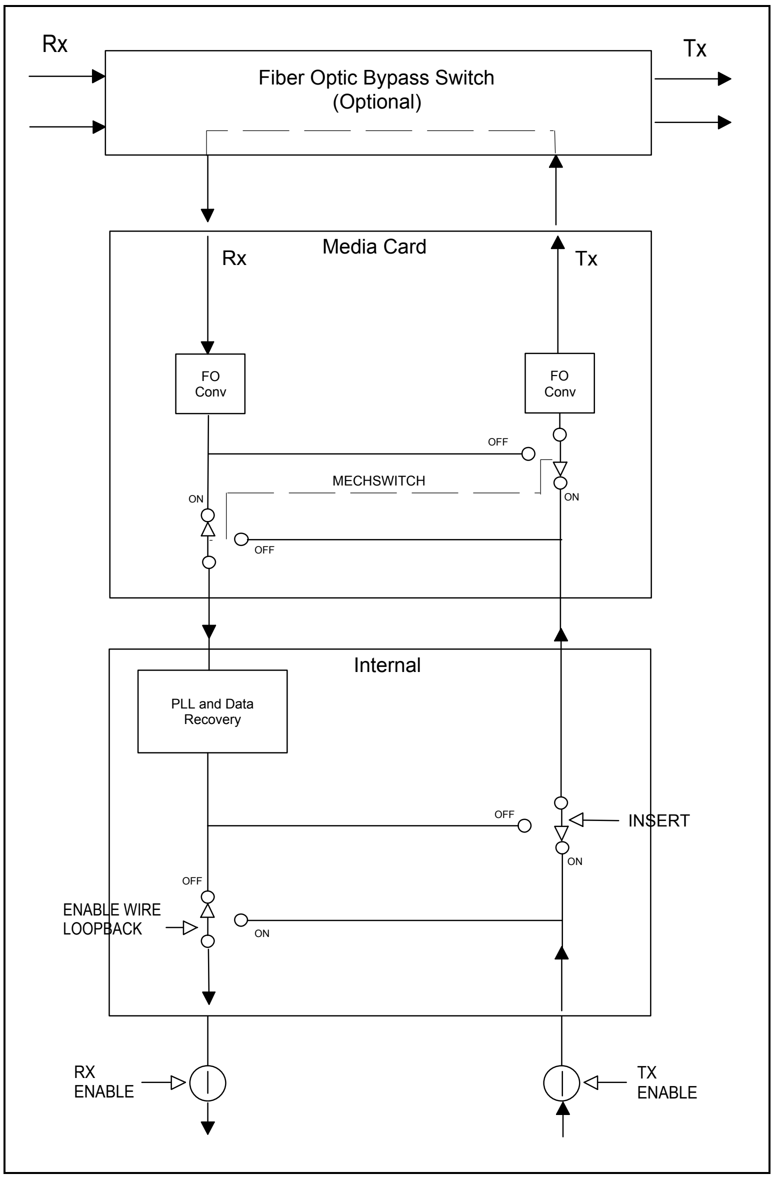



**Figure 5-8 Fiber-optic Loopback Mode**

The optional Fiber Optic Bypass Switch must be installed for this to work. However, in the absence of the Fiber Optic Bypass Switch, fiber-optic cables could be run from the node's transmitter output connectors to the receiver input connectors. This configuration, with Insert Node enabled, would constitute a Fiber-optic Loopback mode for stand-alone testing. Disable Fiber-optic Loopback CSR2[6] must be set ON when the node is in use as a part of the network. However, this configuration could not be used in a network ring in the place of an Fiber Optic Bypass Switch because it would cause a break in ring continuity.

The Fiber Optic Bypass Switch is used to provide fiber-optic ring continuity when a node is powered down or in loopback mode. CSR2[6] controls the operation of the Fiber Optic Bypass Switch by enabling or disabling Loopback mode. To disable the Fiber-optic Loopback mode, set CSR2[6] ON. This state allows data to be transmitted and received on the network ring for this node.

When the Fiber-optic Loopback mode is enabled (CSR2[6] OFF), the Fiber Optic Bypass Switch does not allow network data to be received by the node. Likewise, no data can be transmitted by the node into the network ring.

When power is lost to the Fiber Optic Bypass Switch, Fiber-optic Loopback mode is enabled regardless of its prior state in order to maintain ring integrity. This is also the default power-up state.

## NODE INSERT MODE

In this mode the node becomes part of the network (Figure 5-9).

**Table 5-11 Node Insert Mode**

| State | Register | Setting |
|---|---|---|
| Receive Enable | CSR0[0] | ON |
| Transmit Enable | CSR0[1] | ON |
| Insert Node | CSR0[15] | ON |
| Enable Wire Loopback | CSR2[7] | OFF |
| Disable Fiber-optic Loopback | CSR2[6] | ON |
| Mechanical Switch Override | CSR8[11] | ON |

**NOTE**: Do not enable the Wire Loopback and Fiber-optic Loopback and/or Mechanical Switch loopback modes simultaneously. A node in Wire Loopback mode **and** Insert Node will create a break in the network ring disabling all nodes.

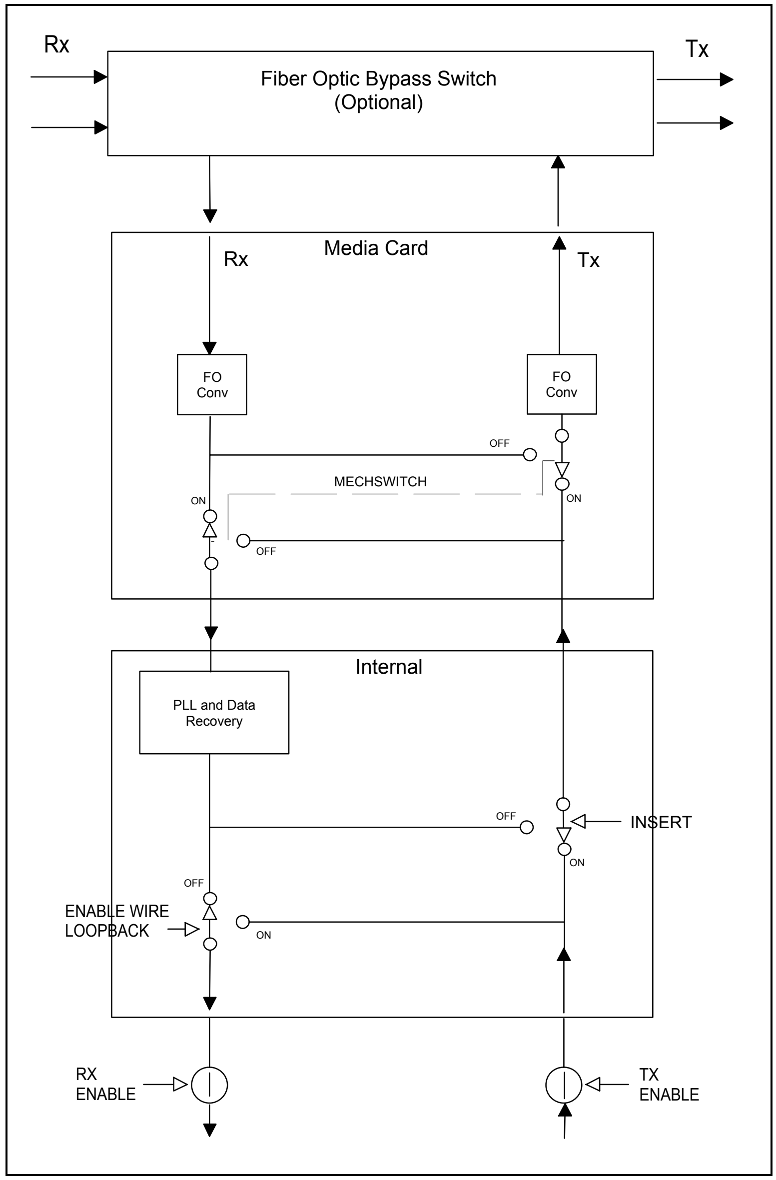



**Figure 5-9 Insert Mode**

## 5.11.4 Holdoff Mode

To enable Holdoff, set CSR8[1] OFF. The Holdoff feature automatically slows down CPU data writes to the SCRAMNet+ memory when the Transmit FIFO becomes full. The Transmit FIFO serves as a buffer between the SCRAMNet+ memory and the SCRAMNet+ network.

The Transmit FIFO can become full when the host CPU is writing to SCRAMNet+ memory faster than the network can absorb the data. If a CPU is capable of writing to the

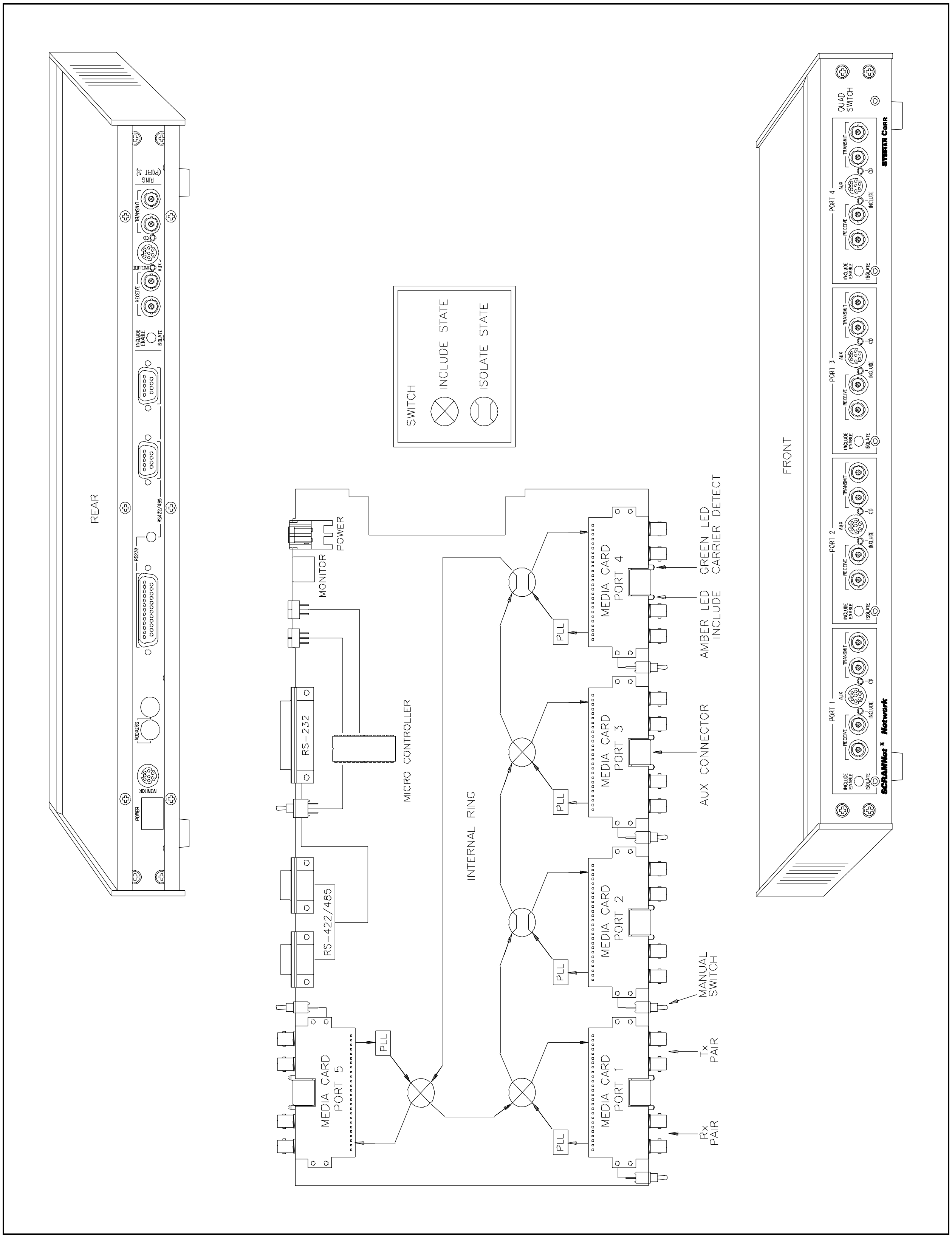



**Figure 5-10 Quad Switch**

SCRAMNet+ memory on the PCI bus at such a rate that the Transmit FIFO becomes full (1024 deep), data could be lost. In the event that the Transmit FIFO becomes full, the hardware will automatically extend the next write cycle until the Transmit FIFO empties at least one message. This prevents the loss of any data and is transparent to the user.

### 5.11.5 Write-Me-Last Mode

The Write-Me-Last mode of operation allows the originating node to be the last node in the ring to have the data deposited to its memory. When the host performs a write to the SCRAMNet+ shared-memory, it is not immediately written to the host memory, but is first sent to the other SCRAMNet+ nodes on the network.

Set CSR2[8] and CSR2[9] to enable the Write-Me-Last mode. If desired, this mode can also be used to generate interrupts to the originating node by setting CSR2[10] as well. CSR2[8] is the Disable Host to Shared Memory Write.

## 5.12 Quad Switch

The Quad Switch is a switching center and is used to dynamically configure active SCRAMNet and SCRAMNet+ ring(s).

The Quad Switch provides dynamic configuration of up to five separate rings. Each separate ring is connected to a port on the Quad Switch. Refer to Figure 5-10. Each ring can be isolated from the other rings or can be included with one or more of the other attached rings.

There is a single logical ring internal to the Quad Switch. The Quad Switch has five external ports, which allow access to this logical ring. Ports 1 through 4 are accessible on the front of the Quad Switch cabinet. Port 5 access is at the rear of the cabinet.

All five ports have standard SCRAMNet transmitters and receivers. Each port can transmit data to and receive data from the internal ring.

The Quad Switch is designed so that a port will be switched into the ring if all its switching controls are enabled. Any one of the switching controls can cause the port to be switched out.

CSR0 IMME and HIE must be set in order to set up interrupts.
CSR1 Bits 0-15 contain various error and status conditions. Interrupts are re-armed whenever any value is written to CSR1.

CSR4 Bits 0-15 contain the interrupt address bits A0-A15.

Read PCI Interrupt Status Register (PCI Configuration Register—offset 0x48) and save value.
Write Interrupt Status Register value back to PCI Interrupt Status Register to clear the interrupt.

CSR5 Bits 0-6 contain the interrupt address bits A22-A16.
Bit 15 contains the Interrupt FIFO Not Empty status.

If an interrupt has been received by the host processor from the SCRAMNet+ Network interrupt logic, then the Interrupt Service Routine will be invoked. Interrupts will be disabled until re-armed by writing to CSR1. Until that time, all other interrupts will be written into the Interrupt FIFO where they can be processed in the Interrupt Service Routine.

If Interrupts on Errors is enabled, then an interrupt due to an error has occurred if the interrupt FIFO is empty on the initial check of CSR5 in the Interrupt Service Routine.

```
Read CSR5
    Test the Interrupt FIFO Not Empty status bit 15
    If (Interrupt FIFO is Empty)
        Read CSR1 to determine the error condition(s)
        Respond to any error conditions
    End if

    While (Interrupt FIFO is NOT Empty)
        Save interrupt address bits A22-A16 from CSR5 (from previous read)
        Read CSR4 and save interrupt address bits A15-A0
                        • • •
        Service interrupt according to interrupt address data or address
                        • • •
        Read CSR5 and save Interrupt FIFO Empty status
    Endwhile

    Write to CSR1 to re-arm interrupts
    Return from interrupt service routine
```

**Figure 5-11 Interrupt Service Routine**

# APPENDIX A

# SPECIFICATIONS

## TABLE OF CONTENTS



## FIGURES

# A.1 Hardware Specifications

| | |
|---|---|
| Hardware Compatibility: | PCI specification version 2.1<br>Target initiator DMA device<br>Operates at maximum 33 MHz clock speed |
| Physical Dimensions: | |
| PCI Card | 9.0" x 4.2", one slot |
| Weight: | |
| PCI Card | 0.34 lbs (W/O SIMMs and Media Card, W/face plate) |
| Media Card, Fiber Optic | 0.0915 lbs |
| SIMM (1) | 0.0285 lbs |
| Electrical Requirements: | +4.75 to +5.25 VDC, 1.5 Amps max. |
| Temperature Range | |
| Storage: | -40° to +70°C |
| Operation: | 0° to +40°C |
| Humidity Range: | |
| Storage | 0% to 95% (noncondensing) |
| Operating | 10% to 90% (noncondensing) |
| Network Line Transmission Rate: | 150 million bits/second |
| Message Length: | |
| Fixed Length: | 82 Bits |
| Variable Length: | 256 or 1024 Data Bytes Maximum |
| Maximum Nodes on Network Ring: | 256 |
| Error Correction: | Available in PLATINUM mode only |
| Maximum Node Separation: | |
| Coax: | 30 meters |
| Standard Fiber: | 300 meters |
| Long Link Fiber: | 3500 meters |
| Shared Memory: | |
| ASIC Memory | 4 KB |
| On-board Memory | 128 KB |
| Optional Sizes: | |
| Low Density SIMMs (512 KB) | 512 KB, 1 MB, 2MB |
| High Density SIMMs (2 MB) | 2 MB, 4 MB and 8 MB |
| Effective Per-Node Bandwidth: | |
| 4 bytes/packet: | 6.5 MB/sec |
| 256 bytes/packet: | 16.2 MB/sec |
| 1024 bytes/packet: | 16.7 MB/sec |
| Node Latency: | |
| 4 bytes/packet: | 250 ns -800 ns |
| 256 bytes/packet: | 250 ns - 16 µs |
| 1024 bytes/packet: | 250 ns - 61.8 µs |
| Mean Time Between Failures (MTBF) | 133,110 hours |
| Internal clock speeds: | |
| SCRAMNet board crystal | 150 MHz, +/-100 ppm[1] |
| 26.66 ns timer is a divide-by-four: | 37.5 MHz[2] |
| 1.706 µs timer is a divide-by-256: | 585.9 KHz[3] |

1. Specifications on the crystal demonstrate the precision and stability of the main clock from which all other clocks are derived. This does not include the vagaries introduced by the circuit.
2. The 37.5 MHz clock is the distributed (on board) clock used by other circuits on SCRAMNet host cards.
3. The 585.9 KHz clock is that which is counted by the internal (to ASIC) users timer.

# A.2 Performance

**Table A-1 Data Performance Table***

| DMA PERFORMANCE in MBytes/s with the data flow direction TowardNetwork/FromNetwork | V3 from SCRAMNet Prefetching on | V3 from SCRAMNet Prefetching off |
|---|---|---|
| SCRAMNet Write Posting on | 16.6/15.0 | 16.6/15.0 |
| SCRAMNet Write Posting off | 10.9/15.0 | 10.9/15.0 |
| | | |
| **I/O PERFORMANCE in MBytes/s with the data flow direction TowardNetwork/FromNetwork** | **V3 from SCRAMNet Prefetching on** | **V3 from SCRAMNet Prefetching off** |
| SCRAMNet Write Posting on | 13.7/11.0 | 13.7/4.4 |
| SCRAMNet Write Posting off | 10.4/11.0 | 10.4/4.4 |

* Results derived from a software test using a time stamp counter

# A.3 Part Number

The PCI adapter part number is in the form:

**H-AS-DPCIL2M-00**

where:

| CODE | DEFINITION | |
|---|---|---|
| H | Hardware | |
| AS | Top Level Assembly | |
| D | Standard SCRAMNet+ | |
| PCI | Standard PCI board | |
| XXX | **Memory (bytes)** | |
| | 04K | = 4 K |
| | 128 | = 128 K |
| | 512 | = 512 K |
| | L2M | = 2 M (LOW DENSITY) |
| | H2M | = 2 M (HIGH DENSITY) |
| | 04M | = 4 M |
| | 08M | = 8 M |
| X | **Transmission Media** | |
| | 0 | = NO Media Card |
| | 1 | = COAX Media Card |
| | 2 | = STANDARD FO Media Card |
| | 3 | = LONGLINK FO Media Card |
| | 4 | = LASERLINK FO Media Card |
| X | **Variable**. Used for product variations and/or modifications | |

# A.4 Board Dimensions

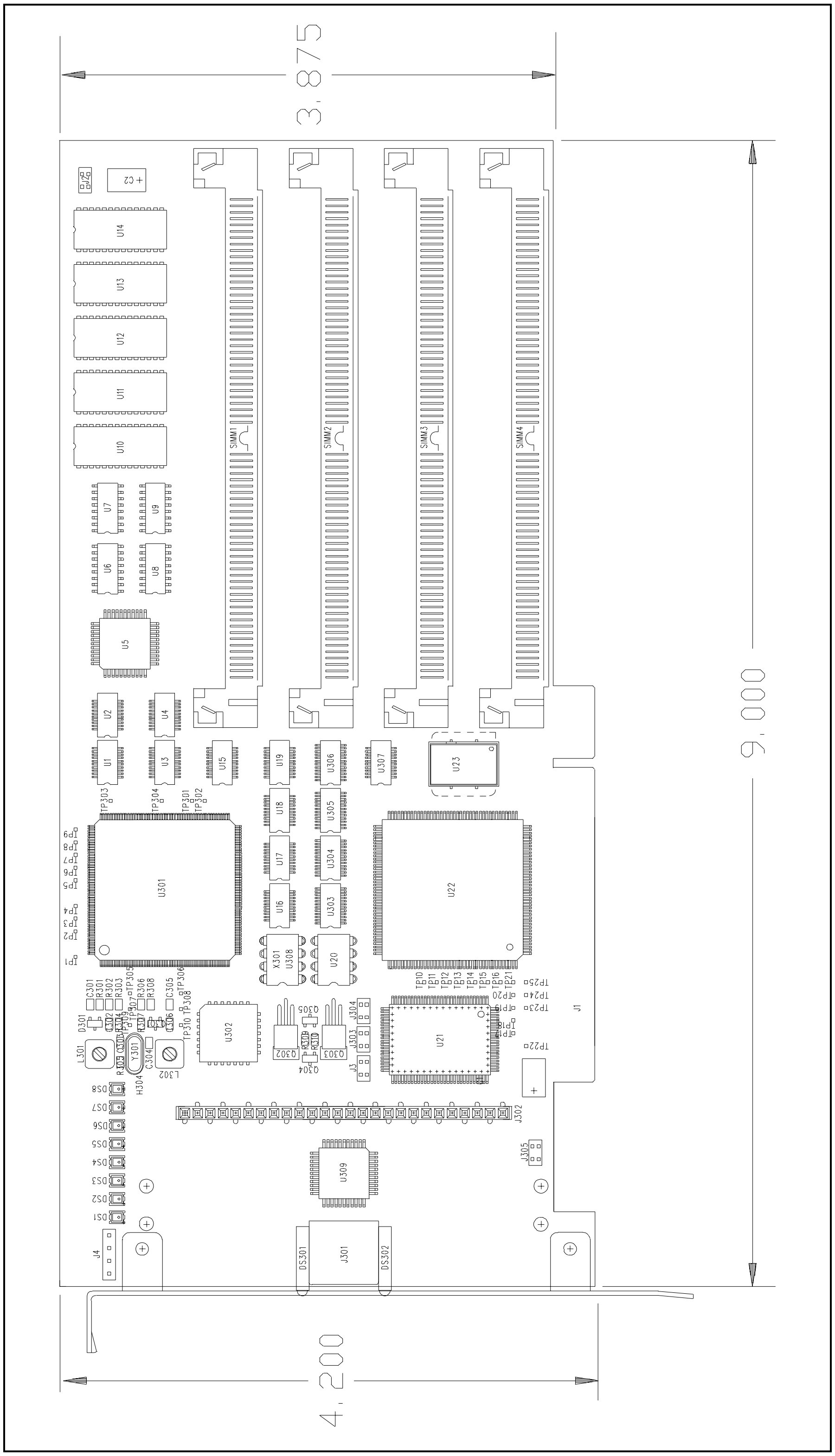
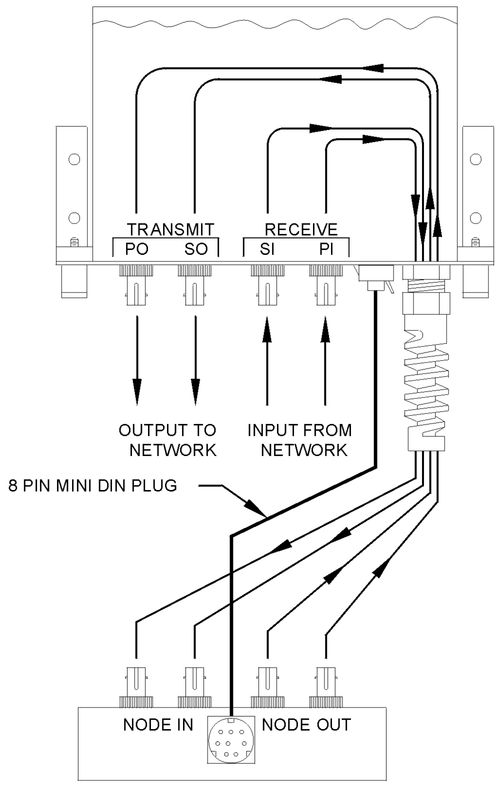



**Figure A-1 PCI Dimensions**

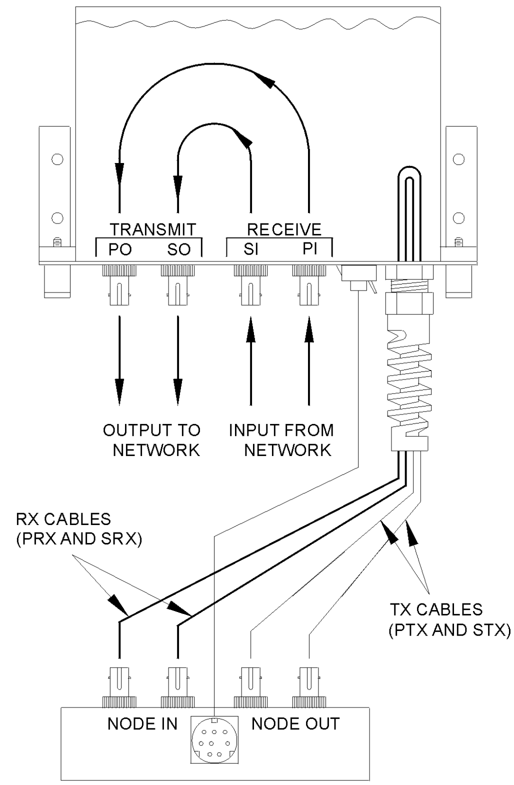
# A.5 PCI Bus Voltage Specification

| Mnemonic | Description | Allowed Variation | Ripple/Noise Below 10 MHz |
|---|---|---|---|
| +5 V | +5 V dc | +0.25 V/-0.125 V | 50 Mv |
| +12 V | +12 dc power | +0.60 V/-0.36 V | 50 Mv |
| -12 V | -12 dc power | -0.60 V/+0.36 V | 50 Mv |
| +5 V STDBY | +5 V dc standby | +0.25 V/-0.125 V | 50 Mv |
| GND | Ground | Reference | -- |

# A.6 Fiber Optic Bypass Switch




INSERTED STATE (POWER ON)




BYPASS STATE (POWER OFF)

## SPECIFICATIONS

### OPTICAL PERFORMANCE

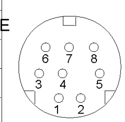
| | |
|---|---|
| INSERTION LOSS | 1.1 DB TYPICAL, 1.8dB MAXIMUM.<br>1.4 DB TYPICAL, 2.1 DB MAXIMUM. |
| LOOPBACK LOSS | 5 DB MAXIMUM. |
| SWITCHING TIME (POWER LOSS TO POINT THAT RELIABLE DATA TRANSFERS OCCUR) | 5 MILLISECONDS TYPICAL, 10 MILLISECONDS MAXIMUM. |
| OPTICAL CROSS-TALK | -45dB MAXIMUM (FOTP-42). |
| MECHANICAL SHOCK | 30 G'S ON 3 AXES, PER FOTP-14. |
| VIBRATION TOLERANCE | 15 G'S ON 3 AXES, PER FOTP-11. |
| ALLOWABLE LENGTH DIFFERENCE BETWEEN PRIMARY AND SECONDARY DATA PATHS IN "INSERT" OR "BYPASS" STATE | LESS THAN 6" (.76NS) MEASURED BY SYSTRAN SKEWMETER IN "BYPASS" MODE AT PTX AND STX TO PRX AND SRX. ALSO TX INNER AND OUTER CON' TO RX INNER AND OUTER CON'. |
| ALLOWABLE SPLICES IN DATA PATH | NONE ON "NEW" UNITS (REPAIRS ON CASE BY CASE BASIS AS CUSTOMER REQUIRES) |
| DURABILITY | ONE MILLION CYCLES MINIMUM. |
| OPERATING TEMPERATURE | -10°C TO +65°C |
| STORAGE TEMPERATURE | -30°C TO +70°C |

### ELECTRICAL PERFORMANCE

| | |
|---|---|
| SWITCHING VOLTAGE | 3.5 VDC MINIMUM VOLTAGE TO SWITCH THE RELAY |
| SWITCHING CURRENT (SYSTRAN USES 2 RELAYS IN PARALLEL) | 160 MA MAXIMUM AT 5.0 VDC |
| CONNECTOR | 8 PIN MINI DIN PLUG |

### RECEPTACLE PIN ASSIGNMENTS

| | |
|---|---|
| 1 | PRIMARY/SECONDARY SWITCH GROUND |
| 2 | NO CONNECT/RESERVED |
| 3 | PRIMARY/SECONDARY SWITCH POSITIVE |
| 4 | NO CONNECT/RESERVED |
| 5 | NO CONNECT/RESERVED |
| 6 | NO CONNECT/RESERVED |
| 7 | NO CONNECT/RESERVED |
| 8 | NO CONNECT/RESERVED |




### SWITCH LEGEND

| | |
|---|---|
| PI = PRIMARY IN | PTX = PRIMARY TRANSMIT |
| SI = SECONDARY IN | STX = SECONDARY TRANSMIT |
| PO = PRIMARY OUT | PRX = PRIMARY RECEIVE |
| SO = SECONDARY OUT | SRX = SECONDARY RECEIVE |

NOTES:

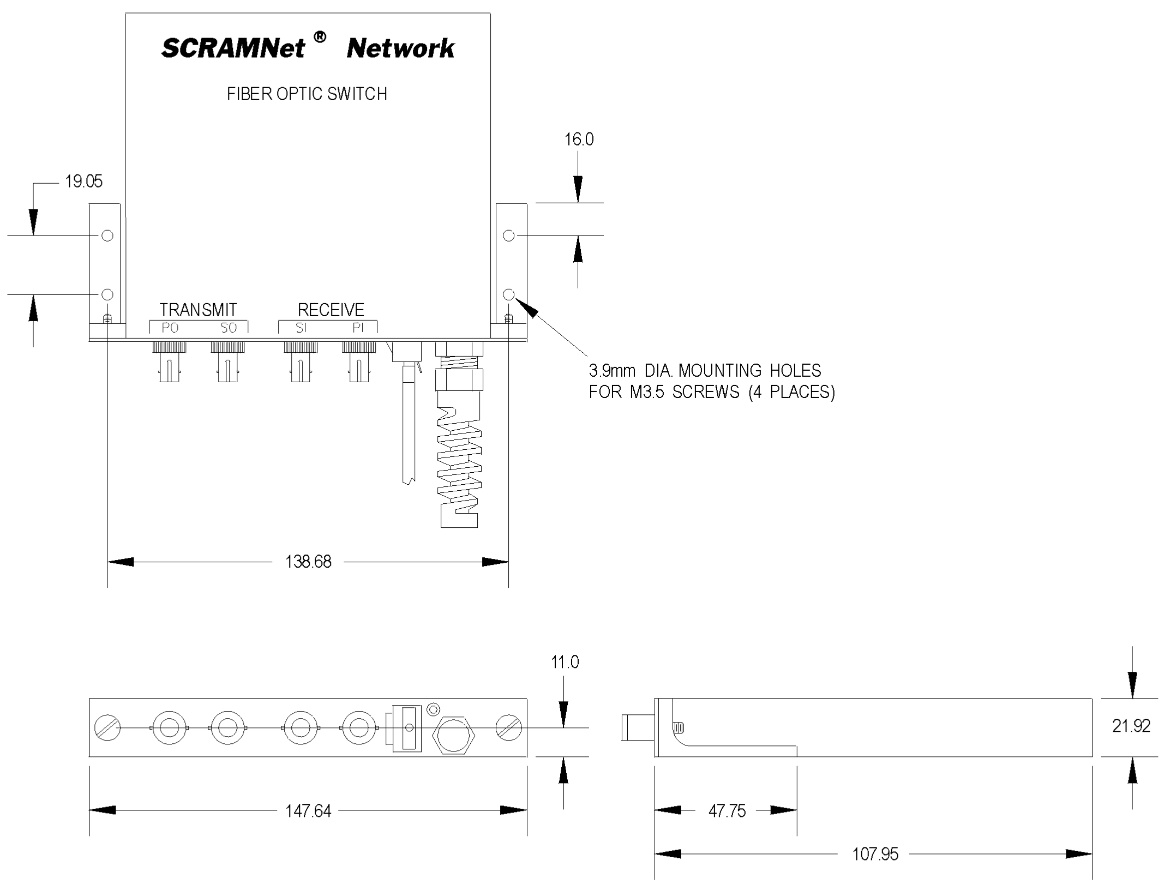
1. INSERTION LOSS MEASUREMENTS ARE BASED ON FOTP 34, METHOD B - LONG LAUNCH AND METHOD A - SHORT LAUNCH (RESPECTIVELY).
2. UNIT CONFIGURED USING ST CONNECTOR RECEPTACLES AND ST CONNECTOR PLUGS.
3. LED WILL BE "ON" WHEN IN INSERTED STATE (POWER "ON").

| | | | |
|---|---|---|---|
| SYSTRAN CORPORATION<br>4126 LINDEN AVE. SUITE 100<br>DAYTON, OHIO 45432 | SIZE<br>A | DWG. NO.<br>A-D-PR-FORELAY-6X | REV.<br>- |
| | | FILE: FORELAY6X2.DWG | SHEET 2 OF 4 |

**Figure A-2 Fiber Optic Bypass Switch**

## A.6.1 Fiber Optic Bypass Switch Dimensions

HOUSING BLUEPRINT




NOTES:
1. ALL UNITS ARE IN mm.
2. DO NOT SCALE DRAWING.

| | | | |
|---|---|---|---|
| SYSTRAN CORPORATION<br>4126 LINDEN AVE. SUITE 100<br>DAYTON, OHIO 45432 | SIZE<br>A | DWG. NO.<br>A–D–PR–FORELAY–6X | REV.<br>A |
| | | FILE: FORELAY6X3.DWG | SHEET 3 OF 4 |

**Figure A-3 Housing Dimensions**

*This page intentionally left blank*

# APPENDIX B

# CSR DESCRIPTIONS

## TABLE OF CONTENTS



## TABLES

# B.1 Description

This section describes each Control/Status Register and the function of each bit. The name of each bit is indicative of its set state.

The registers are described using bit 0 as the Least Significant Bit (LSB). For example: Inserting 0xA7C3 in a 16-bit register would set bits 0, 1, 6, 7, 8, 9, 10, 13, and 15 ON.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| A | | | | 7 | | | | C | | | | 3 | | | |

**Table B-1 CSR0 - General SCRAMNet+ SC150 Enable and Reset (Read/Write)**

| Bits | Description |
|---|---|
| 1-0 | **Network Communications Mode** - Bit 0 controls the receive enable, and Bit 1 controls the transmit enable. |
| | 00 **None** - In this mode, all communications between the node shared memory and the network is inhibited. The node is still able to pass network traffic but does not receive or transmit any data. Loopback modes are also meaningless unless the Host to Shared Memory Write bit is enabled. |
| | 01 **Receive Only** - In this mode, any received message is processed and written to node shared memory. Data written by the host is placed in the node shared memory and in the Transmit FIFO but is not sent out on the network. In this mode, the Transmit FIFO will fill and if the Error Interrupt is enabled, Transmit FIFO full interrupt will be triggered. Before changing modes from Receive-Only to either Transmit-Only or Transmit/Receive, the Transmit FIFO should be cleared. If not, all buffered transmit messages will be sent out on the network. |
| | 10 **Transmit Only** - In this mode, any received message bypasses the shared memory and is passed on. Any message written by the host to node shared memory is transmitted on the network. However, any received message is not written to node shared memory. (Transmissions are subject to data filter characteristics.) |
| | 11 **Transmit/Receive** - In this mode, any received message is processed, written to node shared memory and passed on. Any message written by the host to node shared memory is transmitted on the network. This is the normal operation. (Transmissions are subject to data filter characteristics.) |
| 2 | **Redundant Transceiver Toggle** - When this bit is cycled ‘0’, ‘1’, ‘0’, the optional redundant transceiver selected link is changed. |
| 3 | **Host Interrupt Enable** - When this bit is set, a received message that is written to node shared memory as an interrupt will generate an interrupt request, and the address will be written to the Interrupt FIFO. This bit must be set in order to receive any interrupts from the network. |
| 4 | **Auxiliary Control RAM Enable** - When this bit is set, the ACR bytes are swapped in place of the corresponding least-significant byte of every four-byte word in SCRAMNet+ memory. The values written to those ACR byte locations dictate the type of interrupt that will occur when the 4-byte memory location is written into. The ACR has five bits for interrupt control. They are as follows: |
| | ACR[0] - **Receive Interrupt Enable** - Setting this bit generates an interrupt to the host for network interrupt data received in this location. |
| | ACR[1] - **Transmit Interrupt Enable** - Setting this bit generates an interrupt to the network for a host write to this shared memory location. |
| | ACR[2] - **External Trigger 1** - Setting this bit generates a trigger signal to an external connector whenever there is a host write access to this shared memory location. |
| | ACR[3] - **External Trigger 2** - Setting this bit generates a trigger signal to an external connector whenever there is a network write to this shared memory location. |
| | ACR[4] - **HIPRO location enable** - Setting this bit causes the two 16-bit data or four 8-bit items within the 32-bit address boundary to be transmitted as one 32-bit network message. CSR2, bit 13 must also be set for this action to occur. |
| 5 | **Interrupt On Memory Mask Match Enable** - This bit must be set in order for any type of memory interrupt to occur. |
| 6 | **Override Receive Interrupt Enable Flag** - When this bit is set, an interrupt is generated to the host by any interrupt data received from the network regardless of the status of the ACR Receive Interrupt bit. |
| 7 | **Enable Interrupt on Error** - When this bit is set, Interrupt FIFO Full, Protocol Violation, Bad Message and/or Receiver Overflow conditions causes an interrupt request. |

| Bits | Description |
|---|---|
| 8 | **Network Interrupt Enable** - This bit must be set to transmit interrupt data to the network. |
| 9 | **Override Transmit Interrupt Enable Flag** - When this bit is set, an interrupt is sent out on the network regardless of the status of the ACR Transmit Interrupt bit. |
| 10 | **Enable Transmit Data Filter** - When clear, the entire address space is not filtered and the node is capable of transmitting all messages written to the node shared memory by the host on the network. When set, the data-filter function is enabled for the address space above the first 4 K bytes of SCRAMNet+ memory. Bit 11 controls the lower 4 K bytes. |
| 11 | **Enable Lower 4 K Bytes For Data Filter** - When set, the lower 4 K bytes of address space is data filtered if bit 10 is also set. When disabled, the address space will not be filtered. |
| 12 | **Reset Receive/Transmit FIFO** - This bit must be toggled from ‘0’ to ‘1’ and back to ‘0’ in order to reset the Receive/Transmit FIFO. The R/T FIFO is a temporary high-speed holding area for data flowing through the network.<br>**NOTE**: If the R/T FIFO were to be reset during active network transmissions, the data in the FIFO at that time would be lost and it would cause errors on the downstream nodes in the network ring. |
| 13 | **Reset Interrupt FIFO** - This bit must be toggled from ‘0’ to ‘1’ and back to ‘0’ to reset the Interrupt FIFO. |
| 14 | **Reset Transmit FIFO** - This bit must be toggled from ‘0’ to ‘1’ and back to ‘0’ to reset the Transmit FIFO. |
| 15 | **Insert Node** - This bit controls the nodes communications mode on the network as either a receiver only or a receiver/transmitter. On power-up, this bit is OFF which translates to the receiver-only mode. This allows user-written software (on each host processor on the network) to be initiated from one node whenever the network is started cold. When this bit is ON, the node is “inserted” into the network ring as a receiver/transmitter which is the normal operating mode if the Fiber Optic Loopback CSR2[6] is disabled. This bit is invalid when the Enable Wire Loopback CSR2[7] is ON. |

**Table B-2 CSR1 - SCRAMNet+ SC150 Error Indicators (Read Only with Write/Reset for interrupts)**

**NOTE**: Reading CSR1 will reset the latched error conditions by clearing bits 0,2,4,6,7,8,9,10,11,12,13.

| Bits | Description |
|---|---|
| 0 | **Transmit FIFO Full (Latched)** - When this bit is set, the Transmit-FIFO-Full condition exists. This occurs when there is more data coming from the host to the network than the network can absorb. When the shared memory is full, host writes will be held off by the SCRAMNet+ SC150 host interface logic until the Transmit FIFO is no longer full. |
| 1 | **Transmit FIFO Not Empty** - This bit does not represent any type of error condition, but rather just a report on the state of the Transmit FIFO. A ‘0’ represents an empty FIFO, where a ‘1’ indicates at least one message in the FIFO. |
| 2 | **Transmit FIFO 7/8 Full (Latched)** - This bit indicates that the Transmit FIFO is 7/8 full. A ‘0’ represents a FIFO that is less than 7/8 full, where a ‘1’ indicates the FIFO is backing up and is more than 7/8 full. |
| 3 | Always 0 |
| 4 | **Interrupt FIFO Full (Latched)** - When this bit is set, the Interrupt FIFO Full error condition exists. Reset the Interrupt FIFO by toggling CSR0[13] to ON then to OFF. |
| 5 | **Protocol Violation (Latched)** - When this bit is ON, there has been a signal error at the physical layer (fiber or coaxial) resulting from noise on the transmission lines or a result of hardware failure. It can be any one of the following: Missing transition for two clock periods on either line, Parity error or a Framing error. |
| 6 | **Carrier Detect (Latched)** - This bit is set if the receivers do not detect any or enough output from the previous nodes transmitters. This is usually an indication that the fiber optic lines have become disconnected or there may be dust/dirt where the fiber optic connections have been made. A visual inspection of the network lines will need to be made. |
| 7 | **Bad Message (Latched)** - When this bit is set, the hardware has detected an error in the message packet received on the network. If this error persists, it is an indication that a hardware problem on the SCRAMNet+ board may exist. |
| 8 | **Receiver Overflow (Latched)** - When this bit is set, the Receive FIFO has received more data than the node is able to process. This condition may indicate a hardware problem on the board. |
| 9 | **Transmit Retry (Latched)** - This bit is set if a message returns to the originating node with bit errors. The message is automatically retransmitted indefinitely until it returns without bit errors. This is considered to be an error condition. |
| 10 | **Transmit Retry Time-out (Latched)** - This bit is set if a message does not return to the originating node within the time-out value specified in CSR5. The message is automatically retransmitted indefinitely until it returns. This is considered to be an error condition. |
| 11 | **Redundant Transmit/Receive Fault (Latched)** - This bit is set if the currently selected optional redundant transceiver has faulted and reverted to the other link. The default value is ‘0’ |
| 12 | **General Purpose Counter/Timer Overflow (Latched)** - This bit toggles a 16-bit counter/timer. The events to be counted/timed are set using CSR8[9]; CSR9[13]; and CSR9[14]. The output is held in CSR13. The counter/timer can: count errors, count trigger events for triggers 1 and 2, transmit time, network events, free run @ 26.66 ns, and free run @ 1.706 ns with trigger 2 CLEAR. |

| Bits | Description |
|---|---|
| 13 | **Current Link (Latched)** - This bit tells which of the optional redundant transceivers is currently selected as the active link. The default value is 1=A. |
| 14 | **Interrupts Armed** - During interrupt operation, this bit indicates that the conditions to receive an interrupt are active. If this bit is ‘0’, then the host will receive no interrupts. When CSR1 is written to, then the interrupts-armed bit will return to an active status. |
| 15 | **Fiber Optic Bypass Not Connected** - This is a status bit concerning the installation of the optional Fiber Optic Bypass Switch. A ‘0’ in this bit indicates that the bypass switch is installed while a ‘1’ indicates it is not installed. Fiber Optic Loopback CSR2[6] mode is dependent upon the Fiber Optic Bypass Switch being installed. |

**Table B-3 CSR2 - Node Control (Read/Write)**

| Bits | Description |
|---|---|
| 5-0 | These bits are related to lines connected through the MUX control port and are available to the host interface. They are not required to connect to anything |
| 6 | **Disable Fiber Optic Loopback** - When this bit is '0' (power up default), the output of the transmitter is connected by fiber optics directly to the input of the receiver, and the receiver is disconnected from the network. The optional Fiber Optic Bypass Switch must be installed for this mode to be effective. This mode is valid only when the Insert Node CSR0[15] is ON. Set this bit to disable the loopback mode when the node is in use as a part of the network. |
| 7 | **Enable Wire Loopback** - When this bit is set, the output of the transmitter is connected by wire directly to the input of the receiver, and the receiver is disconnected from the network. The purpose of this bit is purely diagnostic. This mode is valid only when the Insert Node CSR0[15] is OFF. |
| 8 | **Disable Host to Memory Write** - When this bit is set, the host writes are not written to the host node's shared memory, but are sent out on the network if Transmit CSR0[1] is ON. |
| 9 | **Write Own Slot Enable** - When this bit is set, the message slot (packet) sent out to the network can be received by the originating node. This is not the normal procedure but may be used (in conjunction with CSR2[10] when it is desired to generate an interrupt to the host, written by the host. |
| 10 | **Enable Interrupt On Own Slot** - When this bit is set, a message with the interrupt bit set can be received by the originating node if CSR2[9], is also set. This coupling enables a host processor to interrupt itself (Self Interrupt). |

**Write-Me-Last/Self-Interrupt Mode Definition**

| CSR2[10] | CSR2[9] | CSR2[8] | Mode |
|---|---|---|---|
| 0 | 1 | 1 | WRITE ME LAST mode |
| 1 | 1 | 0 | SELF-INTERRUPT mode |
| 1 | 1 | 1 | WRITE ME LAST with SELF-INTERRUPT mode |

| | |
|---|---|
| 11 | **Message Length Limit** - Variable maximum message size: 1024 bytes or 256 byte. It is used in conjunction with CSR2[12], CSR2[14] and CSR2[15] to enable PLUS mode communication protocols. |
| 12 | **Variable Length Messages on Network** - When ON, this bit enables variable length messages. It is used in conjunction with CSR2[11], CSR2[14] and CSR2[15] to enable PLUS mode communication protocols (see below). |
| 13 | **HIPRO Enable** - When this bit is set, the two 16-bit shortwords associated with the longword addressed at ACR[4], will be transmitted onto the network as one 32-bit longword. The first shortword write will be held until the second shortword write occurs, which results in the 32-bit data value to be written to the network.<br>**Exceptions:**<br>HIPRO will not work when Disable Host to Memory Write CSR2[8] is set.<br>HIPRO will not work when writing two separate shortwords while using interrupts. |
| 14 | **Multiple Messages** - This bit allows multiple native messages on the network. It is used in conjunction with CSR2[11], CSR2[12] and CSR2[15] to enable the BURST mode communication protocol (see below). |

| Bits | Description |
|---|---|
| 15 | **No Network Error Correction** - This bit is used in conjunction with CSR2[12] and CSR2[14] to enable communication protocols: BURST or PLATINUM mode and the variable length message PLUS (+) mode (see below). |

**SCRAMNet+ SC150 Protocol Mode Definition**

| Network Mode | CSR2[15] No Error Correction | CSR2[14] Multiple Message | CSR2[12] Variable Length | CSR2[11] Message Size Maximum |
|---|---|---|---|---|
| **BURST** | 1 | 1 | 0 | NO MEANING |
| **PLATINUM** | 0 | 1 | 0 | NO MEANING |
| **BURST+** | 1 | 1 | 1 | 1=1024, 0=256 |
| **PLATINUM+** | 0 | 1 | 1 | 1=1024, 0=256 |

**Table B-4 CSR3 - Node Information (Read Only)**

| Bits | Description |
|---|---|
| 7-0 | **Node Number Count** - These bits represent the total number of SCRAMNet+ SC150 nodes on the network. This value is dynamically determined by the hardware and ranges from 0 to 255 depending upon the number of nodes actually on the network.<br>**Transmit AGE** - This field is also used to read/write the T_AGE[7:0] field. This register reflects this field when the ID_MUX bit in CSR8[0] is set. |
| 15-8 | **Node Identification Number** - These bits represent the SCRAMNet+ SC150 node identification number. Each node must have a unique identification number from 0 to 255 for each network ring. The NODE ID need not be in sequential order.<br>**Receive ID** - This field is also used to read/write the RXID[7:0] field. This register reflects this field when the ID_MUX bit in CSR8[0] is set. |

**Table B-5 CSR4 - Interrupt Address (LSP) (Read Only)**

| Bits | Description |
|---|---|
| 15-0 | **LSP of the Interrupt Address** - These bits represent the LSP of the interrupt address (A15 - A0). Bits 0 and 1 are always '0' since the addresses are on four-byte boundaries. |

**Table B-6 CSR5 - Interrupt Address and Status (Read Only)***

| Bits | Description |
|---|---|
| 6-0 | **MSP of the Interrupt Address** - These 7 bits represent the MSP of the interrupt address (A22 - A16). When coupled with CSR4, this address represents the SCRAMNet+ memory location of the interrupt. |
| 13-7 | Reserved. |
| 14 | **Retry Interrupt FIFO Bit** - This bit is set when an interrupt message is received that has its message retry bit set. This can be checked in the interrupt service routine to guard against double interrupts from the same message if it happens to be retransmitted. |
| 15 | **Interrupt FIFO Not Empty** - When this bit is clear, the Interrupt FIFO is empty. Do not read CSR4 when this bit is '0'. When this bit is set, it signals that CSR5 and CSR4 contain a legitimate interrupt address. |

* Writing the Transmit Time-out value to CSR5 stores it in shadow memory. Do not set this value to '0'. A value of '0' prevents host-generated data from leaving the Transmit FIFO.

**Table B-7 CSR6 - Reserved**

| Bits | Description |
|---|---|
| 15-0 | Not Used |

**Table B-8 CSR7 - Reserved**

| Bits | Description |
|---|---|
| 15-0 | Not Used |

**Table B-9 CSR8 - General SCRAMNet+ SC150 Extended Control Register**

| Bits | Description |
|---|---|
| 0 | **ID Multiplex** - When set to '1', CSR3 contains the T_AGE and RXID fields. |
| 1 | **Disable Holdoff** - When set, this bit disables the Holdoff feature. |
| 7-2 | These bits are used for programming the EEPROM. |
| 8 | **CSR Reset** - Setting this bit causes bus errors. On reset, CSRs loads from EEPROM. |
| 9 | **General Purpose Counter/Timer Free Run** - Setting this bit causes the GPC to free run at a rate of 37.5 MHz (26.66 ns). This counter mode overrides all other counter mode settings. |
| 10 | **Receive Interrupt Override** - When this bit is set, all incoming network messages are treated as interrupt messages. |
| 11 | **Mechanical Switch Override** - Normally set to ON. When OFF, Mechanical Switch Loopback Mode is invoked. |
| 14-12 | **Memory Size Configuration** - These bits indicate the memory-size code and are used in conjunction with the memory address stored in CSR10 and 11. The memory size is automatically calculated. (See below) |
| 15 | Reserved. (Always 1). |

## Memory Size Configuration

| CSR8[14] | CSR8[13] | CSR8[12] | Memory Size |
|---|---|---|---|
| 1 | 1 | 1 | 4 KB |
| 1 | 1 | 0 | 128 KB |
| 1 | 0 | 1 | 512 KB |
| 1 | 0 | 0 | 1 MB |
| 0 | 1 | 1 | 2 MB |
| 0 | 1 | 0 | 4 MB |
| 0 | 0 | 1 | 8 MB |

**Table B-10 CSR9 - SCRAMNet+ SC150 Interrupt On-Error Mask***

| Bits | Description |
|---|---|
| 0 | Transmit FIFO Full Mask |
| 1 | Transmit FIFO not Empty Mask |
| 2 | Transmit FIFO 7/8 Full Mask |
| 3 | Built In Self Test Stream (BIST) - Internal 82-bit BIST shift register output. |
| 4 | Interrupt FIFO Full Mask |
| 5 | Protocol Violation Mask |
| 6 | Carrier Detect Fail Mask |
| 7 | Bad Message Mask |
| 8 | Receiver Overflow Mask |
| 9 | Transmitter Retry Mask |
| 10 | Transmitter Retry Due to Time Out Mask |
| 11 | Redundant TX/RX Fault Mask |
| 12 | Interrupt on General Purpose Counter/Timer Overflow Mask |
| 13 | See Below |
| 14 | See Below |
| 15 | Fiber Optic Bypass Switch Not Connected Mask |

**General Purpose Counter/Timer Modes**

| CSR8[9] | CSR9[14] | CSR9[13] | Counter/Timer Modes |
|---|---|---|---|
| 0 | 0 | 0 | Count Errors |
| 0 | 0 | 1 | Count Trigs (1&2) |
| 0 | 1 | 0 | Transit Time |
| 0 | 1 | 1 | Network Events |
| 1 | 1 | X | Free Run @ 26.66 ns |
| 1 | 0 | 1 | 1.706 µs w/trig 2 CLR |

* To enable an On-Error mask, set the bit to ‘1’.

**Table B-11 CSR10 – Write Posting Control**

| Bits | Description |
|---|---|
| 0 | 1 = Write Posting enabled<br>0 = Write Posting disabled<br>**NOTE**: This CSR is shadowed in the controller state machine so that its value, when written, can be read from CSR10. |
| 15-1 | Not Used |

**Table B-12 CSR11 - Reserved**

| Bits | Description |
|---|---|
| 15-0 | Not Used |

**Table B-13 CSR12 - SCRAMNet+ SC150 Virtual Paging Register**

| Bits | Description | |
|---|---|---|
| 0 | VP | **Virtual Paging Enable**. When ON, this bit enables Virtual Paging. |
| 4-1 | -0- | Always zero |
| 5 | VP_A12 | **Virtual Page number.** The significance of this register is dependent on the memory size. (e.g. For 4 MB, only VP_A22 is valid; for 4 KB, VP_A[22:12] are valid. |
| 6 | VP_A13 | |
| 7 | VP_A14 | |
| 8 | VP_A15 | |
| 9 | VP_A16 | |
| 10 | VP_A17 | |
| 11 | VP_A18 | |
| 12 | VP_A19 | |
| 13 | VP_A20 | |
| 14 | VP_A21 | |
| 15 | VP_A22 | |

**Table B-14 CSR13 - General Purpose Counter/Timer**

| Bits | Description | |
|---|---|---|
| 0 | RD_COUNT[0] | This is a General Purpose Counter/Timer register. It can be used to count trigger 1 and 2 events, count errors, or other events as programmed by CSR9[13] and CSR9[14]. |
| 1 | RD_COUNT[1] | |
| 2 | RD_COUNT[2] | |
| 3 | RD_COUNT[3] | |
| 4 | RD_COUNT[4] | |
| 5 | RD_COUNT[5] | |
| 6 | RD_COUNT[6] | |
| 7 | RD_COUNT[7] | |
| 8 | RD_COUNT[8] | |
| 9 | RD_COUNT[9] | |
| 10 | RD_COUNT[10] | |
| 11 | RD_COUNT[11] | |
| 12 | RD_COUNT[12] | |
| 13 | RD_COUNT[13] | |
| 14 | RD_COUNT[14] | |
| 15 | RD_COUNT[15] | |

**Table B-15 CSR14 - Reserved**

| Bits | Description |
|---|---|
| 15-0 | Not Used |

**Table B-16 CSR15 - Reserved**

| Bits | Description |
|---|---|
| 15-0 | Not Used |

# APPENDIX C

# CSR SUMMARY

## TABLE OF CONTENTS

# C.1 CSR0 - General SCRAMNet+ Enable and Reset

| Bit | Function | Name |
|---|---|---|
| 0 | Receive Enable | RX_ENB |
| 1 | Transmit Enable | TXEN |
| 2 | Redundant TxRx Toggle | RTT |
| 3 | Host Interrupt Enable | HIE |
| 4 | Auxiliary Control RAM Enable | ACRE |
| 5 | Interrupt on Memory Mask Match Enable | IMME |
| 6 | Override RIE Flag | ORF |
| 7 | Interrupt on Errors | IOE |
| 8 | Network Interrupt Enable | NIE |
| 9 | Override TIE Flag | OTF |
| 10 | Enable Tx Data Filter | DFEN |
| 11 | Enable Lower 4 Kbytes for Data Filter | EN4K |
| 12 | RESET Rx/Tx FIFO | RTRF |
| 13 | RESET Interrupt FIFO | RSTIF |
| 14 | RESET Transmit FIFO | RTXF |
| 15 | Insert Node | INSRT |

# C.2 CSR1 - Error Indicators

| Bit | Function | Name |
|---|---|---|
| 0 | Transmit FIFO Full | TXFF |
| 1 | Transmit FIFO Not Empty | TXFNE |
| 2 | Transmit FIFO 7/8 Full | TXFAF |
| 3 | (Always 0) | Not Used |
| 4 | Interrupt FIFO Full | IFF |
| 5 | Protocol Violation | PV |
| 6 | Carrier Detect Failure | CDF |
| 7 | Bad Message | BB |
| 8 | Receiver Overflow | RXO |
| 9 | Transmit Retry | TXRTY |
| 10 | Transmit Retry Time-out | TO |
| 11 | Redundant TxRx Fault | RTF |
| 12 | General Purpose Counter/Timer Overflow | GPCTO |
| 13 | Redundant TxRx Link 1=A/0=B | RTLAB |
| 14 | Interrupts Armed - Write to re-arm | IARM |
| 15 | Fiber Optic Bypass Not Connected | FOB |

# C.3 CSR2 - Node Control

| Bit | Function | Name |
|---|---|---|
| 5-0 | Available to Host | |
| 6 | Disable Fiber Optics Loopback | FO_DIS |
| 7 | Enable Wire Loopback | EN_WR_LPB |
| 8 | Disable Host to Memory Write | DIS_H_M_WR |
| 9 | Enable Write of Our Own Slot to Memory | WOSEN |
| 10 | Enable Interrupt on Receipt of Own Interrupt Slot | IOSEN |
| 11 | 1024 vs 256 variable size max (bytes) | LEN_LIMIT |
| 12 | Variable length messages on network | VAR_LEN |
| 13 | HIPRO Write Enable | HIPRO |
| 14 | Allow multiple native messages on network | MULTIPLE_MSG |
| 15 | No Network Error Correction | NO_ERR_CRCT |

**Write-Me-Last/Self-Interrupt Mode Definition**

| CSR2[10] | CSR2[9] | CSR2[8] | Mode |
|---|---|---|---|
| 0 | 1 | 1 | WRITE ME LAST mode |
| 1 | 1 | 0 | SELF-INTERRUPT mode |
| 1 | 1 | 1 | WRITE ME LAST with SELF-INTERRUPT mode |

**SCRAMNet+ Protocol Mode Definition**

| Network Mode | CSR2[15] No Error Correction | CSR2[14] Multiple Message | CSR2[12] Variable Length | CSR2[11] Message Size Maximum |
|---|---|---|---|---|
| **BURST** | 1 | 1 | 0 | NO MEANING |
| **PLATINUM** | 0 | 1 | 0 | NO MEANING |
| **BURST+** | 1 | 1 | 1 | 1=1024, 0=256 |
| **PLATINUM+** | 0 | 1 | 1 | 1=1024, 0=256 |

# C.4 CSR3 - Node Information

| Bit | Function | Name |
|---|---|---|
| 0 | Node Number Count* (Valid After a Transmission from the Node) | NN0 |
| 1 | | NN1 |
| 2 | | NN2 |
| 3 | | NN3 |
| 4 | | NN4 |
| 5 | | NN5 |
| 6 | | NN6 |
| 7 | | NN7 |
| 8 | Node ID Number* | NID0 |
| 9 | | NID1 |
| 10 | | NID2 |
| 11 | | NID3 |
| 12 | | NID4 |
| 13 | | NID5 |
| 14 | | NID6 |
| 15 | | NID7 |

* When ID_MUX CSR[0] is set:
Bits 7 - 0 are Transmit AGE
Bits 15 - 8 are Receive ID.

# C.5 CSR4 - Interrupt Address (LSW)

| Bit | Function | Name |
|---|---|---|
| 0 | Interrupt FIFO Address Field (LSW) | Always = 0 |
| 1 | | Always = 0 |
| 2 | | RFA2 |
| 3 | | RFA3 |
| 4 | | RFA4 |
| 5 | | RFA5 |
| 6 | | RFA6 |
| 7 | | RFA7 |
| 8 | | RFA8 |
| 9 | | RFA9 |
| 10 | | RFA10 |
| 11 | | RFA11 |
| 12 | | RFA12 |
| 13 | | RFA13 |
| 14 | | RFA14 |
| 15 | | RFA15 |

# C.6 CSR5 - Interrupt Address (MSW) and Status (Read Only*)

| Bit | Function | Name |
|---|---|---|
| 0 | Interrupt FIFO Address Field (MSW) | RFA16 |
| 1 | | RFA17 |
| 2 | | RFA18 |
| 3 | | RFA19 |
| 4 | | RFA20 |
| 5 | | RFA21 |
| 6 | | RFA22 |
| 13-7 | Reserved | 0 |
| 14 | Retry Bit in Interrupt FIFO | (RF_RETRY) |
| 15 | Interrupt FIFO Not Empty | (~RX_F_E) |

* Writing the Transmit Time-out value to CSR5 stores it in shadow memory. Do not set this value to '0'. A value of '0' prevents host-generated data from leaving the Transmit FIFO

# C.7 CSR6 - Reserved

A 16-bit, Read Only SYSTRAN reserved register.

# C.8 CSR7 - Reserved

A 16-bit, Read Only SYSTRAN reserved register.

# C.9 CSR8 - General SCRAMNet+ Extended Control Register

| Bit | Function | Name |
|---|---|---|
| 0 | 1 is CSR3=T_AGE & RXID fields | ID_MUX |
| 1 | Disable Holdoff feature | DIS_HOLD |
| 2 | Chip select to EEPROM | CSR_CS0 |
| 3 | Ext. Chip Select for AUX MICROWIRE peripheral | CSR_CS1 |
| 4 | MICROWIRE DOUT pin | CSR_DOUT |
| 5 | EEPROM program enable | E_PRE |
| 6 | CLK line to MICROWIRE port | CSR_CK |
| 7 | DIN line connected to the MICROWIRE DOUT pins | E_DIN |
| 8 | Initiate initiation sequence - CSR Reset | CSR_RST |
| 9 | Override Counter mode | GPC_FRE |
| 10 | Receive Interrupt Override | RX_INT_OVR |
| 11 | 1 = Mechanical Switch Override<br>0 = Invoke Coax Loopback Mode | C_MECHSW |
| 12 | Memory Size Configuration - (See below) | MC10 |
| 13 | Memory Size Configuration - (See below) | MC11 |
| 14 | Memory Size Configuration - (See below) | MC12 |
| 15 | Reserved (always 1) | 1 |

**Memory Size Configuration**

| CSR8[14] | CSR8[13] | CSR8[12] | Memory Size |
|---|---|---|---|
| 1 | 1 | 1 | 4 KB |
| 1 | 1 | 0 | 128 KB |
| 1 | 0 | 1 | 512 KB |
| 1 | 0 | 0 | 1 MB |
| 0 | 1 | 1 | 2 MB |
| 0 | 1 | 0 | 4 MB |
| 0 | 0 | 1 | 8 MB |

# C.10 CSR9 - SCRAMNet+ Interrupt-On-Error Mask

| Bit | Function | Name |
|---|---|---|
| 0 | Transmit FIFO Full mask | M_TX_F_F |
| 1 | Transmit FIFO Not Empty mask | M_TX_F_E |
| 2 | Transmit FIFO 7/8 Full Mask | M_TX_F_AF |
| 3 | Internal 82 bit BIST shift register output | BIST_STREAM |
| 4 | Receiver FIFO Full Mask | M_RX_F_F |
| 5 | Protocol Violation mask | M_PV |
| 6 | Carrier Detect Fail mask | M_CD_FAIL |
| 7 | Bad Message mask | M_BM |
| 8 | Receiver Overflow mask | M_RX_OVR |
| 9 | Transmitter Retry mask | M_RETRY |
| 10 | Transmitter Retry - Time-out | M_RETRY_T_O |
| 11 | Redundant Transmit/Receive Fault mask | M_FAULT |
| 12 | Interrupt on Utility Counter Overflow | M_COUNT_OVR |
| 13 | General Purpose Counter/Timer Modes (See below) | M_INC_TRIGS |
| 14 | General Purpose Counter/Timer Modes (See below) | M_INC_ERRS |
| 15 | Fiber Optic Bypass Not Connected mask | M_FO_BYPASS |

**General Purpose Counter/Timer Modes**

| CSR8[9] | CSR9[14] | CSR9[13] | Counter/Timer Modes |
|---|---|---|---|
| 0 | 0 | 0 | Count Errors |
| 0 | 0 | 1 | Count Triggers (1 & 2) |
| 0 | 1 | 0 | Transit Time |
| 0 | 1 | 1 | Network Events |
| 1 | 1 | X | Free Run @ 26.66 ns |
| 1 | 0 | 1 | 1.706 µs\w Trig 2 CLR |

## C.11 CSR10 - Write Posting Control

| Bit | Function | Name |
|---|---|---|
| 0 | 1 = Write Posting enabled<br>0 = Write Posting disabled | |
| 15-1 | Not Used | SMA15 |

## C.12 CSR11 - Reserved

A 16-bit, Read Only SYSTRAN reserved register.

# C.13 CSR12 - SCRAMNet+ Virtual Paging Register

(Refer to Section 4, paragraph 4.2.1, and Section 5, page 5-15 for additional information)

| Bit | Function | Name |
|---|---|---|
| 0 | Enables Virtual Paging when set | VP |
| 4-1 | Always ‘0’ | 0 |
| 5 | Virtual Page Number | VP_A12 |
| 6 | | VP_A13 |
| 7 | | VP_A14 |
| 8 | | VP_A15 |
| 9 | | VP_A16 |
| 10 | | VP_A17 |
| 11 | | VP_A18 |
| 12 | | VP_A19 |
| 13 | | VP_A20 |
| 14 | | VP_A21 |
| 15 | | VP_A22 |

## C.14 CSR13 - SCRAMNet+ General Purpose Counter Timer

(Refer to Section 4, paragraph 4.9, and Section 5, page 5-16 for additional information)

| Bit | Function | Name |
|---|---|---|
| 0 | Counter/Timer register | RD_COUNT[0] |
| 1 | Counter/Timer register | RD_COUNT[1] |
| 2 | Counter/Timer register | RD_COUNT[2] |
| 3 | Counter/Timer register | RD_COUNT[3] |
| 4 | Counter/Timer register | RD_COUNT[4] |
| 5 | Counter/Timer register | RD_COUNT[5] |
| 6 | Counter/Timer register | RD_COUNT[6] |
| 7 | Counter/Timer register | RD_COUNT[7] |
| 8 | Counter/Timer register | RD_COUNT[8] |
| 9 | Counter/Timer register | RD_COUNT[9] |
| 10 | Counter/Timer register | RD_COUNT[10] |
| 11 | Counter/Timer register | RD_COUNT[11] |
| 12 | Counter/Timer register | RD_COUNT[12] |
| 13 | Counter/Timer register | RD_COUNT[13] |
| 14 | Counter/Timer register | RD_COUNT[14] |
| 15 | Counter/Timer register | RD_COUNT[15] |

## C.15 CSR14 - Reserved

A 16-bit, Read Only SYSTRAN reserved register.

## C.16 CSR15 - Reserved

A 16-bit, Read Only SYSTRAN reserved register.

# C.17 Auxiliary Control RAM (R/W)

| Bit | Function | Name |
|---|---|---|
| 0 | Receive Interrupt Enable | RIE |
| 1 | Transmit Interrupt Enable | TIE |
| 2 | External Trigger 1 (Host Read/Write) | ET1 |
| 3 | External Trigger 2 (Network Write) | ET2 |
| 4 | HIPRO | HIPRO |
| 7-5 | Reserved | 0 |

# APPENDIX D

# CONFIGURATION AIDS

# SCRAMNet+ CONTROL/STATUS REGISTERS REFERENCE SHEET

**CSR 0**

| Bit | |
|---|---|
| 0 | RX ENB |
| 1 | TX ENB |
| 2 | REDUND LINK TOGGLE |
| 3 | HOST INT ENB |
| 4 | AUX CTRL RAM ENB |
| 5 | INT MEM MASK MATCH |
| 6 | OVRD RIE FLAG |
| 7 | INT ON ERRORS |
| 8 | NET INT ENB |
| 9 | OVRD TIE FLAG |
| 10 | ENB TX DATA FILTER |
| 11 | ENB LOWER 4K FILTER |
| 12 | RST TX/RX FIFO |
| 13 | RST INT FIFO |
| 14 | RST TX FIFO |
| 15 | INSERT NODE |

**CSR 2**

| Bit | |
|---|---|
| 0 | available to host |
| 1 | available to host |
| 2 | available to host |
| 3 | available to host |
| 4 | available to host |
| 5 | available to host |
| 6 | DSB FO LPBCK |
| 7 | ENB WIRE LPBCK |
| 8 | DSB HOST TO SM WRT |
| 9 | ENB WRT OWN SLOT |
| 10 | ENB INT RX OWN SLOT |
| 11 | MSG LENGTH LIMIT |
| 12 | VAR LENGTH MSGS |
| 13 | ENB HIPRO WRITE |
| 14 | MULT NATIVE MSGS |
| 15 | NO NTWK ERR CRCT |

**CSR 4**

| Bit | |
|---|---|
| 0 | always 0 |
| 1 | always 0 |
| 2 | RFA 2 |
| 3 | |
| 4 | |
| 5 | RX FIFO ADDRESS FIELD |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |
| 11 | |
| 12 | |
| 13 | |
| 14 | |
| 15 | RFA 15 |

**CSR 6**

| Bit | |
|---|---|
| 0 | reserved |
| 1 | reserved |
| 2 | reserved |
| 3 | reserved |
| 4 | reserved |
| 5 | reserved |
| 6 | reserved |
| 7 | reserved |
| 8 | reserved |
| 9 | reserved |
| 10 | reserved |
| 11 | reserved |
| 12 | reserved |
| 13 | reserved |
| 14 | reserved |
| 15 | reserved |

**ACR**

| Bit | | Bit | |
|---|---|---|---|
| 0 | RIE | 4 | HIPRO ENB |
| 1 | TIE | 5 | reserved |
| 2 | EXT TRG 1 | 6 | reserved |
| 3 | EXT TRG 2 | 7 | reserved |

**LED STATUS**

| | |
|---|---|
| G | INSERT |
| G | CARRIER DETECT |

**CSR 1** (READ RESET)

| Bit | |
|---|---|
| 0 | TX FIFO FULL |
| 1 | TX FIFO NOT EMPTY |
| 2 | TX FIFO 7/8 FULL |
| 3 | always 0 |
| 4 | INT FIFO FULL |
| 5 | PROTOCOL VIOLATION |
| 6 | CARRIER DETECT FAIL |
| 7 | BAD MESSAGE |
| 8 | RX OVERFLOW |
| 9 | TX RETRY |
| 10 | TX RETRY TIME-OUT |
| 11 | REDUND TXRX FAULT |
| 12 | GP CTR/TIMER OVRFLO |
| 13 | CURRENT LINK FOR USE |
| 14 | INTERRUPTS ARMED* |
| 15 | FO BYPASS NOT CNCTD |

**CSR 3**

| Bit | |
|---|---|
| 0 | NN0 |
| 1 | |
| 2 | NUMBER |
| 3 | OF |
| 4 | NODES |
| 5 | |
| 6 | |
| 7 | NN7 |
| 8 | TXID0 |
| 9 | |
| 10 | |
| 11 | NODE ID |
| 12 | |
| 13 | |
| 14 | |
| 15 | TXID7 |

**CSR 5**

| Bit | |
|---|---|
| 0 | RFA16 |
| 1 | |
| 2 | RX FIFO |
| 3 | ADDRESS |
| 4 | (MSW) |
| 5 | |
| 6 | RFA22 |
| 7 | reserved |
| 8 | reserved |
| 9 | reserved |
| 10 | reserved |
| 11 | reserved |
| 12 | reserved |
| 13 | reserved |
| 14 | RF RETRY |
| 15 | INT FIFO NOT EMT |

**CSR 7**

| Bit | |
|---|---|
| 0 | reserved |
| 1 | reserved |
| 2 | reserved |
| 3 | reserved |
| 4 | reserved |
| 5 | reserved |
| 6 | reserved |
| 7 | reserved |
| 8 | reserved |
| 9 | reserved |
| 10 | reserved |
| 11 | reserved |
| 12 | reserved |
| 13 | reserved |
| 14 | reserved |
| 15 | reserved |

* Write to CSR1 to re-arm interrupts.

| Bit | CSR 8 |
|---|---|
| 0 | AGE & RXID MUX |
| 1 | HOLDOFF DISABLE |
| 2 | CHP SELECT EEPROM |
| 3 | AUX MICROWIRE |
| 4 | MICROWIRE DOUT |
| 5 | EEPROM PROG ENABLE |
| 6 | MICROWIRE CLOCK LN |
| 7 | MICROWIRE DOUT DIN |
| 8 | INIT ASIC/CSR RESET |
| 9 | GP CTR FREE |
| 10 | RX INT OVERRIDE |
| 11 | MECH SW OVR |
| 12 | MEM SIZE |
| 13 | MEM SIZE |
| 14 | MEM SIZE |
| 15 | Reserved |

| Bit | CSR 10 |
|---|---|
| 0 | WRITE POST ENB |
| 1 | reserved |
| 2 | reserved |
| 3 | reserved |
| 4 | reserved |
| 5 | reserved |
| 6 | reserved |
| 7 | reserved |
| 8 | reserved |
| 9 | reserved |
| 10 | reserved |
| 11 | reserved |
| 12 | reserved |
| 13 | reserved |
| 14 | reserved |
| 15 | reserved |

| Bit | CSR 12 |
|---|---|
| 0 | VIRT PG ENB |
| 1 | always 0 |
| 2 | always 0 |
| 3 | always 0 |
| 4 | always 0 |
| 5 | VPA 12 |
| 6 | |
| 7 | |
| 8 | VIRTUAL |
| 9 | PAGE |
| 10 | NUMBER |
| 11 | |
| 12 | |
| 13 | |
| 14 | |
| 15 | VPA 22 |

| Bit | CSR 14 |
|---|---|
| 0 | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| | reserved |
| 15 | reserved |

| Bit | CSR 9 |
|---|---|
| 0 | TX FIFO FULL MASK |
| 1 | TX FIFO NOT EMP MASK |
| 2 | TX FIFO 7/8 FULL MASK |
| 3 | BIST STREAM (R/O) |
| 4 | RX FIFO FULL MASK |
| 5 | PROTOCOL VIOL MASK |
| 6 | CARRIER DETECT FAIL MASK |
| 7 | BAD MESSAGE MASK |
| 8 | RX OVERFLOW MASK |
| 9 | TX RETRY MASK |
| 10 | TX RETRY TIME-OUT |
| 11 | REDUN TXRX FAULT MASK |
| 12 | GP CTR/TIMER OVRFLO |
| 13 | UTIL CTR MODES |
| 14 | UTIL CTR MODES |
| 15 | FO BYPASS NOT CNCTD MASK |

| Bit | CSR 11 |
|---|---|
| 0 | reserved |
| 1 | reserved |
| 2 | reserved |
| 3 | reserved |
| 4 | reserved |
| 5 | reserved |
| 6 | reserved |
| 7 | reserved |
| 8 | reserved |
| 9 | reserved |
| 10 | reserved |
| 11 | reserved |
| 12 | reserved |
| 13 | reserved |
| 14 | reserved |
| 15 | reserved |

| Bit | CSR 13 |
|---|---|
| 0 | RD COUNT 0 |
| 1 | |
| 2 | |
| 3 | |
| 4 | GENERAL |
| 5 | PURPOSE |
| 6 | COUNTER/ |
| 7 | TIMER |
| 8 | REGISTER |
| 9 | |
| 10 | |
| 11 | |
| 12 | |
| 13 | |
| 14 | |
| 15 | RD COUNT 15 |

| Bit | CSR 15 |
|---|---|
| 0 | reserved |
| 1 | reserved |
| 2 | reserved |
| 3 | reserved |
| 4 | reserved |
| 5 | reserved |
| 6 | reserved |
| 7 | reserved |
| 8 | reserved |
| 9 | reserved |
| 10 | reserved |
| 11 | reserved |
| 12 | reserved |
| 13 | reserved |
| 14 | reserved |
| 15 | reserved |

## SCRAMNet+ NETWORK CONFIGURATION DATA SHEET

| NODE ID | HOST MACHINE | MEMORY ADDRESS & BUS | MEMORY SIZE | CSR ADDRESS & BUS | INT LEVEL | SCRAMNet SERIAL # |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

# GLOSSARY

**A16** ------------------------------------A type of module that provides or decodes an address on address lines A01 through A15.

**A24** ------------------------------------A type of module that provides or decodes an address on address lines A01 through A23.

**A32** ------------------------------------A type of module that provides or decodes and address on address lines A01 through A31.

**address-only cycle** -----------------A DTB cycle that consists of an address broadcast, but no data transfer. The slave does not acknowledge address-only cycles and the master terminates the cycle without waiting for an acknowledgment.

**alarm**---------------------------------Manually resetable latched error condition.

**ACR** -----------------------------------**auxiliary control RAM**. A memory buffer typically used as a data bus width extension for control purposes only. Also referred to as shadow memory .

**bad message**------------------------A message error condition reported by a node's receiver circuitry. This condition is automatically corrected by SCRAMNet+ hardware.

**block read cycle** --------------------A DTB cycle used to transfer a block of 1 to 256 bytes from a slave to a master. This transfer is done using a string of 1-, 2-, or 4-byte data transfers. Once the block transfer is started, the master does not release the DTB until all of the bytes have been transferred. It differs from a string of read cycles in that the master broadcasts only one address and address modifier (at the beginning of the cycle.) The slave increments this address on each transfer so that the data for the next cycle is retrieved from the next higher location.

**block write cycle** -------------------A DTB cycle used to transfer a block of 1 to 256 bytes from a master to a slave. The block write cycle is very similar to the block read cycle. It uses a string of 1-, 2-, or 4-byte data transfers and the master does not release the DTB until all of the bytes have been transferred. It differs from a string of write cycles in that the master broadcasts only one address and address modifier (at the beginning of the cycle). Then the slave increments this address on each transfer so that the next transfer is stored in the next higher location.

**board**---------------------------------A printed circuit board (pcb), its collection of electronic components, and either one or two 96-pin connectors that can be plugged into the backplane connectors.

**BURST** ------------------------------A protocol where messages are transmitted without error correction to gain higher throughput.

**BURST+** ----------------------------Also BURST PLUS. A variable-length message packet size enhancement for the burst protocol. Maximum packet size may be set to 256 bytes or 1024 bytes, plus a 46-bit header.

**bus timer**----------------------------A functional module that measures the time each data transfer takes on the DTB and terminates the DTB cycle if a transfer takes too long. Without this module, it could wait forever for a slave to respond if the master tries to transfer data to or from a nonexistent slave location. The bus timer prevents this by terminating the cycle.

**carrier loss**--------------------------A hardware failure reported when the incoming light link has failed because it is too weak or nonexistent in one or both fibers from the preceding node.

**CSR**----------------------------------**Control/status register**. These registers are used for configuration and control and provide status values that can be interrogated. These registers are located in the computer's address space. Power-up register values are contained in the EEPROM.

**data filter** ---------------------------A process of comparing a host WRITE to shared memory with contents of the specified memory location to eliminate transmission of redundant data and reduce network traffic.

**deterministic** -----------------------Completely predictable message transit time from application to application.

**data transfer bus**------------------One of the four buses provided by the backplane. The data transfer bus allows masters to direct the transfer of binary data between themselves and slaves (data transfer bus is often abbreviated DTB).

**data-transfer-bus cycle**----------A sequence of level transitions on the signal lines of the DTB that result in the transfer of an address or an address and data between a master and a slave. There are 34 types of data transfer bus cycles.

**direct memory access**------------(DMA) transfer.  An I/O transfer conducted by a device controller which accesses memory directly and, as a result, can transfer a large volume of data without requesting a processor interrupt after each unit amount. Contrast with programmed I/O (PIO) transfer.

**device interrupt** -------------------An interrupt received on interrupt priority levels 20-23. Device interrupts can be requested only by devices, controllers, and memories.

**DTB** ----------------------------------A mnemonic for data transfer bus.

**edges** ---------------------------------Transitions that appear on a signal line.

**EEPROM**---------------------------The EEPROM stores the initial power-up register values. The EEPROM can be programmed either over the backplane or by most PROM programmers. An EEPROM Initialization (EPI) Program is included in the Systran software utilities.

**falling edge** ------------------------The time during which a signal makes its transition from high to low.

**FIFO** ---------------------------------A data storage method; First In First Out. Also refers to the specific storage area; Transmit FIFO, Interrupt FIFO, etc.

**foreign message** -------------------A message that is in (passing through) a node other than the one of origin.

**functional module** ----------------A collection of electronic circuitry that resides on one board and works together to accomplish a task.

**halfword**-----------------------------Any double byte on even 16 bit boundaries.

**insert a node** -----------------------The act of placing a node on a network for the purpose of transmitting and receiving messages.

**interrupt** ----------------------------An event that changes the normal flow of instruction execution other than an exception or a branch, jump, case or call instruction.

**interrupt acknowledge cycle**---A DTB cycle, initiated by an interrupt handler, that reads a status/ID from an interrupter. An interrupt handler generates this cycle when it detects an interrupt request from an interrupter and it has control of the DTB.

**interrupter**--------------------------A functional module that generates an interrupt request on the priority interrupt bus and then provides states/ID information when the interrupt handler requests it.

**interrupt handler** -----------------A functional module that detects interrupt requests generated by interrupters and responds to those requests by asking for status/ID information.

**ISR**------------------------------------**interrupt service routine**. A routine executed when a device interrupt occurs.

**I/O space** ----------------------------The regions of host processor physical address space that contain the configuration registers, device control, states registers and data registers. These regions are physically noncontiguous.

**latched** -------------------------------Data is electrically stored in a circuit until it is needed. A method of coordinating two synchronous events.

**locking a page in memory** ------Making a page ineligible for either paging or swapping. A page stays locked in physical memory until the operating system specifically unlocks it.

**longword** ----------------------------Four bytes (32 bits) of data.

**loopback**-----------------------------A method of transmitting to the same node's receivers for testing purposes. Applies to both fiber optic and wire media. Also, a test that loops the outgoing signal back to its source.

**master**--------------------------------A functional module that initiates DTB cycles to transfer data between itself and a slave module.

**message packet** --------------------See packet.

**native message**---------------------A message that is received by the node of origin.

**node latency** ------------------------The time delay at a node before a foreign message can be retransmitted.

**packet**--------------------------------A message that travels on the network. The minimum packet consists of 81 bits and 1 start bit. The packet includes five fields: Source ID (8 bits), Age (8 bits), Control (3 bits), Data Address (21 bits), Data (32 bits), and 9 parity bits; one for every 8 bits.

**physical address**-------------------The address used by hardware to identify a location in physical memory or on directly-addressable secondary storage devices (such as disks). A physical memory address consists of a page-frame number and the number of a byte within the page.

**PLATINUM**------------------------A protocol where messages are transmitted as fast as the system will allow with error correction enabled.

**PLATINUM+**----------------------Also platinum plus. A variable-length message packet size enhancement for the platinum protocol. Maximum packet size may be set to 256 bytes or 1024 bytes, plus a 46-bit header.

**priority interrupt bus** -----------One of the four buses provided by the backplane. The priority interrupt bus allows interrupter modules to send interrupt requests to interrupt handler modules, and interrupt handler modules to acknowledge these interrupt requests.

**(PIO)** ---------------------------------programmed I/O transfer. An I/O transfer, primarily conducted by a driver program, that requires processor intervention after each byte or word is transferred. Contrast with Direct Memory Access (DMA) transfer.

**protocol violation** -----------------A signal error at the physical layer (fiber or coax) resulting from noise on the transmission lines or a result of hardware failure. This violation can be any one of the following:

- Missing transition for two clock periods on either line
- Parity error
- Framing error

**read cycle** ---------------------------A DTB cycle used to transfer 1-, 2-, 3-, or 4-bytes from a slave to a master. The cycle begins when the master broadcasts an address and an address modifier. Each slave captures this address and address modifier, and checks to see if it is to respond to the cycle. If so, it retrieves the data from its internal storage, places it on the data bus, and acknowledges the transfer. Then the master terminates the cycle.

**read-modify-write cycle**---------A DTB cycle that is used to both read from, and write to, a slave's byte location(s) without permitting any other master to access that location during that cycle. This cycle is most useful in multi-processing systems where certain memory locations are used to control access to certain systems resources, for example, semaphore locations.

**requester** ----------------------------A functional module that resides on the same board as a master or interrupt handler and requests use of the CTB whenever its master or interrupt handler needs it.

**retry** ----------------------------------A hardware failure condition reported when the first attempt to send a message around the network has resulted in some type of bit error. The message will be retransmitted indefinitely by the originating node until it is received correctly by the originating node. Valid only in error correction mode (PLATINUM.)

**retry time-out**----------------------A hardware failure condition reported when the first attempt to send a message around the network is not received by the originating node within the time out period specified in CSR5. The message will be retransmitted indefinitely by the originating node until it is received correctly by the originating node. Valid only in error correction mode (PLATINUM.)

**rising edge** --------------------------The time during which a signal makes its transition from low to high.

**Rx** -------------------------------------Abbreviation for receive or receiver.

**SCSI** ----------------------------------Refers to the American National Standard for Information Systems Small Computer System Interface - 1 (X3.131-1986) or the ANSI Small Computer System Interface - 2 (X3.131-1989). This standard defines mechanical, electrical and functional requirements for attaching small computers to each other and to intelligent peripheral devices.

**shadow memory**-------------------See Auxiliary Control RAM (ACR)

**shared memory (SM)** ------------SCRAMNet memory physically located on the network board. This dual-ported memory is accessible by the host and the network. A host write to shared memory results in a transmitted write to all SCRAMNet nodes at the same relative location.

**shortword**---------------------------16 bits. Also referred to as halfword.

**signal mnemonics**-----------------Terms used to identify signal line events. (1) An asterisk following the name of signals that are level-significant denotes the signal is true/valid when the signal is low. (2) A asterisk following the name of signals that are edge-significant denotes the actions initiated by that signal occur on the falling edge.

**slave** ----------------------------------A functional module that detects DTB cycles initiated by a master and, when those cycles specify its participation, transfers data between itself and the master.

**slot** ------------------------------------A position where a board can be inserted into a backplane. If the system has both a J1 and a J2 backplane (or a combination J1/J2 backplane) each slot provides a pair of 96-pin connectors. If the system has only a J1 backplane, then each slot provides a single 96-pin connector. Also, another name for message packet.

**system clock driver**---------------A functional module that provides a 16 MHz timing signal on the utility bus.

**time-out** -----------------------------Also network time-out. The time written to CSR5 that must elapse before a native message will be retransmitted. The time-out must be a non-zero value.

**Tx** -------------------------------------Abbreviation for transmit or transmitter.

**UAT** ----------------------------------A master that sends or receives data in an unaligned fashion.

**utility bus** ---------------------------One of the four buses provided by the backplane. This bus includes signals that provide periodic timing and coordinate the power-up and power-down of sequence of the system.

**write cycle** --------------------------A DTB cycle used to transfer 1-, 2-, 3-, or 4-bytes from a master to a slave. The cycle begins when the master broadcasts an address and address modifier and places data on the DTB. Each slave captures this address and address modifier, and checks to see if it is to respond to the cycle. If so, it stores the data and then acknowledges the transfer. The master then terminates the cycle.

*This page intentionally left blank*

# INDEX