SCRAMNet®+ Network

VME3U
Hardware Reference

Document No. D-T-MR-VME3U###-A-0-A2

FOREWORD

The information in this document has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. SYSTRAN reserves the right to make changes without notice.

SYSTRAN makes no warranty of any kind with regard to this printed material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

SCRAMNet® is a registered trademark of SYSTRAN Corporation.
VAX™ and VMS™ are trademarks of DIGITAL Equipment Corporation.
UNIX® is a registered trademark of Novell.
ST® is a registered trademark of AT&T.

Revised: November 15, 1995

SYSTRAN Corporation
4126 Linden Avenue
Dayton, OH 45432-3068
(513) 252-5601

TABLE OF CONTENTS

APPENDICES

FIGURES

TABLES

1. HOW TO USE THIS MANUAL

1.1 Scope

This document is a reference manual for the **SCRAMNet+** VME3U host interface board.

This document provides a physical and functional description of the **SCRAMNet+** VME3U board designed for a VME based host system.

This information is intended for systems designers, engineers and network installation personnel.

The reader should have at least a systems level understanding of general computer processing, of memory and hardware operation, and of the specific host processor.

1.2 Organization

This document is divided into six sections: Introduction, Description, Installation, Operation, Control/Status Register Description, and Physical Features.

Introduction:	An overview of the **SCRAMNet+** VME3U product and host interface compatibility.
Description:	A functional description of the **SCRAMNet+** network node.
Installation:	Procedures for unpacking, configuring and installing the **SCRAMNet+** VME3U node board.
Operation:	A discussion of how the node works; including Control/Status registers, Shared Memory, Message Passing, Data Filtering, Interrupt Initialization and Handling, Auxiliary Control RAM, and other features.
CSR Description:	A detailed explanation of all the Control/Status Registers.
Physical Features:	A description of the physical features of the **SCRAMNet+** VME3U board.

1.3 Appendices

A - CSR Summary:	A quick reference summary of the Control/Status Registers by bit, function and name.
B - Cabinet Kit:	A discussion of how the board is connected to the cabinet kit.
C - Specification Summary:	General board specifications, Bus Voltage Specification, part number breakdown, and Fiber Optic Bypass Switch specifications.

D - Host Access Timing:	A discussion of timing coordination between the board and host.
E - Configuration Aids:	Control/Status Register (CSR) Reference Sheet and network configuration form.
F - Glossary:	A glossary of words, phrases and terms used in the reference manual.

1.4 Related Documentation

SCRAMNet Network Utilities User Manual (Doc. Nr. C-T-MU-UTIL####-A-0-A1) - A user's manual for the **SCRAMNet Classic**, **SCRAMNet+**, and **SCRAMNet+** hardware diagnostic software, **SCRAMNet+** EEPROM initialization software, and the **SCRAMNet** Network Monitor.

SCRAMNet Network Programmer's Reference Guide (Doc. Nr. D-T-MR-PROGREF#-A-0-A2) - A collection of routines to assist SCRAMNet users with application development.

2. 0 INTRODUCTION

2.1 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+** VME3U host interface node board is backwards-compatible with the original **SCRAMNet Classic** product with the exception of the GOLD Ring communication protocol. The programmable byte-swapper is no longer available on VMEbus-based products.

The **SCRAMNet+** VME3U board requires a single slot in the VMEbus chassis.

The **SCRAMNet+** VME3U board base address for Control/Status Registers (CSRs) is switch selectable. The 128 KB on-board 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 128 KB memory.

2.1.1 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 communicated to the other network 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 length message 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.

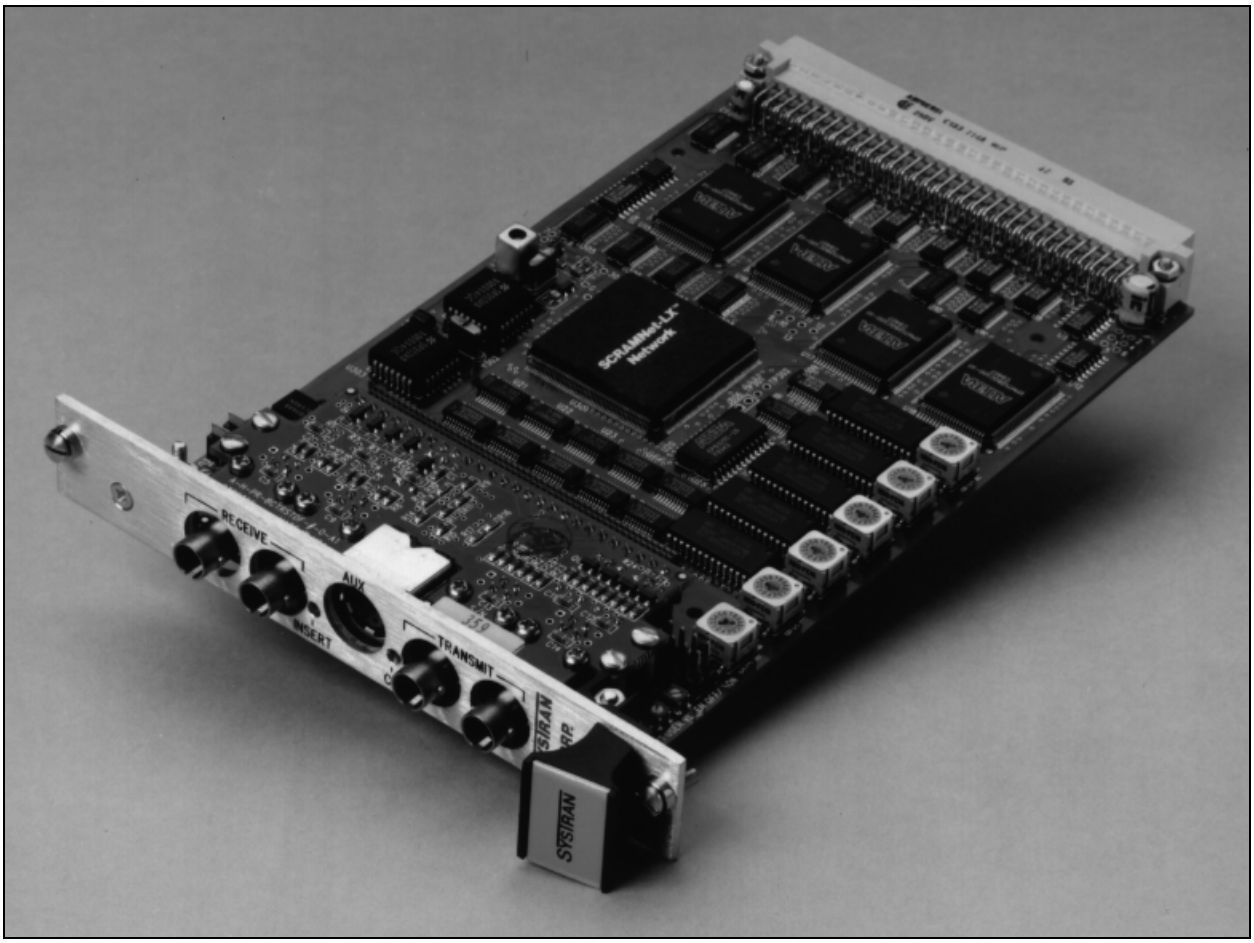


Figure 2-1 VME3U Board

2.1.2 Options

Optional paired fiber optic or coax 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, and functions as a repeater.

2.1.3 VME3U Board Features

Mezzanine board memory upgrade option
General Purpose Counter
Error Interrupt Mask
Dynamic shared memory addressing
Switch selectable CSR Address Selector
Virtual paging for Shared Memory (CSR selectable)
Variable length message packet capability
Dual port memory
Dual vector memory/error interrupt (Single Level Interrupt)
Single Slot Solution
EEPROM initialization

2.2 VMEbus Specification Level

The SCRAMNet+ VME3U host board was designed in accordance with the VMEbus specification Revision C.3, ANSI/IEEE Std 1014-1987.

Slave device
SADO24 (No UAT, no BLT)
SRMW16 (D16, D08 (EO))
8-bit vector ROAK
3U single height card size

2.3 Addressing Compatibility

2.3.1 Memory

The shared memory resident on the **SCRAMNet+** VME3U host interface board must be located on the A24 standard bus. The memory address selected must be an address boundary that is a multiple of the shared-memory size, and must be loaded and enabled through the CSRs.

2.3.2 Control/Status Registers

2.3.3 I/O

Control in the form of CSRs can be located on either the A16 bus or the A24 bus. The address is set manually using the rotary switches on the board. This is independent of the shared memory address. The CSRs require 64 contiguous bytes of address space.

2.4 Data Transfer Capability

2.4.1 Memory

Data transfers to the shared memory on the **SCRAMNet+** VME3U host interface can be 8- or 16-bits wide and may be of the read-modify-write type. If the host needs to pass 32-bit data to shared memory, it must be accomplished in two 16-bit segments. The host system does this automatically; transparent to the user. Three-byte unaligned transfers are not permitted.

2.4.2 I/O

Data transfers to the I/O control area of the **SCRAMNet+** VME3U host interface can be 8- or 16- bits wide and may be of the read-modify-write type. If the host needs to pass 32-bit data to the Control/Status Registers, it must be accomplished in two 16-bit segments. The host system does this automatically; transparent to the user. Three-byte unaligned transfers are not permitted.

2.5 Interrupt Capability

The **SCRAMNet+** VME3U host interface is an interrupter of the type D08. Interrupt level 1 through 7 may be selected by setting a value in CSR15. The IRQ is mapped using a bit-wise format; bit 1 is IRQ 1, bit 2 is IRQ 2, etc. (Bit 0 is not used.) The vector generated by the node is 8 bits wide. Being an ROAK (Release On interrupt AcKnowledge) type of interrupter means that the device releases the interrupt request

during the interrupt acknowledge cycle. The 8-bit vector address is loaded at CSR6 - Memory, and CSR7 - Error. If Interrupt-on-Error is not used, CSR7 must contain the same vector as CSR6.

2.6 P1 Connector

The **SCRAMNet+** VME3U card's P1 backplane connector is in accordance with the VMEbus specifications.

2.7 Utility Software

2.7.1 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.

2.7.2 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.

2.7.3 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. 0 DESCRIPTION

3.1 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 which is updated over a high-speed, serial-ring network. It 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.

3.2 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.

The **SCRAMNet+** memory size can range from 128 KB on-board memory to 8 MB of expansion memory. Available options include: 128 KB, 512 KB, 1 MB, 2 MB, 4 MB and 8 MB. No software driver is required except for interrupt handling. When a host computer writes to the shared memory, the proper handshaking for a memory card is supplied by the **SCRAMNet+** node host adapter. The shared memory behaves somewhat like resident or local memory.

3.2.1 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.

3.2.2 Control/Status Registers (CSRs)

The operation of the **SCRAMNet+** board is controlled by 17 Input/Output (I/O) CSRs. The location of the CSRs in the computer's address space is switch selectable. In most

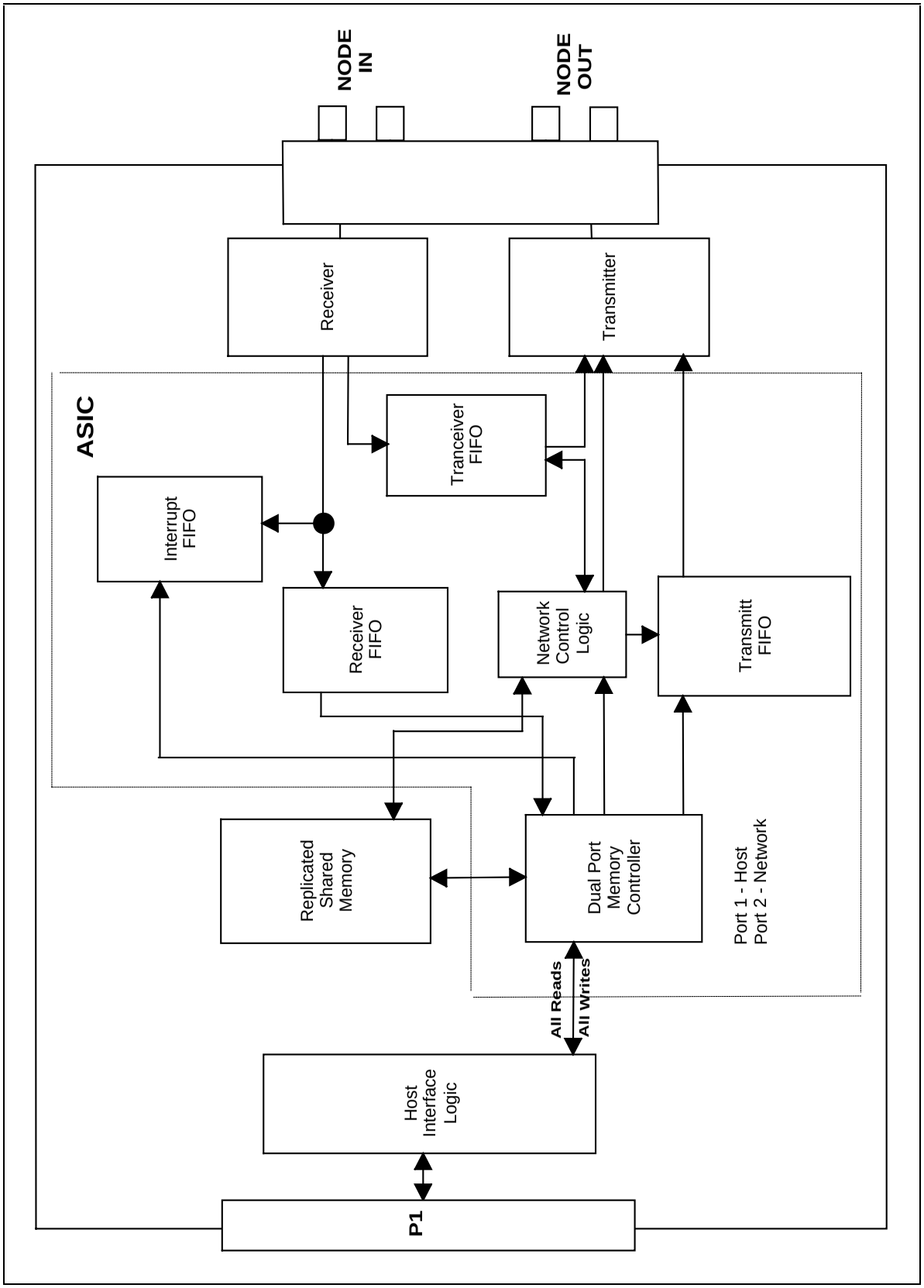


Figure 3-1 Functional Diagram

cases, the mode of operation is set during initialization and remains unchanged during run time. The CSRs are described in detail in Section 5.

3.2.3 Virtual Paging

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 (e.g. 2 MB can page to 0, 2, 4 or 6 MB address).

3.3 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 3-1.

3.3.1 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.

The Transmit FIFO has a 7/8 Full condition indicator (CSR1, bit 2 ON). When the Transmit FIFO reaches a FULL condition (CSR1, bit 0 ON), one more host WRITE could cause a message to be lost. To prevent this, the CSR-controllable, built-in **SCRAMNet+** feature called VME Holdoff extends the computer WRITE cycle until the Transmit FIFO is able to empty at least one message.

3.3.2 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.

3.3.3 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.

3.3.4 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.

3.4 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 slot size, the network can accommodate over 1,800,000 message slots passing by each node every second. There is an approximate 247 ns (minimum) delay at each node as the message slot works its way around the ring. The maximum delay depends on the selection of fixed or variable length message packets. A fixed message packet has a maximum delay of 800 ns, a 256-byte variable packet is 16 µs, and a 1024-byte variable 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.

3.4.1 Protocol

The protocol is a register-insertion methodology and is NOT 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 32-bit messages per node on the ring at a time. The limited 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 detection enabled. The messages are transmitted as fast as the system will allow, but error checking is used to detect and re-transmit corrupted 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 50-bit header plus the data. The user-selectable maximum packet size increases the data size from the normal 32 bits to either 256 or 1024 bytes of data. Data must be written sequentially.

3.5 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.

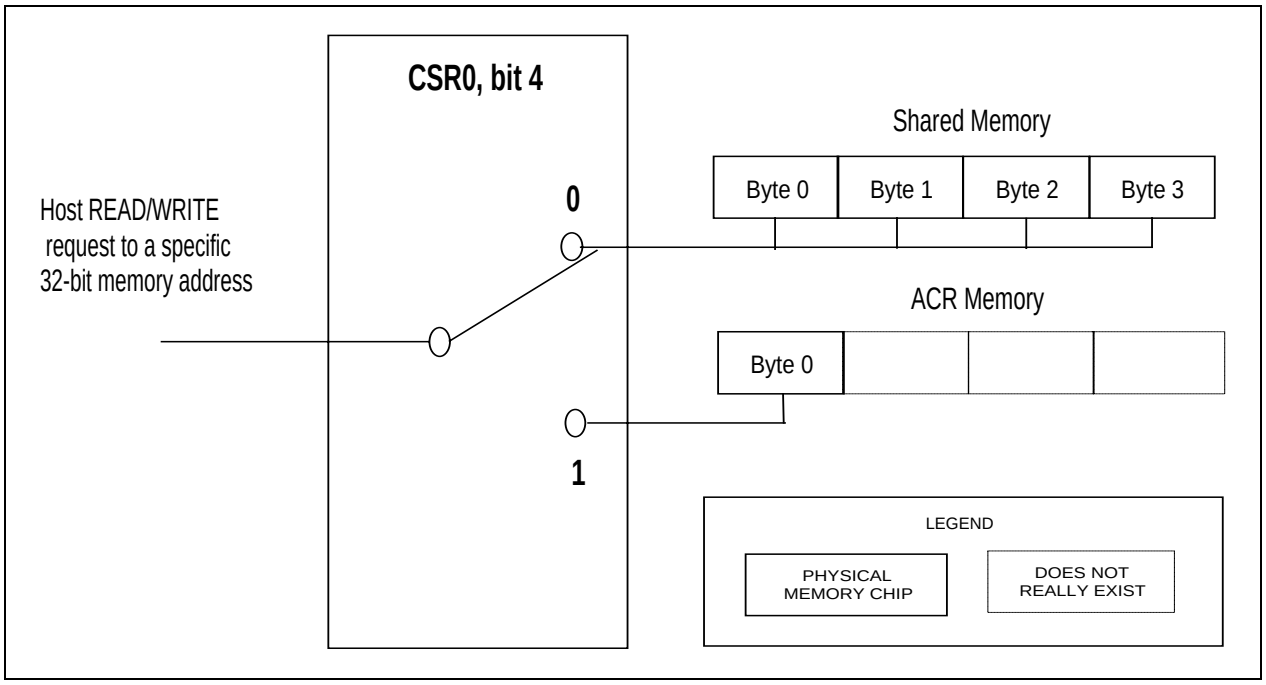


Figure 3-2 ACR/Memory Access

In Figure 3-2, 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.

3.6 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. There must also be a host-dependent interrupt vector placed in CSR6 and CSR7 identifying the Interrupt Service Routine (ISR). The driver is required primarily to permit the host processor to handle interrupts from the **SCRAMNet** device.

3.6.1 Network Interrupt WRITEs

FOREIGN MESSAGE

The node can receive a message from another node with the interrupt bit set. If Receive Interrupt Enable (ACR, bit 0) and Interrupt Mask Match Enable (CSR0, bit 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, bit 9) and Enable Interrupt on Own Slot (CSR2, bit 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" .

3.6.2 Selected

The "selected" 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 3.5.

OUTGOING INTERRUPT

The Outgoing Interrupt is described in Figure 3-3. If both Transmit Interrupt Enable (ACR, bit 1) and Network Interrupt Enable (CSR0, bit 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 3-4 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, bit 0), Host Interrupt Enable (CSR0, bit 3), and the Interrupt Memory Mask Match Enable (CSR, bit 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.

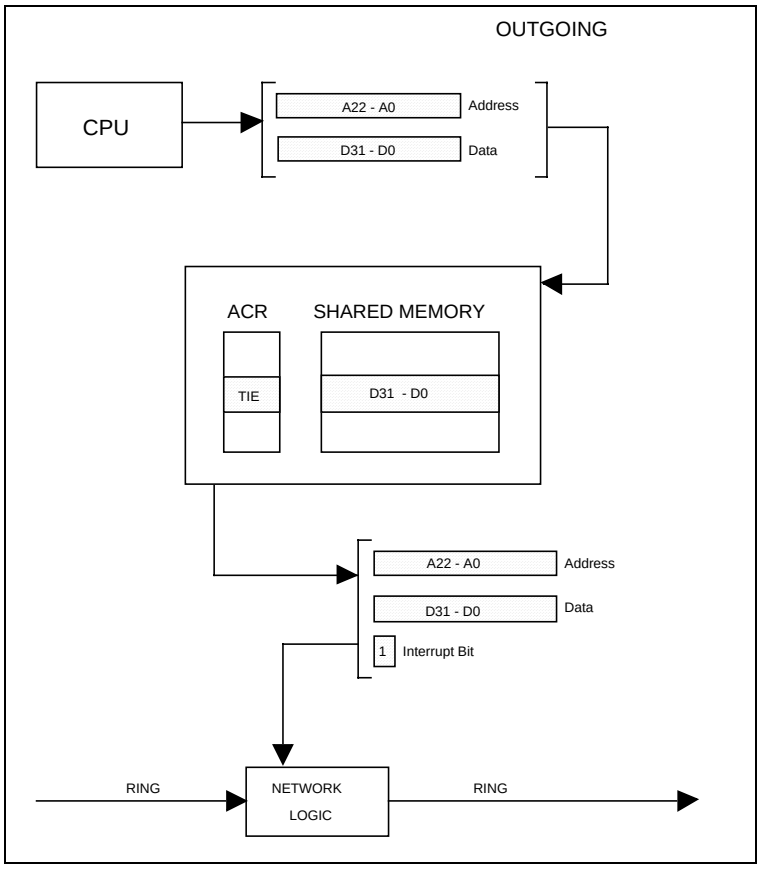


Figure 3-3 Outgoing Interrupt

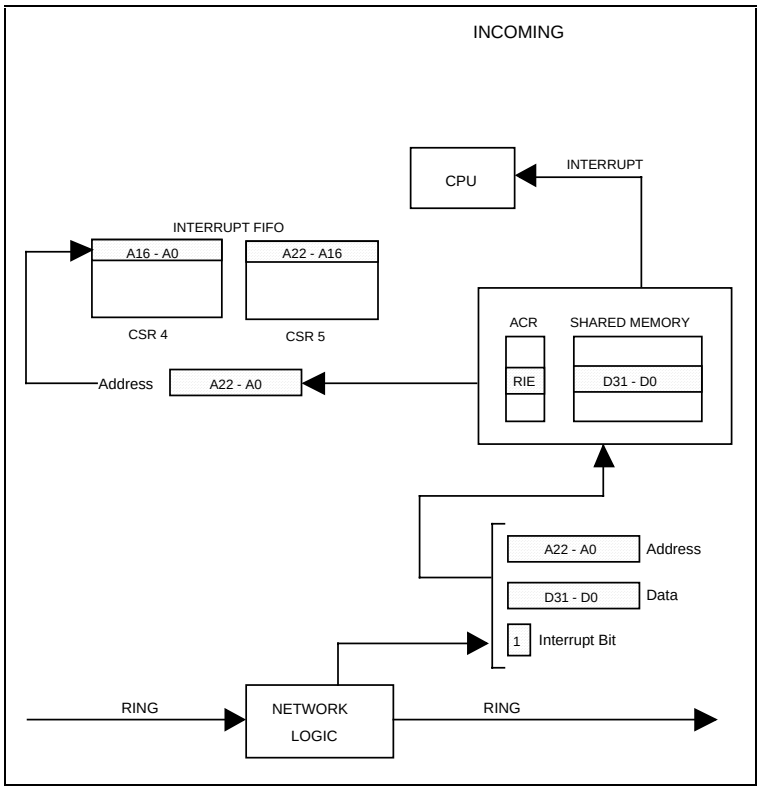


Figure 3-4 Incoming Interrupt

NETWORK ERRORS

The Interrupt on (Network) Errors mode is enabled by setting CSR0, bit 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.

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.

A Network Interrupt vector may be placed in CSR7 to identify a Network Error Interrupt Service Routine.

3.6.3 Forced

The forced interrupt method works the same as for selected 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, bits 6 and/or 9.

When Override Receive Interrupt Enable (CSR0, bit 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, bit 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, bit 10), is used to designate all incoming network traffic as interrupt messages. The network message interrupt bit does not need to be set.

3.7 External Triggers

Two external triggers are supported by **SCRAMNet+** VME6U. 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 bit 2 enables)
Trigger 2 - Network Write (ACR bit 3 enables)

3.8 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 3.7: External Triggers). A specific shared-memory location may be identified with External Trigger 2 (ACR, bit 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.

3.9 LED Status Indicators

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).

3.10 Modes of Operation

3.10.1 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.

3.10.2 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.

HIPRO READ

The HIPRO READ is controlled by CSR16. This register is CSR enabled and ACR location selectable.

To conserve host cycles and increase host throughput, HIPRO READ mode allows the host to get part of the information (1 shortword or 2 bytes) during the first READ on that longword boundary. On the next READ operation (not the same location or within the same longword boundary) the remaining data is provided.

3.10.3 VME 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 VME 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, bit 1 ON. Transmit FIFO 7/8 Full (CSR1, bit 2) can then be used to control the data flow via software control.

3.10.4 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, bit 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, bit 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, bit 6) set to OFF (power up default), and Insert Node (CSR0, bit 15) enabled to be valid. When 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, bit 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.

3.10.5 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.

3.11 Options

3.11.1 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 nano-second 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.

3.11.2 Quad Switch

The Quad Switch is a switching center and is used to dynamically configure active **SCRAMNet** Classic 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. 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 5 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 5 ports have standard **SCRAMNet** transmitters and receivers. Each port can transmit data to and receive data from the internal ring.

The Quad Switch may be controlled by a host processor via the RS-232 serial interface.

Multiple Quad Switches may be daisy-chained via the RS 422/485 serial interface port.

3.11.3 Cabinet Kit

The Cabinet Kit for the **SCRAMNet+** Network permits adapting the node to the host cabinet while still maintaining the shielding of the chassis. Access to the node's connections, and cable integrity is exterior to the cabinet. This eliminates the need to remove the node board, once proper installation has been accomplished.
Cabinet kits are available for a variety of vendor chasis, and include the Compact and the Extended models. The Compact Cabinet Kit is described in Appendix B.

4. 0 OPERATION

4.1 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 4-11, page 4-28 at the end of this section.

4.2 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.

4.2.1 Virtual Paging

CSR12 is the virtual-paging register. Set CSR12, bit 0 to ‘1’ to enable virtual paging.

All **SCRAMNet+** nodes use the same 8 MB shared-memory map. Virtual paging allows a node with less than 8 MB shared memory to move their memory window throughout the **SCRAMNet+** physical 8 MB 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 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 4 MB. Any WRITE accesses to the lower 4 MB will be ignored since there is no memory addressed there.

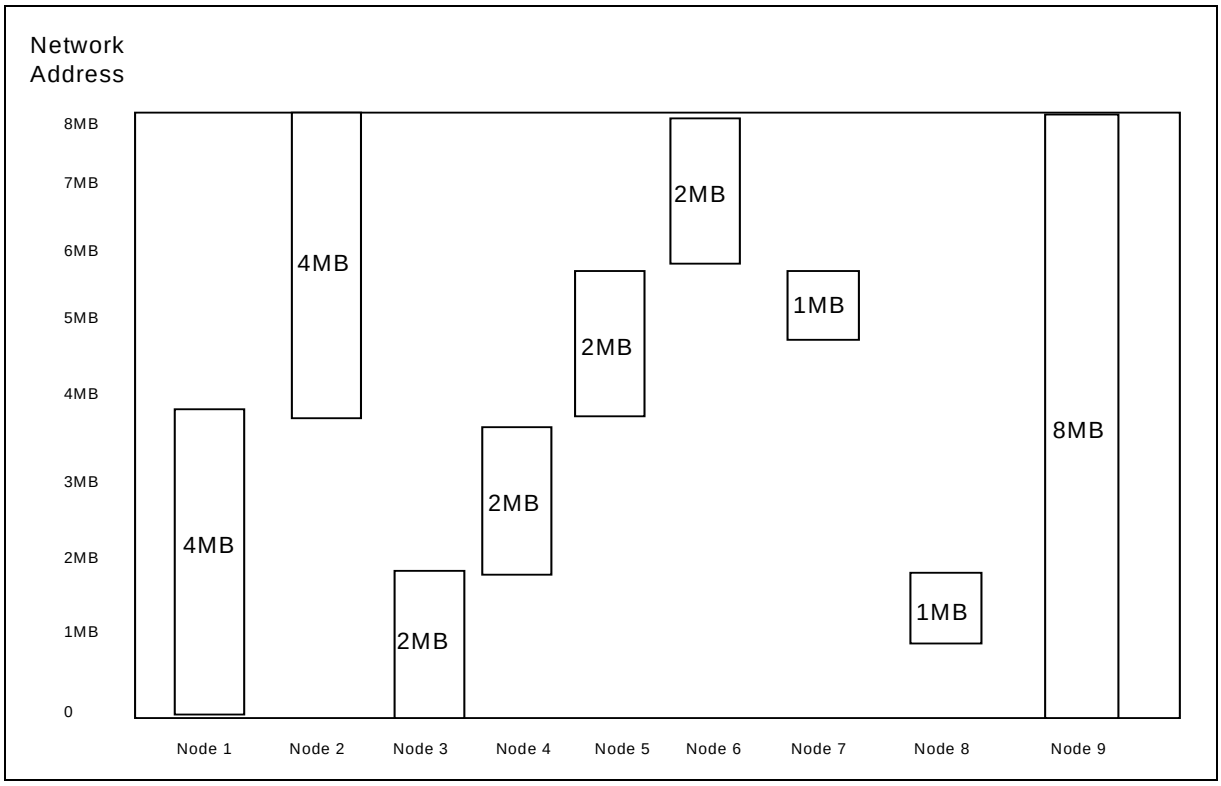


Figure 4-1 Memory Sharing With Virtual Paging

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 4-1 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.

4.2.2 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 which 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 use unused portions of **SCRAMNet+** memory to store other program segments, the **SCRAMNet+** memory common blocks should be declared 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.

4.2.3 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. Section 5.0 CSR DESCRIPTIONS discusses the definition and use of each bit in the CSRs. Appendix A contains a list of the CSRs and a brief identification of each bit.

4.3 Initialization

NOTE: The EEPROM initializes the RX_ID and AGE fields to 0. Therefore, upon power-up, READ the Node ID (TX_ID) CSR3, bits 15-8 and WRITE it back. This will set the RX_ID field equal to the TX_ID field. If this is not done, native messages will not be recognized and will never be removed from the network, and the time-out will not go into effect. A routine to perform this task is included with the Initialization Software package.

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

Table 4-1 EEPROM Table

	0	**2**	**4**	**6**	**8**	**A**	**C**	**E**
00	0000	0000	8000	0001	0000	0010	0000	0000
10	0800	0FF0	0000	0000	0000	0000	0000	0020
...	0000	0000	0000	0000	0000	0000	0000	0000
70	5555	5555	5555	5555	5555	5555	5555	5555

If you want to communicate only to the local host's **SCRAMNet+** memory, no fiber optic cable connections are required. 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 Program.

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

4.4 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.

4.4.1 Message Contents

The smallest **SCRAMNet+** Network message consists of 82 bits, including 4 bytes of data. When the PLUS mode is enabled, data size may vary in size up to 256 bytes or 1024 bytes depending on the option selected. This basic message format contains five fields: Source ID, Age, Control, Data Address, and Data Value. The message can be described as follows:

Table 4-2 SCRAMNet+ Message Contents

START	ID	AGE	CONTROL	DATA ADDRESS	DATA VALUE
1	8+P	8+P	1 1 1 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 - Not used in BURST mode.

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 of the word in **SCRAMNet+** memory that is currently being updated around the ring.

4.4.2 Protocol

BURST MODE

BURST Mode is the normal protocol for **SCRAMNet+**. The BURST mode is enabled by setting CSR2, bit 12 off, and bits 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 packet length for data transfer.

PLATINUM MODE

The PLATINUM mode is BURST mode with error detection enabled. PLATINUM mode is enabled by setting CSR2, bit 12 OFF; CSR2, bit 14 ON; and CSR2, bit 15 OFF See the Protocol Mode Definition table on page 5-9.

PLUS MODES

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.

Enable BURST PLUS by setting CSR2, bits 15, 14, and 12 ON, and selecting the desired maximum message packet size by setting CSR2, bit 11 to '1' for 1024 bytes, or '0' for 256 bytes. Both BURST modes are open loop, non-error-corrected modes of operation.

Enable PLATINUM PLUS by setting CSR2, bit 15 OFF, CSR2, bits14, and 12 ON, and selecting the desired maximum message packet size by setting CSR2, bit 11 to '1' for 1024 bytes, or '0' for 256 bytes.

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 algorthym 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. In real-time systems latency is an important factor in networked application design.

The **SCRAMNet+** device automatically increases PLUS mode throughput (when blocking is not used) when needed due to high-throughput host, very busy network, etc.

4.4.3 Performance

NODE LATENCY

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 message path on 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.

VME 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 4.10.5 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.

4.4.4 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 value to CSR5 as described in paragraph 3.8.2.

4.5 Auxiliary Control RAM

The ACR is an 8-bit register. However, only ACR[4:0] are implemented. ACR[7:5] are not defined.

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 of every shared-memory four-byte address. The ACR byte value will control the external trigger and/or interrupt action(s) taken whenever a WRITE occurs to any byte of the shared-memory 4-byte word. Table 4-3 describes the ACR functions.

Table 4-3 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
7-5	Reserved

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

The external trigger and/or interrupt action and/or HIPRO mode for a particular shared-memory location is defined by setting these bits. Once the ACR has been defined, the ACR Enable bit (CSR0, bit 4) should be set 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** to receive network interrupts
- **Network Interrupt Enable** to transmit network interrupts
- **Interrupt on Memory Mask Match Enable** 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. Section 5 describes the **SCRAMNet+** CSRs in detail.

In order for the external triggers 1 and 2 to take place, ACR bits 2 and 3, respectively, must be set. In order for the HIPRO mode to become active, ACR, bit 4 must be set for those selected memory addresses where this is to occur. Additionally, CSR2, bit 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.

4.6 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 4-4 indicates the various sources for interrupt control.

4.6.1 Interrupt Options

Table 4-4 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 for 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.
Override RIE	CSR0[6]	Generates an interrupt to the host regardless of the ACR RIE setting.
Enable Interrupt on Error	CSR0[7]	Generates an interrupt request as specified in the CSR9 Mask register.
Network Interrupt Enable	CSR0[8]	Permits transmission of interrupt data to the network.
Override TIE	CSR0[9]	Interrupt will be transmitted 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.
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 error conditions .
VME Interrupt Priority Level (IRQ)	CSR15[7:1]	7-bit, host specific register that holds the VME Interrupt Levels

SEND/RECEIVE WITH INTERRUPTS

- Set CSR0 to '0010' *hex* to enable the Auxiliary Control RAM (ACR).
- Clear the **SCRAMNet+** ACR by writing zeros to the entire address range.
- Set the **SCRAMNet+** ACR locations that you wish to receive and/or transmit interrupts.
- Reset CSR0 to disable the ACR.
- Set CSR0 to 'F000' *hex* to insert the node and initiate the reset of the FIFOs.
- Set CSR0 to '8003' *hex* to insert the node, toggle the reset of the FIFOs and enable network activity.
- Set CSR2 to '8040' *hex* 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 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.
- Set CSR6 to the proper interrupt vector number.
- Have the interrupt service routine in place.
- Set CSR0 to '812B' *hex* to enable receive and transmit interrupts.
- Set CSR0 to '81AB' *hex* if you wish to enable interrupt on errors.

SEND/RECEIVE WITHOUT INTERRUPTS

- Set CSR0 to 'F000' *hex* to insert the node and initiate the reset of the FIFOs.
- Set CSR0 to '8003' *hex* to insert the node, toggle the reset of the FIFOs and enable network activity.
- Set CSR2 to '8040' *hex* 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 conditions.

4.7 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

4.7.1 Network Data WRITE

As indicated in Figure 4-2, the Transmit Enable (CSR0, bit 1) must be set before any message can be sent. Only those nodes which have the Transmit Interrupt Enable (ACR, bit 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, bit 0) bit set for that address will generate an interrupt signal to their host processor.

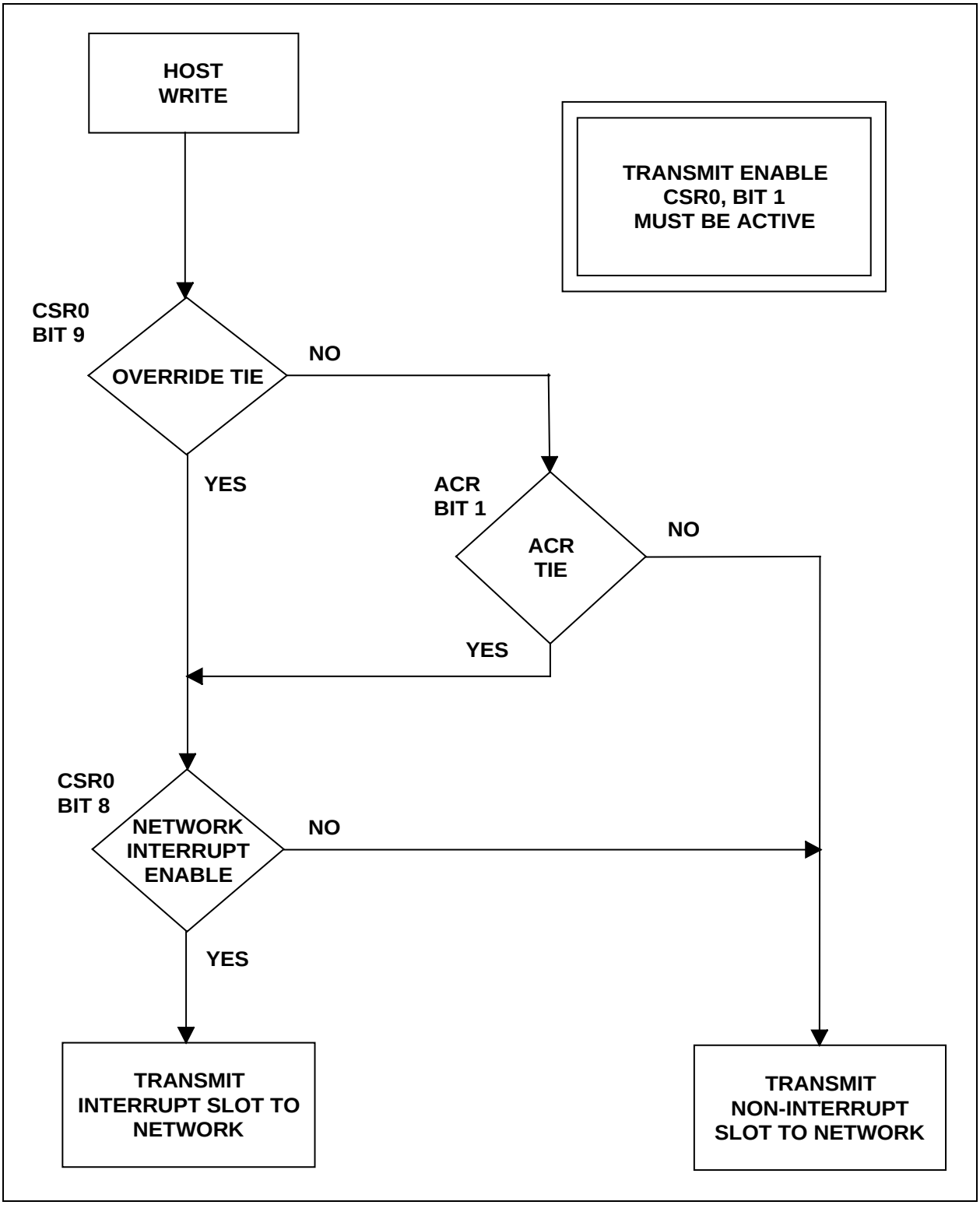


Figure 4-2 Transmit Interrupt Logic

The host issues a WRITE to **SCRAMNet+** shared memory. If Override TIE (CSR0, bit 9) or ACR TIE (ACR, bit 1) is set and Network Interrupt Enable (CSR0, bit 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 4-2, page 4-5)

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 these two methods, an interrupt can be generated and received by the same host processor if desired. This condition is termed as Self-Interrupt.

FORCED

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

The masked or selected 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 4.5: Auxiliary Control RAM.

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

Receive Interrupt logic is described in Figure 4-3. If a native message is received and Write Own Slot (CSR2, bit 9) is enabled, and Enable Interrupt on Receipt in own Slot (CSR2, bit 10) is set, the logic then checks for Receive Interrupt Enable. If Override RIE (CSR0, bit 6) is set or ACR RIE (ACR, bit 0) is set, and if Interrupt Mask Match Enable is set, the address is placed on the Interrupt FIFO.

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

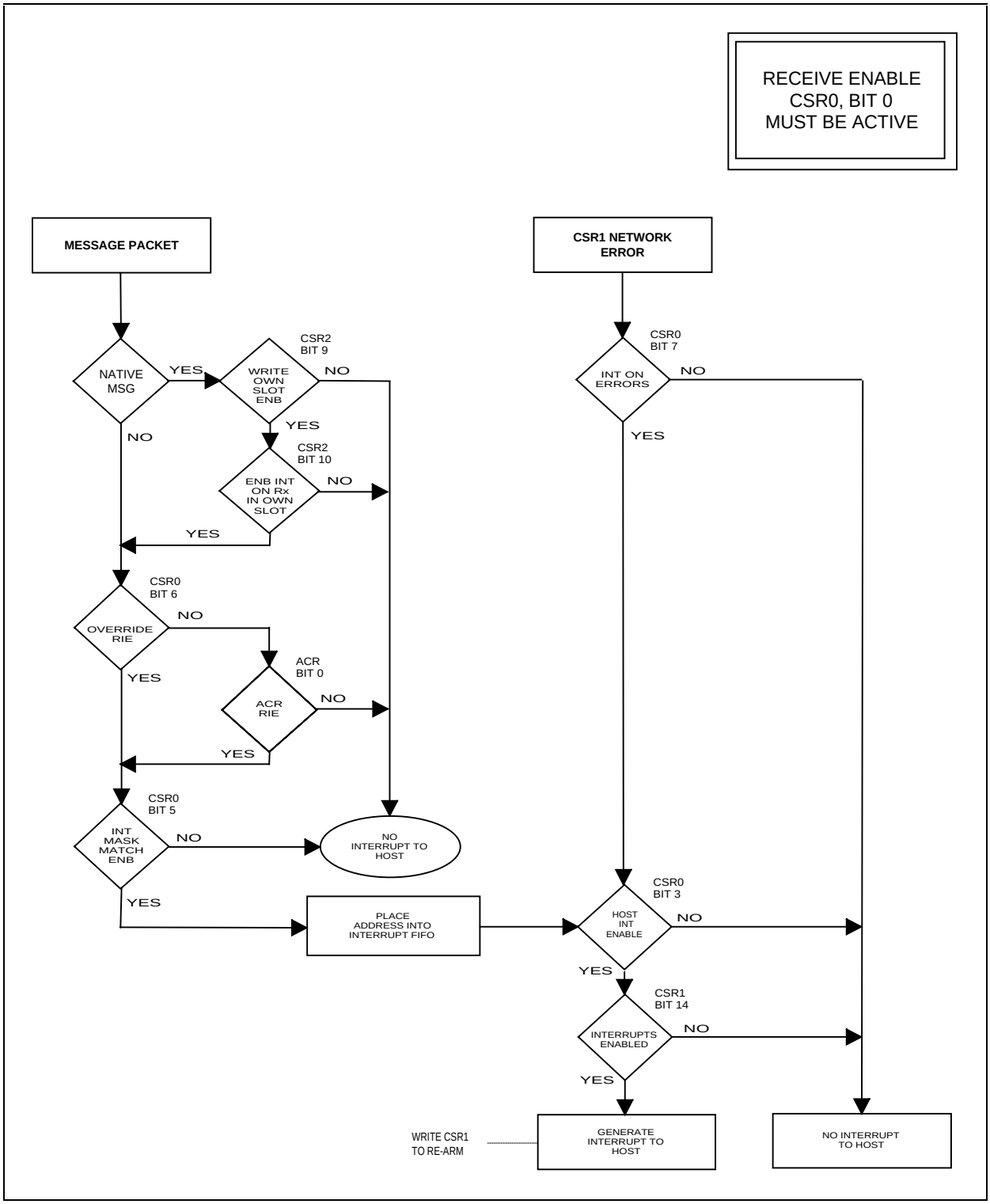


Figure 4-3 Receive Interrupt Logic

4.7.2 Network Error

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

Table 4-5 Interrupt Error Conditions

Bit	Interrupt
0	Transmit FIFO Full
1	Transmit FIFO Not Empty
2	Transmit FIFO $\frac{7}{8}$ 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

Each of these conditions is identified by the corresponding bit being set (value 1) in CSR1. If any of the preceding conditions are set and the Interrupt On Memory Mask Match Enable (CSR0, bit 5) is set, then an interrupt will be generated to the host computer. Additional information about each error condition is contained in Section 5, Table 5-2: CSR1.

If a Network Error is received (Figure 4-3), and if Interrupt on Error (CSR0, bit7) and Host Interrupt Enable (CSR0, bit 3) are set, and Interrupts are Enabled (CSR1, bit 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.

There must also be an interrupt vector to the Interrupt Service Routine. The data vector is stored in CSR6, and the error vector is in CSR7. CSR6 is associated with a memory update and CSR7 is used to identify an error interrupt. Details are included in the *Programmer's Reference Guide* (Doc. Nr. C-T-MR-PROGREF#-A-0-A2).

4.7.3 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 has a longword (32-bit) oriented shared memory. External Triggers and Interrupts will occur when any of the four bytes associated with a long word 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.

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 CLEAR (value is 0) then the Interrupt FIFO is empty. Therefore, the interrupt was due to an error, assuming that Interrupt on Errors is enabled.

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, bit 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. See Page 4-28 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.

When exiting the Interrupt Service Routine, if an interrupt occurs before enabling interrupts, the interrupt will be placed in the Interrupt FIFO and the interrupt will occur when the interrupts are armed.

4.8 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 bit 2 enables)
- Trigger 2 - Network WRITE (ACR bit 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.

4.9 General Purpose Counter/Timer

This 16-bit counter/timer can be programmed by changing CSR9, bits 13 and 14 to select the desired mode as described in **Error! Reference source not found.**. CSR8, bit 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, bit 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 Section 5, page 5-7, 5-13, and 5-16 for more information.

4.9.1 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 5-13. Table 4-6 describes the counter/timer modes available:

Table 4-6 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 when the next 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 MHz clock. Counter will roll over every 111.8 ms. Assertion of Trigger 2 will clear the counter.

4.9.2 Rollover/Reset

A rollover/reset can generate an interrupt if selected to do so from the error mask register CSR9, bit 12. When this bit is set, whenever the counter register (CSR 13) rolls over or overflows, an interrupt is generated to the host system. The interrupt-on-errors mode (CSR0, bit 7) must be enabled in order for this to work properly. The counter/timer will roll over when it reaches 65,536.

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

4.9.3 Presetting Values

The counter/timer register counts upward and may be preset with a value to arrive at the desired interrupt interval. To set an interrupt to occur every 100 ms, the counter register is preloaded with ‘6919’, so that when the counter reaches 65,536, only 100 ms would have passed instead of 111.8 ms.

The value of ‘6919’ was arrived at by dividing the desired interrupt time of 100 ms (100,000 µs) by the increment frequency of 1.760 µs which results in 58,616. This is the number that would be in the counter register after 100 ms. So, in order to offset the start of the counter which will not rollover until 65,536, subtract 58,616 to get the starting value.

4.10 Modes of Operation

4.10.1 Data Filter

Many implementations of shared-memory tend to rewrite data values to memory which 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 4-4, page 4-18.

Two bits in CSR0 control the operation of data filtering (see Section 5 for details of CSR operation):

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

If bit 11 is OFF and bit 10 is ON, only the address space above 4 K bytes of **SCRAMNet+** memory is data-filtered. If both bits 11 and 10 are ON, all **SCRAMNet+** memory is data-filtered. In either case, bit 10 must be ON for any data filtering to take place on that node.

4.10.2 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 will be to memory but will be prevented from going onto the network. The second 16-bit WRITE to memory will trigger the WRITE of the 32-bit location to the network. Because the first 16-bit WRITE must be to the SCRAMNet+ memory, the Disable Host to Shared Memory Write (CSR2, bit 8) cannot be used with HIPRO.

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.

HIPRO mode is selected for those memory addresses which have ACR, bit 4 set. **HIPRO** Enable (CSR2, bit 13) must also be set. A non-HIPRO location WRITE should be used to synchronize the HIPRO flags.

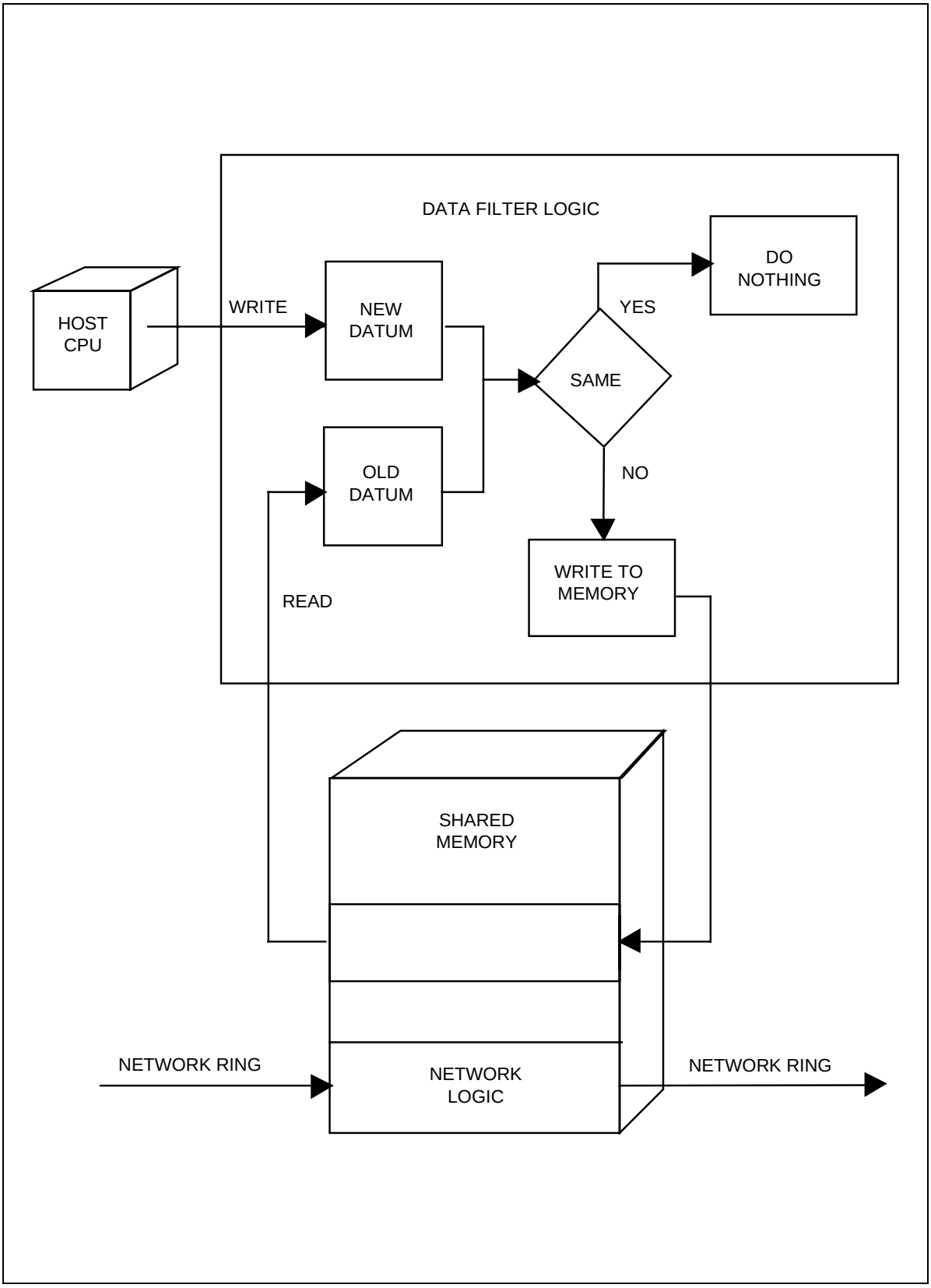


Figure 4-4 Data Filter Logic

READ

The HIPRO READ is controlled by CSR16. This register is CSR enabled and ACR location selectable.

CSR16, bit 0 ON - HIPRO READ enabled for every longword address location.

CSR16, bit 1 ON - HIPRO READ enabled for all ACR selected HIPRO WRITE locations only. Bit 0 must be enabled to use this mode.

To effectively conserve host cycles, and in turn increase throughput, HIPRO READ mode allows the host to store half (1 Shortword or 2 bytes) of the information during the first half access of the data on that longword boundary. On the next host READ operation (not the same location within the same longword boundary) the remaining data is provided without issuing another READ to shared-memory.

4.10.3 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 4-7 depicts the data path for the Monitor and Bypass mode.

Table 4-8 depicts the data path for Wire Loopback Mode.

Table 4-9 depicts the data path for Mechanical Switch Loopback Mode.

Table 4-10 depicts the data path for the Fiber Optic Loopback Mode.

Table 4-11 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 4-7 Monitor and Bypass Mode States

State	Register	Setting
Receive Enable	CSR0, bit 0	ON
Transmit Enable	CSR0, bit 1	OFF
Insert Enable	CSR0, bit 15	OFF
Enable Wire Loopback	CSR2, bit 7	OFF

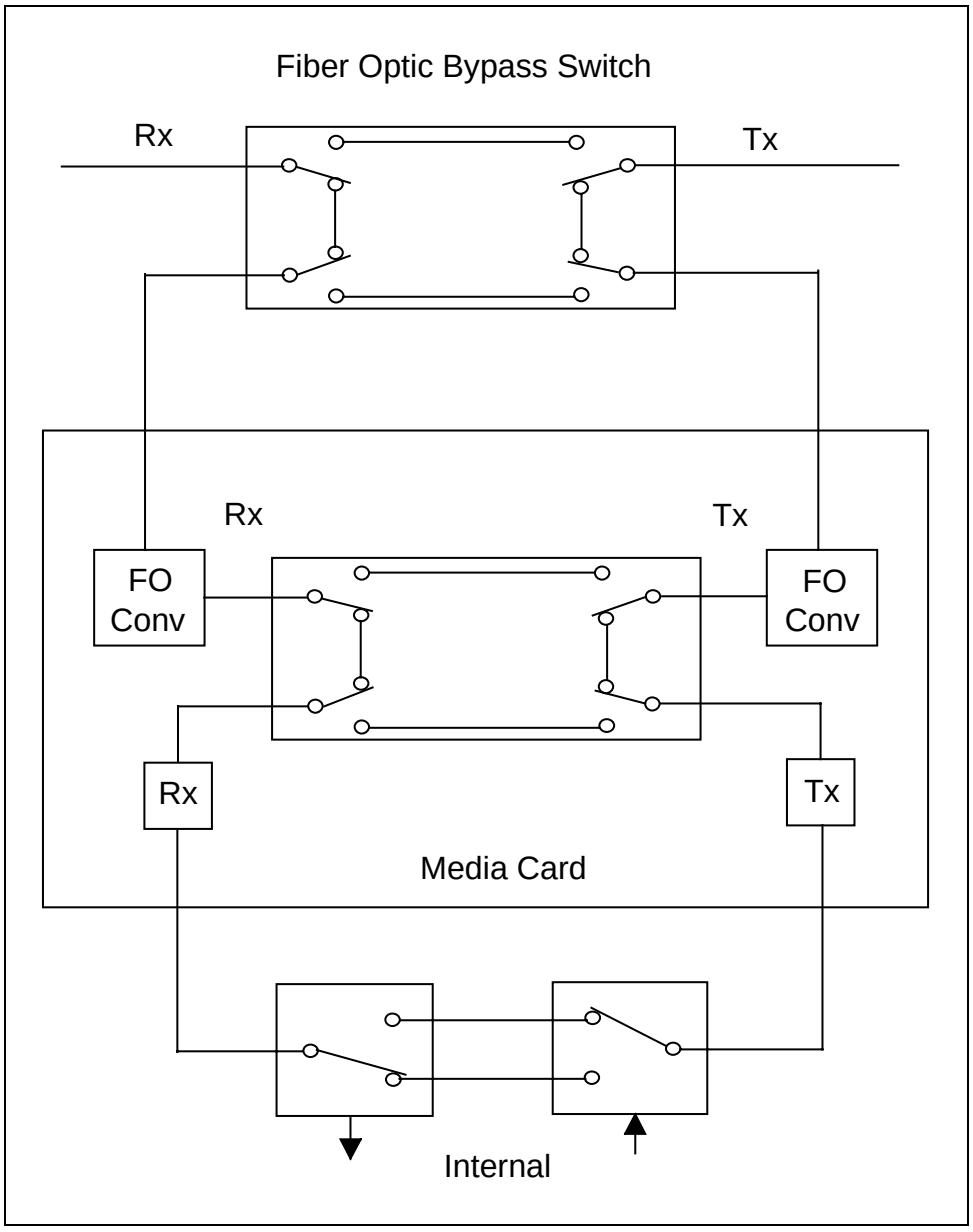


Figure 4-5 Monitor and Bypass Mode

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 4-8 Wire Loopback Mode States

State	Register	Setting
Receive Enable	CSR0, bit 0	OFF
Transmit Enable	CSR0, bit 1	OFF
Insert Enable	CSR0, bit 15	OFF
Enable Wire Loopback	CSR2, bit 7	ON

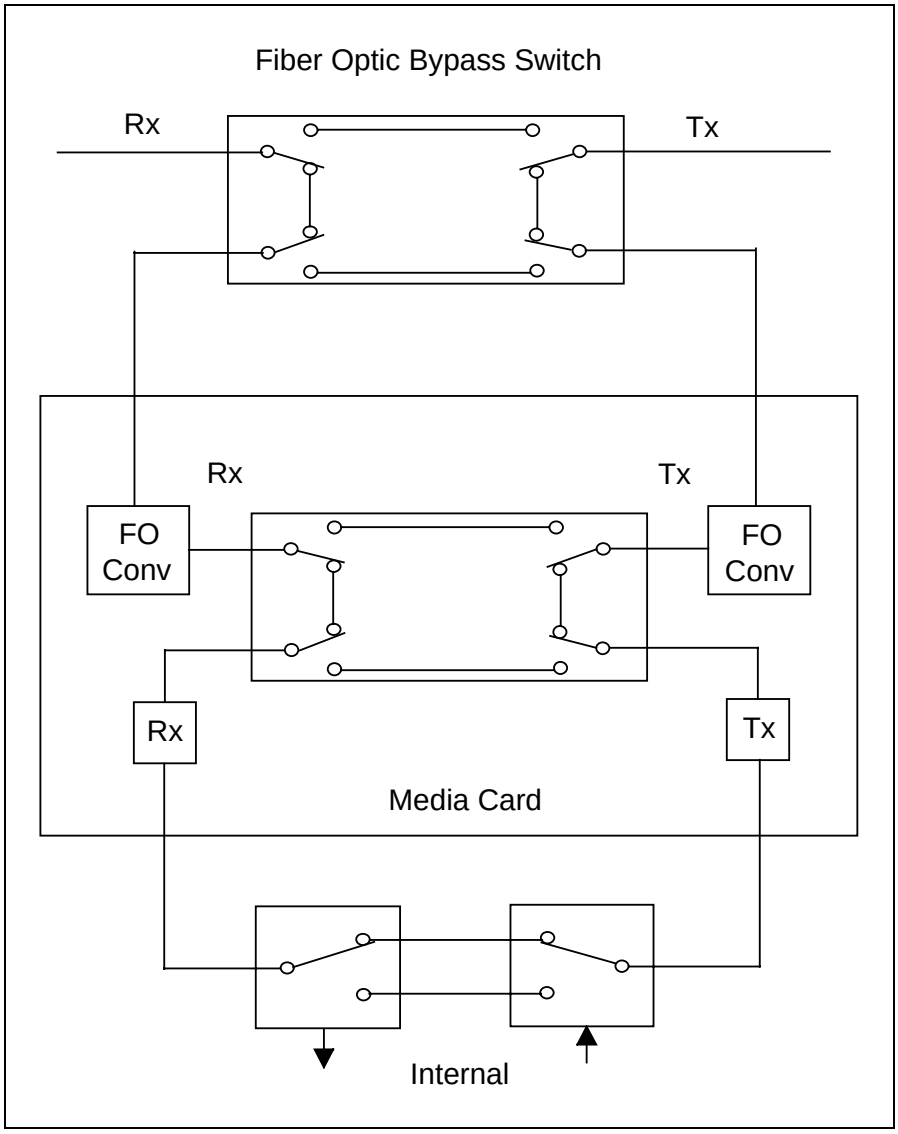


Figure 4-6 Wire Loopback Mode

MECHANICAL SWITCH LOOPBACK MODE

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

Table 4-9 Mechanical Switch Loopback Mode States

State	Register	Setting
Receive Enable	CSR0, bit 0	ON
Transmit Enable	CSR0, bit 1	ON
Insert Enable	CSR0, bit 15	ON
Enable Wire Loopback	CSR2, bit 7	OFF
Mechanical Switch Override	CSR8, bit 11	OFF

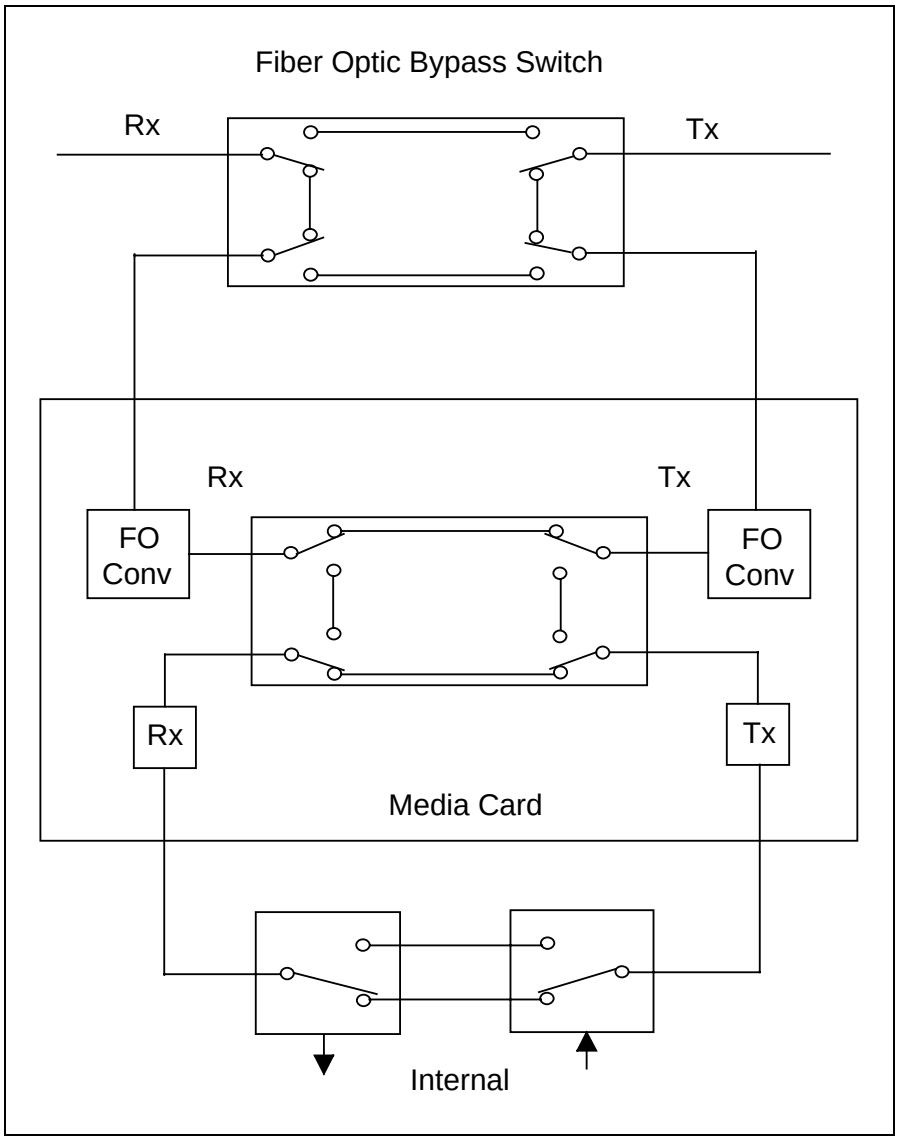


Figure 4-7 Mechanical Switch Loopback Mode

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 4-10 Fiber Optic Loopback Mode States

State	Register	Setting
Receive Enable	CSR0, bit 0	ON
Transmit Enable	CSR0, bit 1	ON
Insert Enable	CSR0, bit 15	ON
Enable Wire Loopback	CSR2, bit 7	OFF
Disable Fiber Optic Loopback	CSR2, bit 6	OFF
Mechanical Switch Override	CSR8, bit 11	ON

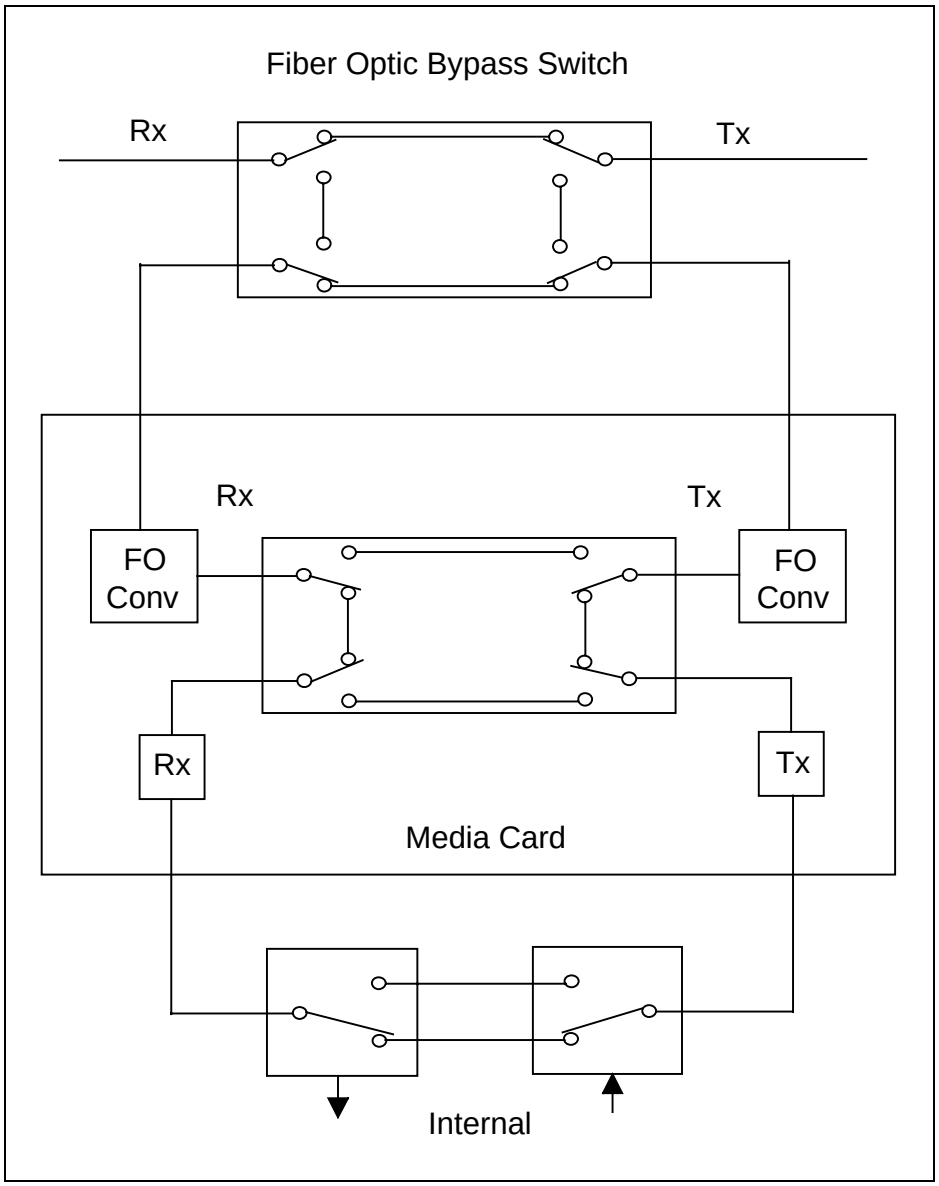


Figure 4-8 Fiber Optic Loopback Mode

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, bit 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 a Fiber Optic Bypass Switch because it would cause a break in ring continuity.

4.10.4 Node Insert Mode

In this mode the node becomes part of the network (Figure 4-9).

Table 4-11 Node Insert Mode

State	Register	Setting
Receive Enable	CSR0, bit 0	ON
Transmit Enable	CSR0, bit 1	ON
Insert Node	CSR0, bit 15	ON
Enable Wire Loopback	CSR2, bit 7	OFF
Disable Fiber Optic Loopback	CSR2, bit 6	ON
Mechanical Switch Override	CSR8, bit 11	ON

NOTE: A node in Wire Loopback mode and Insert Node will create a break in the network ring which will disable all nodes. The Wire Loopback and Fiber Optic Loopback and/or Mechanical Switch loopback modes should not be enabled simultaneously.

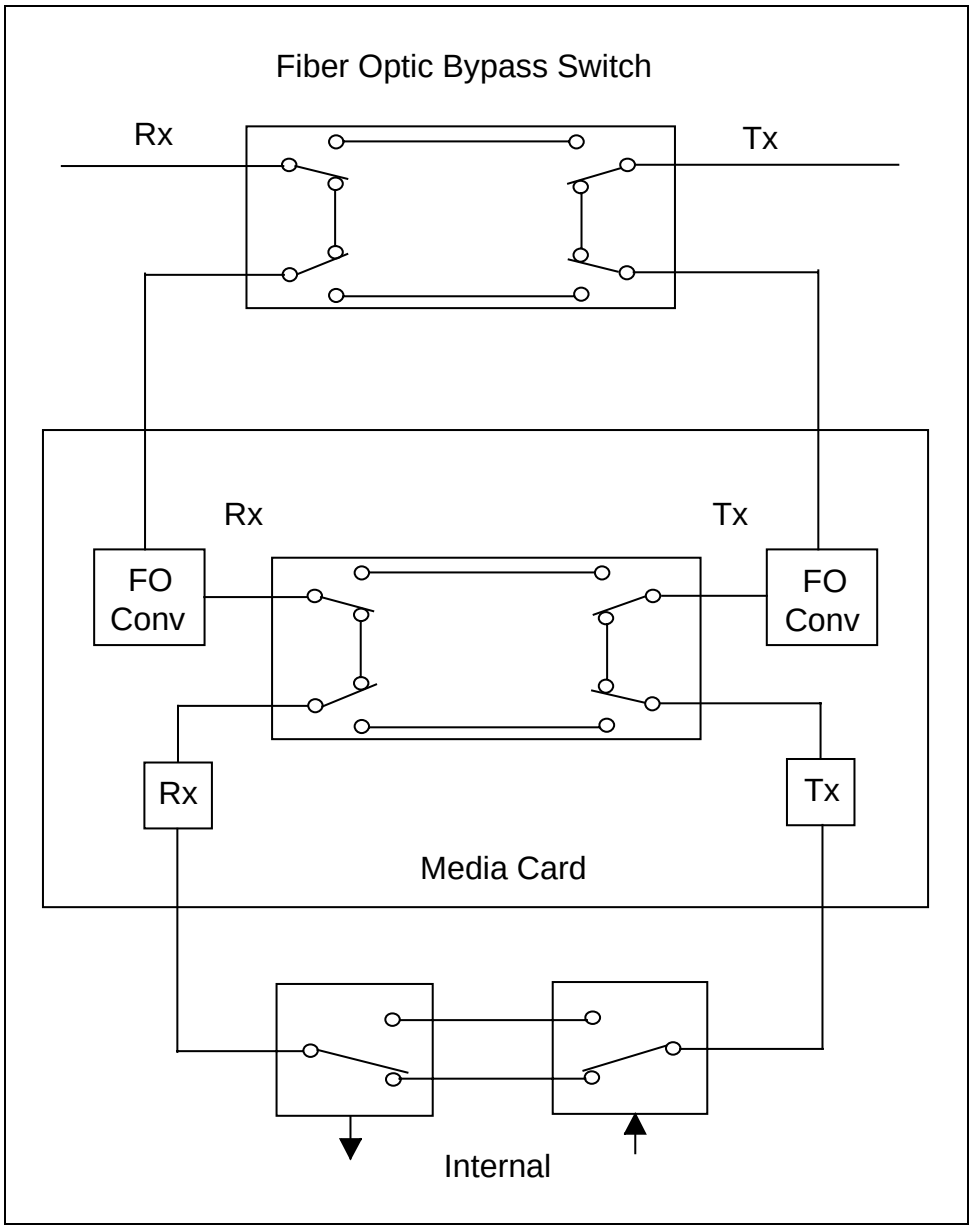


Figure 4-9 Insert Mode

4.10.5 VME Holdoff Mode

To enable VME Holdoff, set CSR8, bit 1 OFF. The VME 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 **SCRAMNet**+ memory on the VMEbus at such a rate that the Transmit FIFO becomes full (1024 deep), data could be lost.

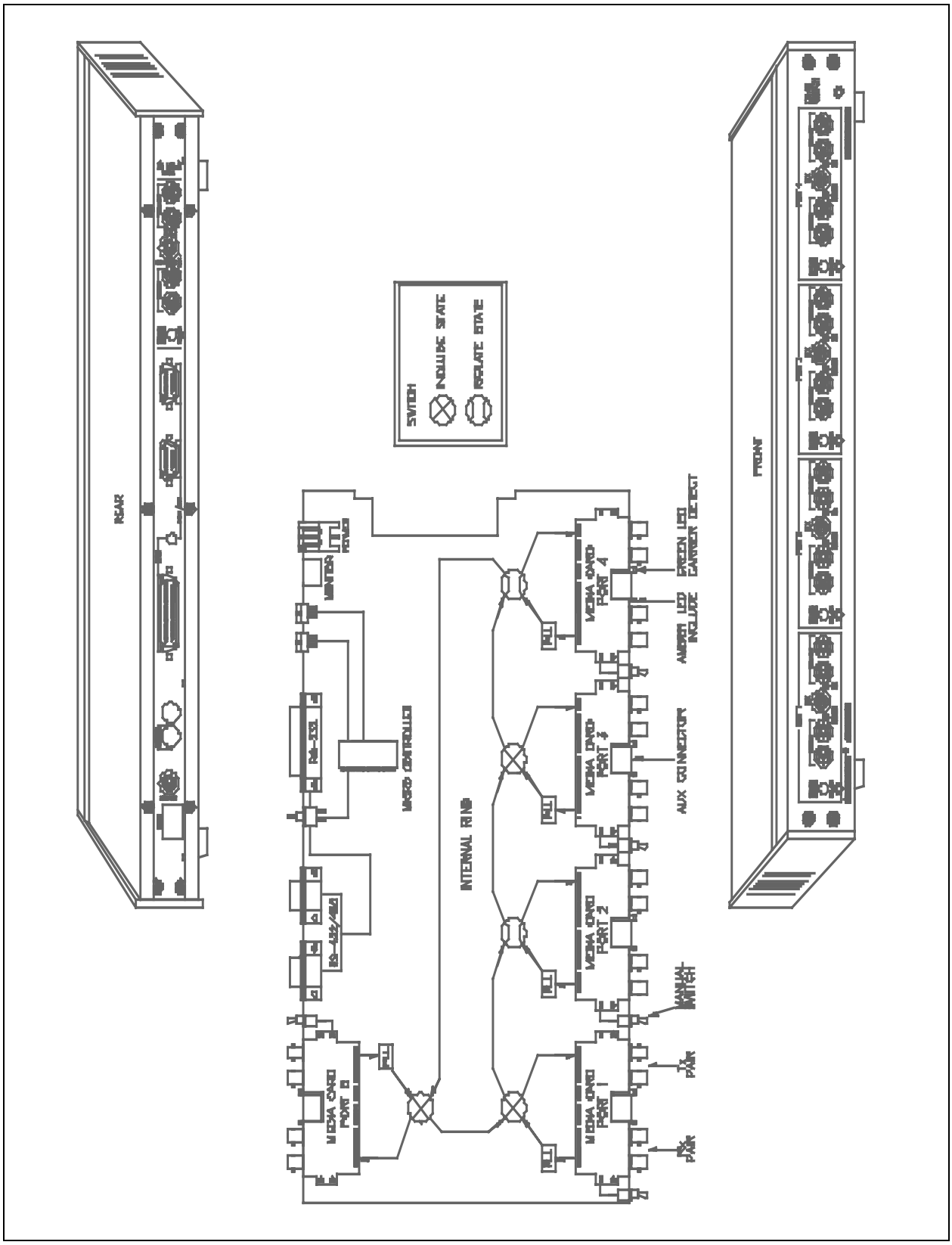


Figure 4-10 Quad Switch

In the event that the Transmit FIFO becomes full, the hardware will automatically extend the next VME write cycle until the Shared-memory FIFO empties at least one message. This prevents the loss of any data and is transparent to the user.

4.10.6 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, bits 7 and 8 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, bit 9 as well. CSR2, bit 7 is the Disable Host to Shared Memory Write.

4.11 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 4-10. 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 5 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 MME and HIE must be set in order to re-arm 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.

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

CSR6 Bits 0-7 contain the interrupt vector.

If an interrupt has been received by the host processor from the SCRAMNet+ Network interrupt logic and the appropriate interrupt vector has been initialized in CSR6, then the Interrupt Service Routine linked to that vector 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-enable interrupts
        Return from interrupt service routine
```

Figure 4-11 Interrupt Service Routine

5. 0 CSR DESCRIPTIONS

5.1 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 A7C3 *hex* 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 5-1 CSR0	
Bits	**General SCRAMNet+ Enable and Reset (READ/WRITE)**
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.

Table 5- CSR0 (Continued)	
Bits	**General SCRAMNet+ Enable and Reset (READ/WRITE)**
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 will 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 bit 0 - **Receive Interrupt Enable** - Setting this bit will generate an interrupt to the host for network interrupt data received in this location.
	ACR bit 1 - **Transmit Interrupt Enable** - Setting this bit will generate an interrupt to the network for a host WRITE to this shared memory location.
	ACR bit 2 - **External Trigger 1** - Setting this bit will generate a trigger signal to an external connector whenever there is a host READ/WRITE access to this shared memory location.
	ACR bit 3 - **External Trigger 2** - Setting this bit will generate a trigger signal to an external connector whenever there is a network WRITE to this shared memory location.
	ACR bit 4 - **HIPRO location enable** - Setting this bit will cause 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 will be 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 will cause an interrupt request.
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 will be sent out on the network regardless of the status of the ACR Transmit Interrupt bit.

Table 5-1 CSR0 (Continued)

Bits	General SCRAMNet+ Enable and Reset (READ/WRITE)
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 zero to one and back to zero 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. **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, bit 6) is disabled. This bit is invalid when the Enable Wire Loopback (CSR2, bit 7) is ON.

Table 5-2 CSR1	
Bits	**Error Indicators** (READ Only with WRITE/RESET for interrupts) Reading CSR1 will reset the latched error conditions by clearing bits 0,2,4,6,7,8,9,10,11,12,13.
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+** 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 Almost 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, bit 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 coax) 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 is ever re-transmitted. This is considered to be an error condition.
10	**Transmit Retry Time-out (Latched)** - This bit is set if the Transmit Retry error (CSR1, bit 9) was the result of a time-out while waiting for the transmitted message to return.

Table 5-2 CSR1 (continued)	
Bits	**Error Indicators (READ Only with WRITE/RESET for interrupts)**
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, bit 9; CSR9, bit 13; and CSR9, bit 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.
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 no interrupts will be received by the host. 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 mode (CSR2, bit 6) is dependent upon the Fiber Optic Bypass Switch being installed.

Table 5-3 CSR2	
Bits	**Node Control (READ/WRITE)**
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, bit 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, bit 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, bit 1) is ON.
9	**Write Own Slot Enable** - When this bit is set, the message slot 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, bit 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, bit 9, is also set. This coupling enables a host processor to interrupt itself (Self Interrupt).
11	**Message Length Limit** - Variable maximum message size: 1024 bytes or 256 byte. It is used in conjunction with CSR2, bits 12, 14 and 15 to enable Plus mode communication protocols.

Bit 10	Bit 9	Bit 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

Table 5-3 CSR2 (continued)	
Bits	**Node Control (READ/WRITE)**
12	**Variable Length Messages on Network** - When ON, this bit enables variable length messages. It is used in conjunction with CSR2, bits 11, 14 and 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 bit 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. HIPRO will not work when Disable Host to Memory WRITE (CSR2, bit 8) is set.
14	**Multiple Messages** - This bit allows multiple native messages on the network. It is used in conjunction with CSR2, bits 11, 12 and 15 to enable the BURST mode communication protocol (see below).
15	**No Network Error Correction** - This bit is used in conjunction with CSR2, bit 12 and CSR2, bit 14 to enable communication protocols: BURST or PLATINUM mode and the variable length message PLUS (+) mode (see below).

SCRAMNet+ Protocol Mode Definition

Network Mode	CSR2, Bit 15	CSR2, Bit 14	CSR2, Bit 12	CSR2, Bit 11
	No Error Correction	Multiple Message	Variable Length	Message Size Maximum
Not Valid	0	0	0	NO MEANING
BURST	1	1	0	NO MEANING
PLATINUM	0	1	0	NO MEANING
Not Valid	0	0	1	NO MEANING
BURST+	1	1	1	1=1024, 0=256
PLATINUM+	0	1	1	1=1024, 0=256

Table 5-4 CSR3	
Bits	**Node Information (READ ONLY)**
7-0	**Node Number Count** - These bits represent the total number of **SCRAMNet+** 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. 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, bit 0 is set.
15-8	**Node Identification Number** - These bits represent the **SCRAMNet+** 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. 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, bit 0 is set.

Table 5-5 CSR4	
Bits	**Interrupt Address (LSP) (READ ONLY)**
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 5-6 CSR5	
Bits	**Interrupt Address and Status (READ ONLY)***
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.
*	WRITE the Transmit Time-out value to CSR5 and it will be stored in shadow memory.

Table 5-7 CSR6	
Bits	**External Control Status Register (READ/WRITE)**
7-0	**Interrupt Vector -** This host specific register stores the VMEbus interrupt vector for the interrupt generated by a Memory Update. This register must be pre-loaded with the vector before interrupt processing can occur. *
15-0	Reserved.

Table 5-8 CSR7	
Bits	**External Control Status Register (READ/WRITE)**
7-0	**Interrupt Vector -** This host specific register stores the VMEbus interrupt vector for the interrupt generated by a **SCRAMNet+** Error. This register must be pre-loaded with the vector before interrupt processing can occur. *
15-8	Reserved.

* Both Interrupt Vectors are tied to the same Interrupt Request (IRQ) level set in CSR15

Table 5-9 CSR8	
Bits	**General SCRAMNet+ Extended Control Register**
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 will cause bus errors. On reset, CSRs will load from EEPROM.
9	**General Purpose Counter/Timer Free Run** - Setting this bit will cause 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, COAX 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).

Bit 14	Bit 13	Bit 12	Memory Size
1	1	1	Not Used
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 5-10 CSR9	
Bits	**SCRAMNet+ Interrupt On-Error Mask***
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**

CSR8, bit 9	CSR9, bit 14	CSR9, bit 13	Utility Counter 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 5-11 CSR10		
Bits	**SCRAMNet+ Replicated Shared Memory Address (LSW)**	
0	SMA_ENABLE	**Shared Memory Address Enable**. This bit enables the on-ASIC comparator for shared-memory access.
11-1	-0-	Always zero
12	SMA12	**Shared Memory Address**
13	SMA13	
14	SMA14	
15	SMA15	

Table 5-12 CSR11		
Bits	**SCRAMNet+ Replicated Shared Memory Address (MSW)**	
0	SMA16	This is the most significant part of the replicated shared memory.
1	SMA17	
2	SMA18	
3	SMA19	
4	SMA20	
5	SMA21	
6	SMA22	
7	SMA23	
15-8		Not Used

Table 5-13 CSR12		
Bits	**SCRAMNet+ Virtual Paging Register**	
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 5-14 CSR13		
Bits	**General Purpose Counter/Timer**	
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, bits 13 and 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 5-15 CSR14	
Bits	**External Control Status Register**
15-0	Reserved

Table 5-16 CSR15	
Bits	**VME Interrupt Priority Level (IRQ) External Control Status Register**
0	Not Used
7-1	This is a 7-bit wide, host-specific, READ/WRITE register that holds the VME Interrupt Priority Level (IRQ) Bits reflect the Interrupt Priority Level. For example: IPL 5 translates to setting bit 5 (20 *hex*) **Selector Chart** Level Bit Hex 7 6 5 4 3 2 1 1 0 0 0 0 0 0 10x2 2 0 0 0 0 0 1 00x4 3 0 0 0 0 1 0 00x8 4 0 0 0 1 0 0 00x10 5 0 0 1 0 0 0 00x20 6 0 1 0 0 0 0 00x40 7 1 0 0 0 0 0 00x80 Only one level is set at a time. Multiple level selection disables the WRITE operation. For example: You cannot WRITE a value of ‘0x18’ (Level 5 and Level 4).
15-8	Reserved.

Table 5-17 CSR16	
Bits	**HIPRO READ Control Bits Register (External Control Status Register)**
1-0	This is a 2-bit wide, High Performance (HIPRO) READ Control Bits Register. Only bits 1 and 0 are valid. BIt 1 BIt 0 0 1 HIPRO READ enabled 1 1 HIPRO READ ACR enabled Bit 0 is CSR enabled. HIPRO READ enabled for every longword-address location. This is an override bit. Bit 1 is ACR selectable. HIPRO READ enabled for all ACR HIPRO WRITE (ACR, bit 4) locations only. Both 0 must also be enabled for this mode.
15-2	Reserved

This page intentionally left blank.

6. 0 PHYSICAL FEATURES

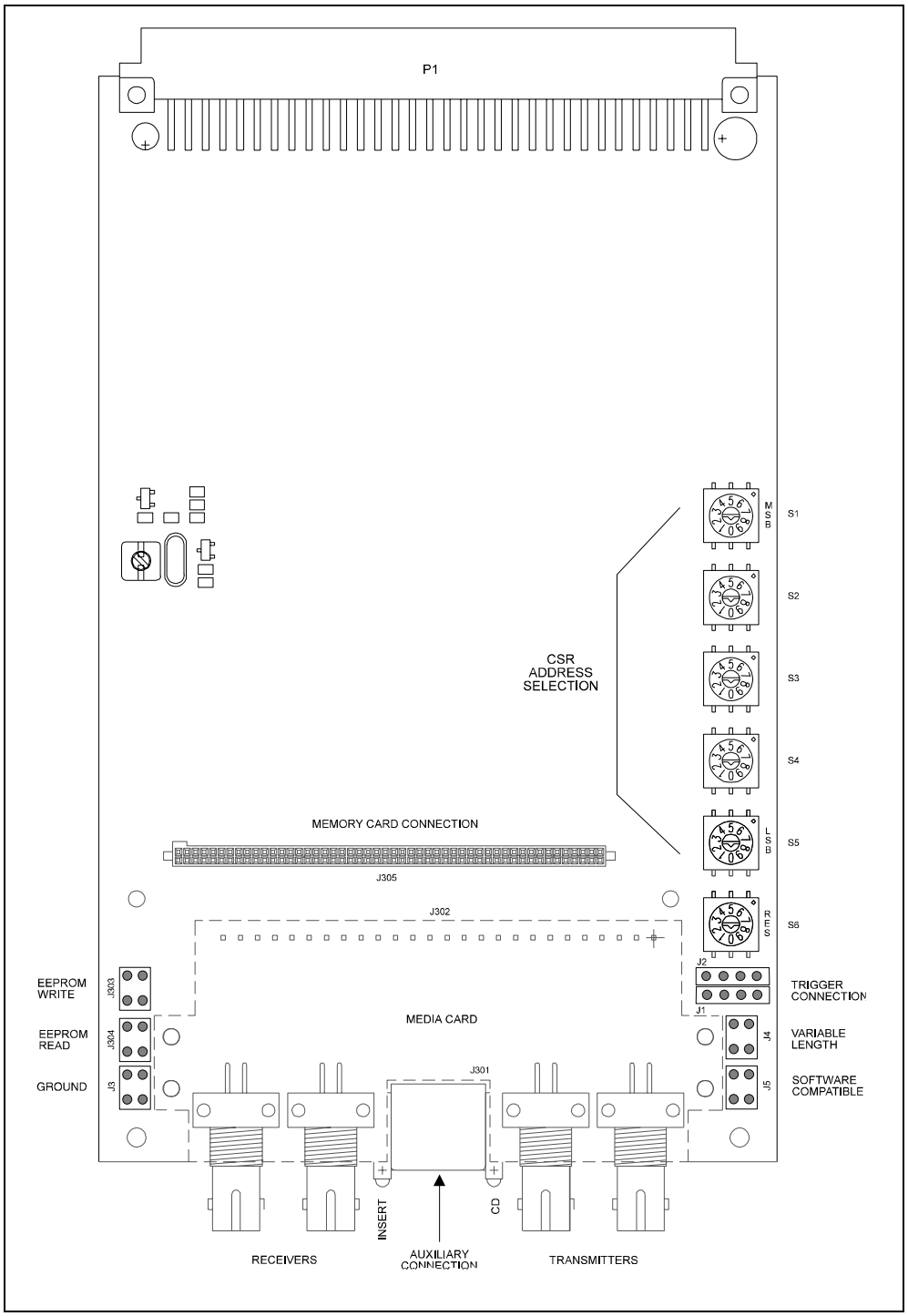


Figure 6-12 VME3U Layout

6.1 CSR Address Switches (S1-S5)

Hexadecimal rotary switches S1 through S5 are used to set the CSR address. Switch S1 contains the Most Significant Byte (MSB) and switch S5 contains the Least Significant Byte (LSB).

6.2 Resolution Bus Switch (S6)

The Hexadecimal rotary switch S6 is used to designate the host interface bus to be used for CSR addressing and the bus to be used for memory.

6.3 External Trigger Connections (J1/J2)

Eight pin connections are available to provide external access.

6.4 Ground Jumper (J3

The VME3U comes with the Signal Ground (pins 2 and 4) as the default. The Chassis Ground (pins 1 and 3) is a user option.

6.5 Variable Length Enable (VL_EN) Jumper (J4)

If the network will be using variable length message packets, jump pins 2 and 4. The default connects pins 1 and 3.

6.6 Software Compatibility (SW_CMPT) Jumper (J5)

To make the **SCRAMNet+** board software compatible with **SCRAMNet** interrupt addressing jump pins 2 and 4. Default setting connects pins 1 and 3.

6.7 Media Card Connection (J302)

The Media Card is the interface between the **SCRAMNet+** support circuitry and the transmission medium: optical fiber or coaxial cables. It is mounted on the VME3U board as a Mezzanine board. The VME3U board will support either option, depending on the configuration of your network.

One model of the Media Card has coaxial connections and the other has fiber optic connections.

The **SCRAMNet+** Support Circuitry supplies the same signal to the Media Card, regardless of the transmission medium supported by the Media Card. The Media Card converts the generic signal from the **SCRAMNet+** Support Circuitry to one appropriate to the transmission medium.

If a cabinet kit is used, the Media Card is replaced with two 8-pin cable connectors. These are used for direct-connect to the cabinet kit face plate. (See Appendix B for more information).

6.8 EEPROM WRITE (J303)

Enable to record new power-up values for the Control/Status Registers. Disable to prevent WRITE.

6.9 EEPROM READ (J304)

Enable to READ EEPROM on power-up. Disable to prevent READ.

6.10 Mezzanine Memory Card Connection (J305)

Shared memory is mounted on a mezzanine board. If no additional memory was ordered, there will not be a mezzanine board.

6.11 LED Status Indicators

6.11.1 Insert

The green Insert LED is ON when the node is Inserted into the **SCRAMNet+** Network ring. This is the result of setting CSR 0, bit 15.

6.11.2 Carrier Detect

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. 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).

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

This page intentionally left blank.

APPENDIX A
CSR SUMMARY

TABLE OF CONTENTS

A.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

A.2 CSR1 - SCRAMNet+ Error Indicators

Bit	Function	Name
0	Transmit FIFO Full	TXFF
1	Transmit FIFO Not Empty	TXFNE
2	Transmit FIFO _ 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

A.3 CSR2 - General SCRAMNet+ 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 SM 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

SCRAMNet+ Protocol Mode Definition

Network Mode	CSR2, Bit 15 No Error Correction	CSR2, Bit 14 Multiple Message	CSR2, Bit 12 Variable Length	CSR2, Bit 11 Message Size Maximum
Not Valid	0	0	0	NO MEANING
BURST	1	1	0	NO MEANING
PLATINUM	0	1	0	NO MEANING
Not Valid	0	0	1	NO MEANING
BURST+	1	1	1	1=1024, 0=256
PLATINUM+	0	1	1	1=1024, 0=256

A.4 CSR3 - Number of Nodes and Node ID

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

A.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

A.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)

* WRITE the Transmit Time-out value to CSR5 and it will be stored in shadow memory. A value of '0' will keep host-provided data from leaving the Transmit FIFO.

A.7 CSR6 - External Control Status Register (R/W)

An 8-bit register that holds the VMEbus interrupt vector generated by a memory update. Only bits [7:0] are valid; [15:8] are reserved.

A.8 CSR7 - External Control Status Register (R/W)

An 8-bit register that holds the VMEbus interrupt vector generated due to a **SCRAMNet+** error. Only bits [7:0] are valid; [15:8] are reserved.

A.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 0 = Invoke Coax Loopback Mode	C_MECHSW
12	Memory Configuration - Size (See below)	MC10
13	Memory Configuration - Size (See below)	MC11
14	Memory Configuration - Size (See below)	MC12
15	Reserved (always 1)	1

Bit 14	Bit 13	Bit 12	Memory Size
1	1	1	Not Used
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

A.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	Utility Counter Modes (See below)	M_INC_TRIGS
14	Utility Counter Modes (See below)	M_INC_ERRS
15	Fiber Optic Bypass Not Connected mask	M_FO_BYPASS

CSR8 Bit 9	CSR9 Bit 14	CSR9 Bit 13	Utility Counter 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

A.11 CSR10 - SCRAMNet+ Shared Memory Address (LSW)

Bit	Function	Name
0	Enable comparator for SM access	SMA_ENABLE
11-1	Reserved	0
12	Shared Memory Address	SMA12
13		SMA13
14		SMA14
15		SMA15

A.12 CSR11 - SCRAMNet+ Shared Memory Address (MSW)

Bit	Function	Name
0	Shared Memory Address	SMA16
1		SMA17
2		SMA18
3		SMA19
4		SMA20
5		SMA21
6		SMA22
7		SMA23
15-8	Not Used	

A.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

A.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]

A.15 CSR14 - External Control Status Register

A 16-bit, READ Only SYSTRAN reserved register.

A.16 CSR15 - VME Interrupt Priority Level (IRQ) External Control Status Register

A 7-bit, READ/WRITE register that holds the VME Interrupt Priority Level (IRQ).

A.17 CSR16 - HIPRO Read Control Bit External Control Status Register

A 2-bit wide, READ/WRITE register that holds the HIPRO Read Control Bits. It is CSR enabled and ACR selectable. Only bits [1:0] are used; [15:8] are reserved.
Bit 0: HIPRO READ Enabled (CSR Enabled)
Bit 1: HIPRO READ ACR Enabled (ACR Selectable)

A.18 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 B
CABLE KIT

TABLE OF CONTENTS

B.1 Introduction

The cabinet kit for the **SCRAMNet+** Network board permits adapting the node to the host cabinet while still maintaining the shielding of the chassis.

B.2 Cable Connections

Connections between the cabinet kit bulkhead plate and the **SCRAMNet+** Network board are shown in Figure B-1. Cable lengths vary depending upon the distance between the installed board's media card and the cabinet kit.

Connect cables between the **SCRAMNet+** Network board media card receiver (Rx_1 and Rx_2) and the bulkhead plate receiver (Rx_1 and Rx_2) connections; and the media card transmitter (Tx_1 and Tx_2) with the bulkhead plate transmitter (Tx_1 and Tx_2) connections.

NOTE: It does not matter if Rx_1 or Rx_2 is connected to the bulkhead plate's Rx_1 or Rx_2 as long as both cables are connected to both of the Rx connectors. Likewise, Tx_1 and Tx_2 may be connected to the bulkhead plate's Tx_1 or Tx_2.

B.2.1 Cabinet Kit Connections

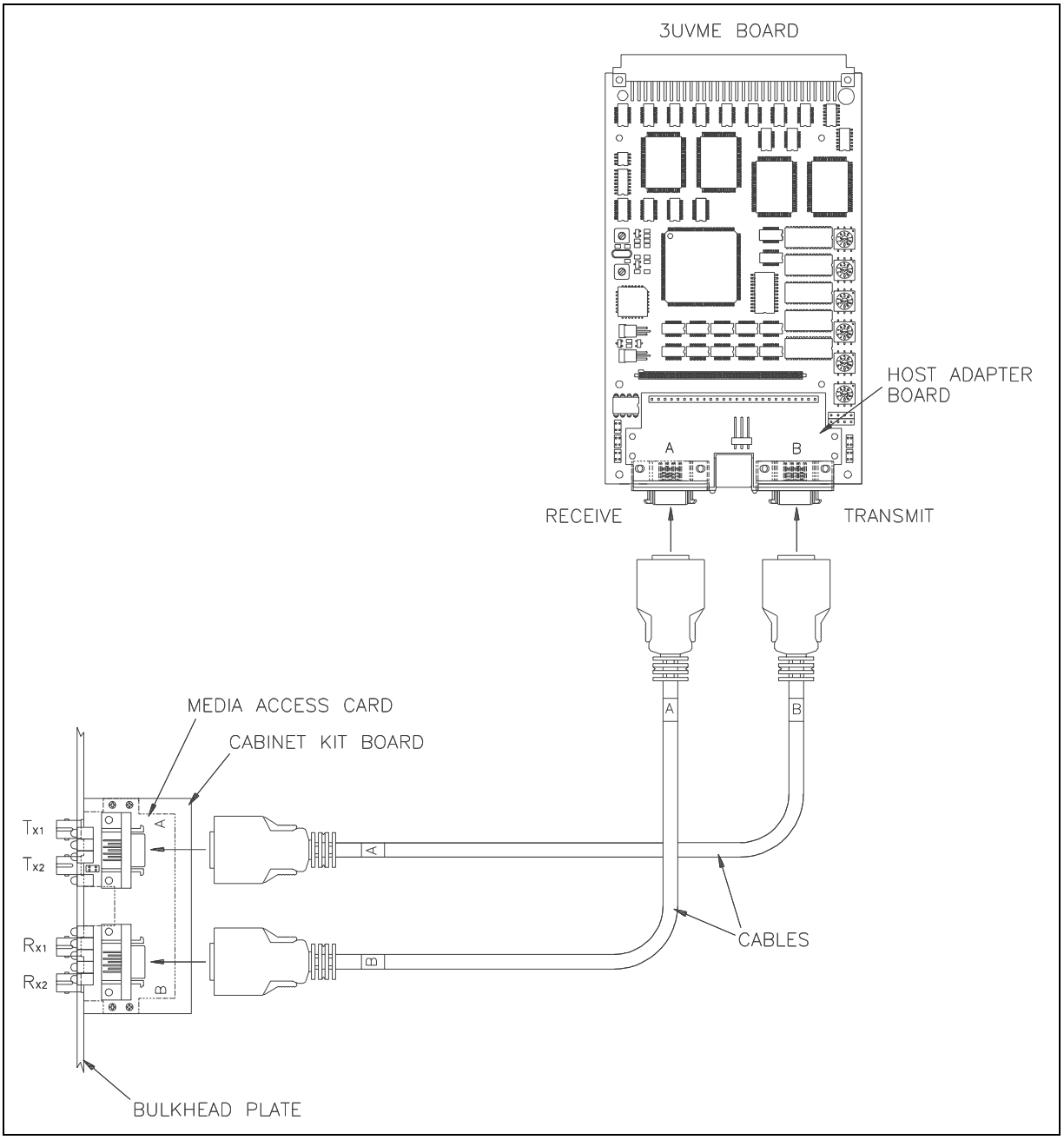


Figure B-1 Cabinet Kit Connections

APPENDIX C
SPECIFICATIONS

TABLE OF CONTENTS

C.1 Hardware Specifications

Hardware Compatibility:	VMEbus
Physical Dimensions:	
VME3U Card	6.299" x 3.937"
	3U Eurocard, one slot
Weight:	
VME3U Card	0.3 lbs (W/O SIMMs and Media Card, W/face plate)
Media Card, Fiber Optic	0.0915 lbs
Electrical Requirements:	+5 VDC, 1.5 Amps max.
Storage	
Temperature Range:	-40° to +70°C
Humidity Range:	0% to 95% (noncondensing)
Operation	
Temperature Range:	0° to +40°C
Humidity Range:	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:Standard Size:	128 Kbytes (No SIMMs)
Optional Sizes:	
Low Density SIMMs (512 KB)	512 KB, 1 MB,
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 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

C.2 Bus Voltage Specifications

VMEbus 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	--

* Not used by **SCRAMNet+** VME3U

C.3 Part Number

The VME3U adapter part number is in the form:

H-AS-D3VMEL2M-00

where:

CODE	DEFINITION
H	Hardware
AS	Top Level Assembly
D	Standard SCRAMNet+
3VME	3U 16-BIT VME
XXX	**Memory (bytes)**
	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

C.4 Board Dimensions

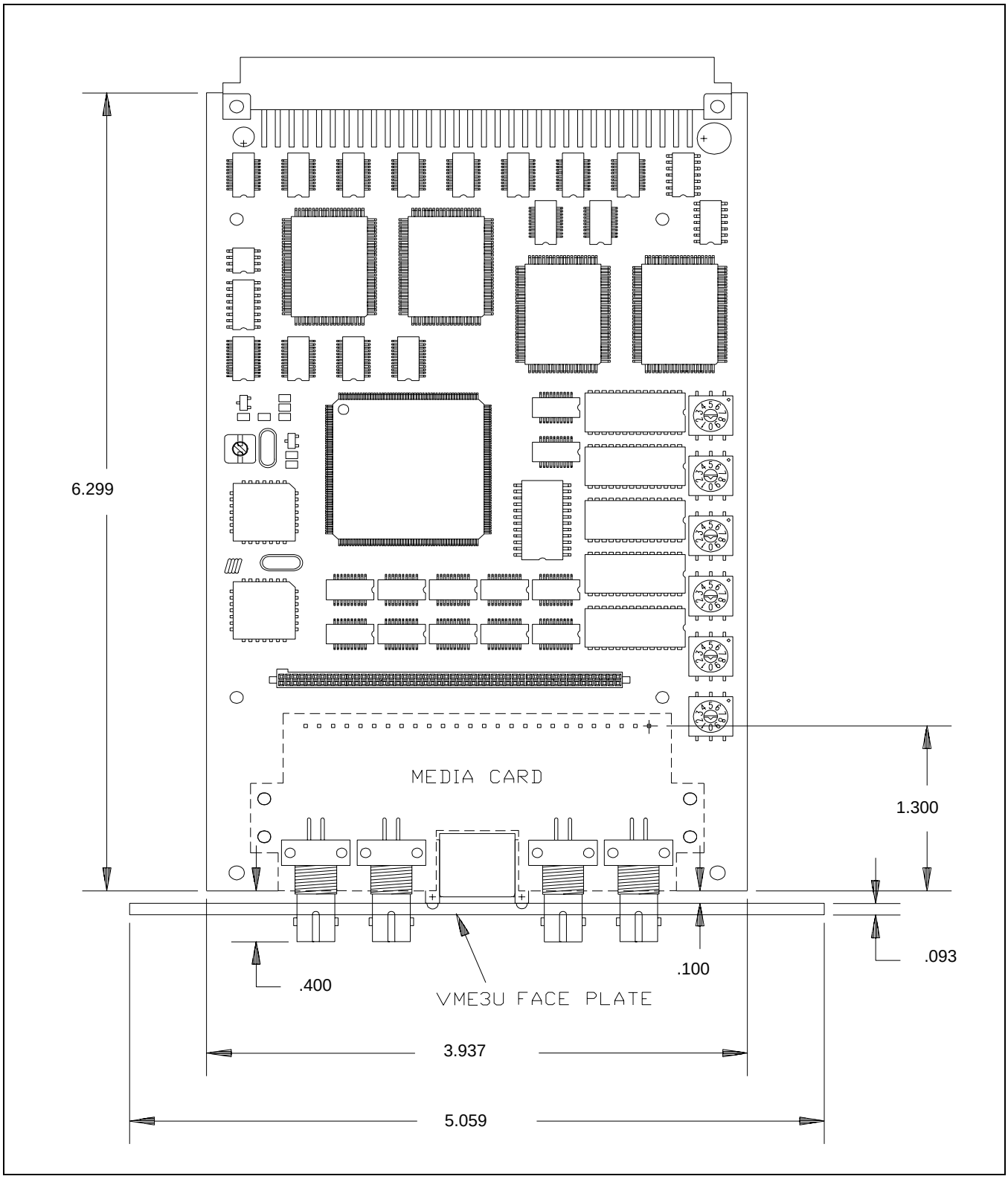
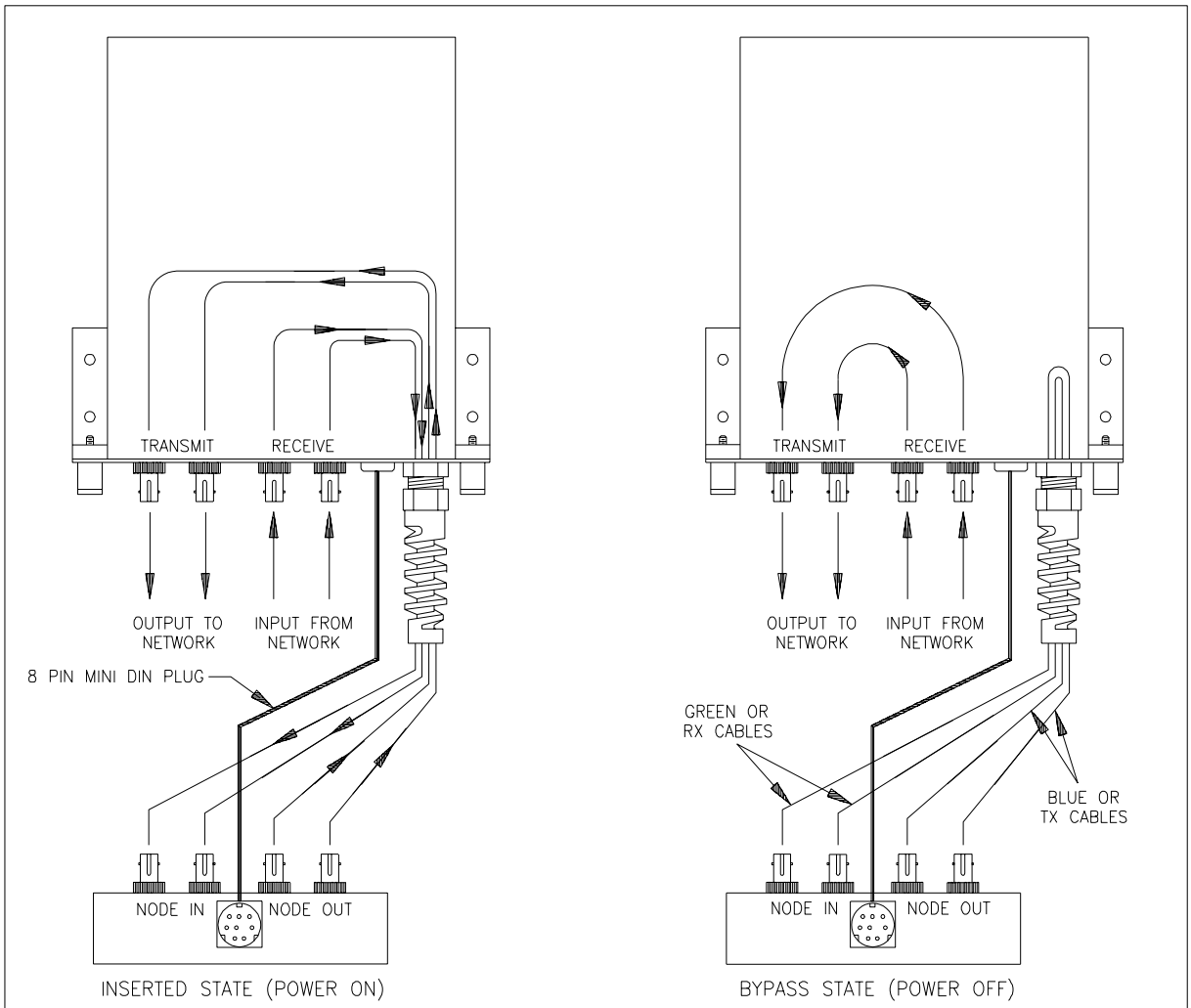


Figure C-1 VME3U Board Dimensions

C.5 Fiber Optic Bypass Switch



SPECIFICATIONS

OPTICAL PERFORMANCE

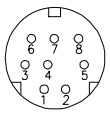
INSERTION LOSS	1.2 db TYPICAL, 1.8 db MAXIMUM
	3 – 9 db ON LOOPBACK
SWITCHING TIME	20 msec MAXIMUM
CROSS–TALK	–60 db MAXIMUM
DURABILITY	10^7 CYCLES MINIMUM
REPEATABILITY	0.02 db MAXIMUM

ELECTRICAL PERFORMANCE

SWITCHING VOLTAGE	4.3 VDC MINIMUM, 6 VDC MAXIMUM
SWITCHING CURRENT	110 mA @ 5 VDC TO SWITCH
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



NOTES:
1. INSERTION LOSS MEASUREMENTS ARE BASED ON FOTP 34, METHOD B– STEADY STATE EQUILIBRIUM LIGHT LAUNCH CONDITIONS.
2. UNIT CONFIGURED USING ST CONNECTOR RECEPTACALS AND ST CONNECTOR PLUGS.
3. LED WILL BE ON WHEN IN INSERTED STATE (POWER ON).

SYSTRAN CORP.
4126 LINDEN AVE. SUITE 100
DAYTON, OHIO 45432

SIZE	DWG. NO.	REV.
A	A–H–PR–FORELAY–50	A
	FILE: FORELAY2.DWG	SHEET 2 OF 4

Figure C-2 Fiber Optic Bypass Switch

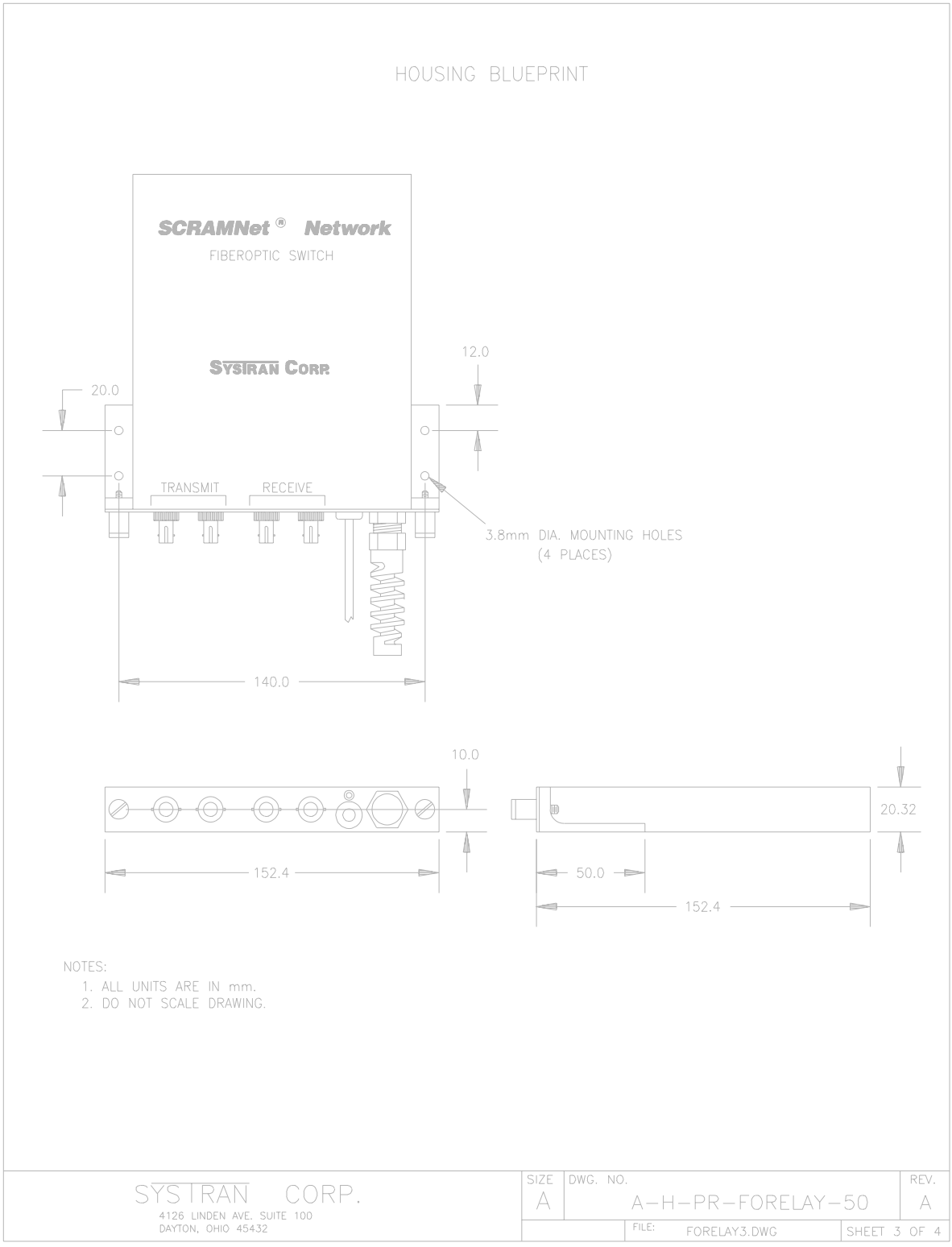


Figure C-3 Housing Dimensions

APPENDIX D
HOST ACCESS TIMING

TABLE OF CONTENTS

D.1 Introduction

The **SCRAMNet+** host access timing is comprised of three separate module timings.

- Dual-Port RAM Controller (DPRC).
- Host Interface Logic to shared memory/CSR.
- Host Interface Logic to host-specific CSR.

The first module allows shared memory to be updated by the high speed serial network without utilizing valuable CPU bus bandwidth. The second module is needed to interface shared memory and the ASIC internal CSRs to the host CPU bus. The third module is needed to interface the host-specific external CSRs to the host CPU.

D.2 Dual-Port RAM Controller Module

SCRAMNet+ memory is controlled by the DPRC. The DRPC has two ports: one for host access and one for network access. The DPRC arbitrates requests for these two ports on a first come, first serve basis. In case of a tie, the high speed serial network has priority. The first port, which is enabled for READ/WRITE, is connected to the host CPU. The second port is WRITE ONLY and is connected to the high speed serial network. There are three types of accesses to the DPRC. They are shown in Table D-1.

Table D-1 Dual Port RAM Controller Access Types

Access Type	Port	Cycle Time
Host WRITE	1	240 ns
Host READ	1	133 ns
Network WRITE	2	133 ns

D.2.1 Contention

The **SCRAMNet+** Network, being an intelligent peripheral, does buffered WRITEs to shared memory. This speeds up host WRITEs by latching the data and address when a host WRITE is detected, and then replying to the host immediately without waiting for the DPRC to finish the WRITE activity. Since the WRITE buffering is only one level deep, it is possible that on back-to-back accesses the second access will be delayed until the buffer is available (i.e., when the current DPRC cycle is finished).

This phenomenon results in the stretching of two types of cycles outside the normal case:

- **Back-to-Back WRITE cycle.** In this cycle the second reply on the bus is held off until the first WRITE cycle has finished.
- **WRITE cycle followed by a READ cycle**. In this case the READ cycle is delayed from starting until the previous WRITE cycle is completed.

NOTE: The cases resulting in stretched cycles describe a very fast host bus condition and are not normal. In reality, **SCRAMNet+** memory was designed and optimized for CPU data storage. Therefore CPU activities such as instruction fetches, instruction execution, and other miscellaneous activities will ensure the phenomenon of cycle stretching will rarely, if ever, occur.

D.3 Host Interface Logic to Shared Memory/CSR

The second module is the actual host interface logic needed to interface the ASIC resources (memory and internal CSR) to the host CPU bus. This is the logic that translates all VME host transactions to **SCRAMNet+**. This host logic is responsible for determining the format of the access.

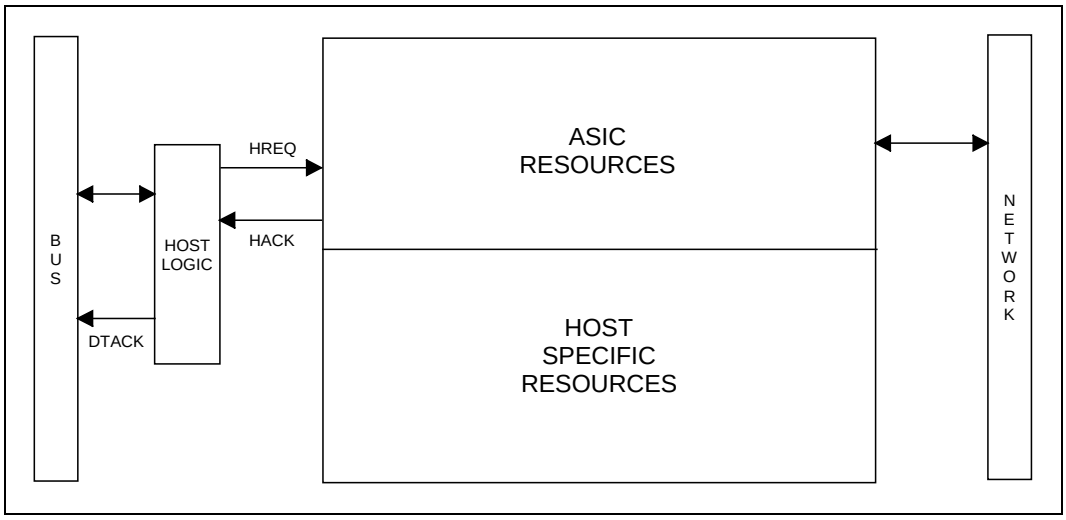


Figure D-1 ASIC Resources

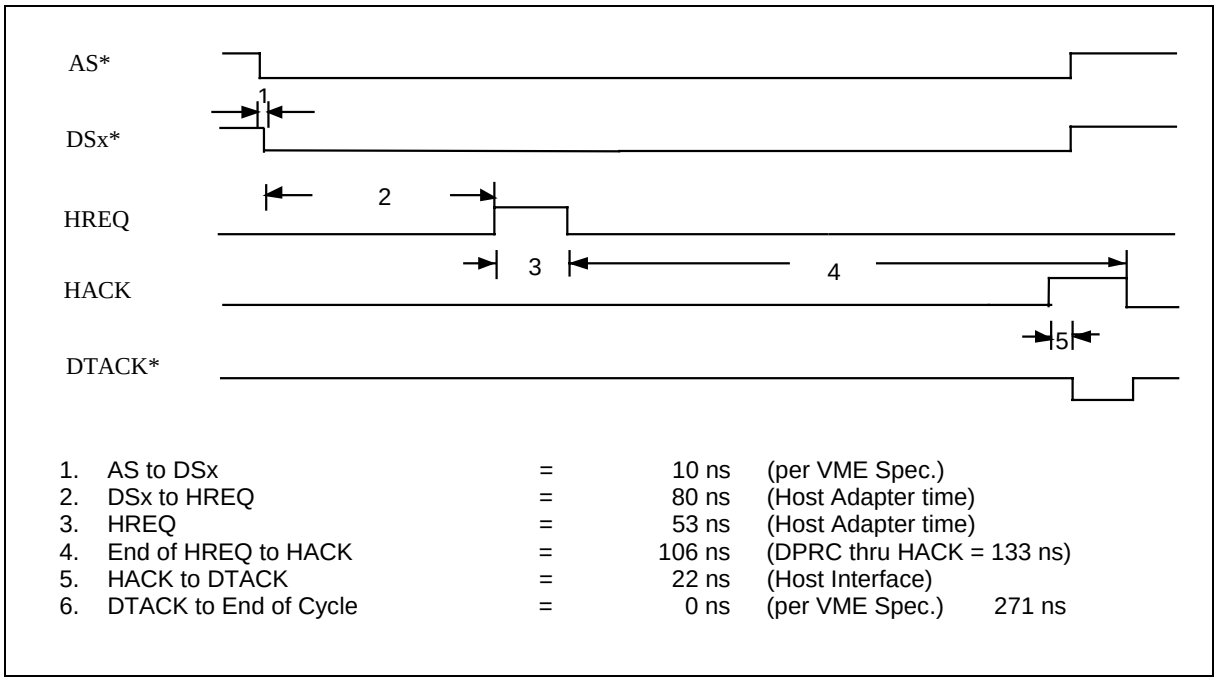
This format is typically a READ or WRITE operation to the ASIC resources; the shared memory and CSRs.

D.3.1 ASIC-Internal CSR READ

The ASIC is typically configured through a number of control and status registers (CSR). The READ and WRITE cycles are not any different from any other ASIC resource cycles. The following is a READ cycle of an ASIC-internal CSR:

- AS and DS fall
- HREQ requested for a particular CSR (address)
- Wait for HACK to provide READ data
- Assert DTACK
- Requester (Master/CPU) de-asserts AS and DS (end of cycle)

Figure D-2 is an example of a READ from an internal CSR.



1.	AS to DSx	=	10 ns	(per VME Spec.)	
2.	DSx to HREQ	=	80 ns	(Host Adapter time)	
3.	HREQ	=	53 ns	(Host Adapter time)	
4.	End of HREQ to HACK	=	106 ns	(DPRC thru HACK = 133 ns)	
5.	HACK to DTACK	=	22 ns	(Host Interface)	
6.	DTACK to End of Cycle	=	0 ns	(per VME Spec.)	271 ns

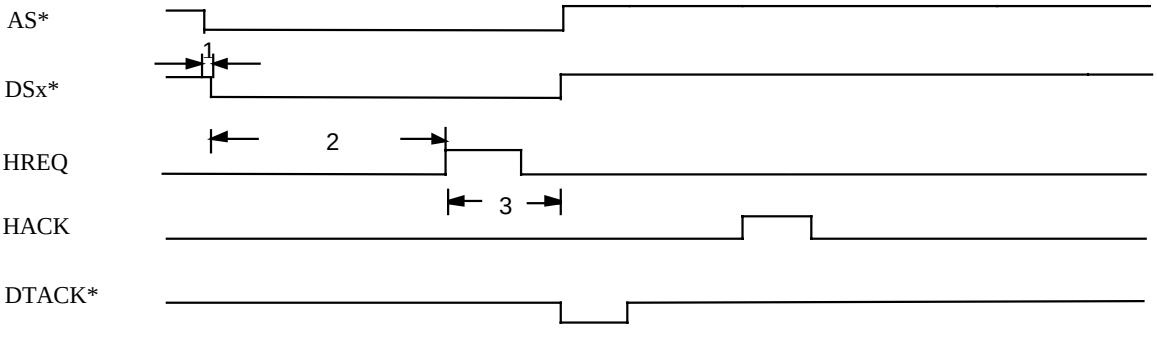
Figure D-2 READ From Internal CSR

D.3.2 ASIC-Internal CSR WRITE

This is a typical WRITE cycle of an ASIC-internal CSR:

- AS and DS fall
- HREQ requested for a particular CSR (address)
- Data accepted and ASIC asserts HACK
- Assert DTACK
- Requester (Master/CPU) de-asserts AS and DS (end of cycle)

Figure D-3 is an example of a WRITE to an internal CSR.



1.	AS to DSx	=	10 ns	(per VME Spec.)	
2.	DSx to HREQ	=	106 ns	(Host Adapter time)	
3.	HREQ to DTACK	=	53 ns	(Host Interface)	
4.	DTACK to End of Cycle	=	0 ns	(per VME spec)	169 ns

Figure D-3 WRITE to Internal CSR

D.4 Host Interface Logic to Host Specific CSR

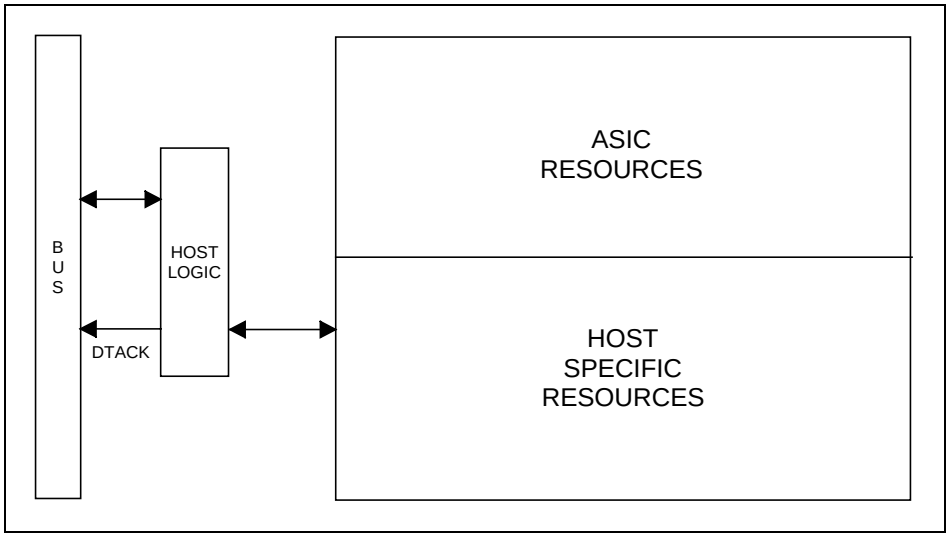


Figure D-4 Non-ASIC Resources

The third module is the actual host logic needed to interface the non-ASIC resources (external host-specific CSR) to the host CPU as depicted in Figure D-4. The Interrupt Vector register is a non-ASIC resource. Timings to this register vary significantly from that of ASIC-related resources.

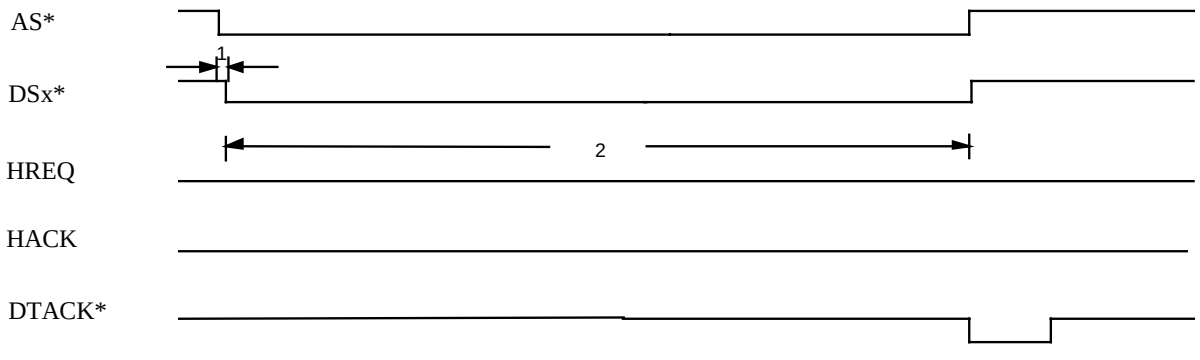
D.4.1 Host-Specific CSR READ

All READ operations on the external CSRs do not go through normal ASIC data paths. Therefore, timings for this cycle varies from other CSR READ cycles. These external CSRs are not ASIC resources and therefore do not request ASIC cycles to READ from them. Additional host logic monitors access to these locations from the host and CSR data is provided directly to the host CPU without performing a ASIC Host Request (HREQ).

The following is the general format of an external CSR READ cycle:

- AS and DS asserted
- Additional host captures request
- Provides READ data
- Asserts DTACK
- Requester (Master/CPU) de-asserts AS and DS

Figure D-5 is an example of a READ from an external CSR.



1.	AS to DSx	=	10 ns	(per VME Spec.)	
2.	DSx to DTACK	=	186 ns	(Host Adapter time)	
3.	DTACK to End of Cycle	=	0 ns	(per VME Spec.)	196 ns

NOTE: There is no Host Request (HREQ) because this is not an ASIC resource

Figure D-5 READ From External CSR

D.5 Access Times

Host access to shared memory is affected by two things:

- Priority on contention between the two ports
- Asynchronous request handling

There are minimum and maximum times which depend on Network WRITE priority and the asynchronous handling of the requests. The minimum DPRC access times (i.e., no contention between ports) are shown in. This table does not include host timing. The "worst case" scenerio resulting in maximum access times would be a host WRITE followed immediately by another host WRITE, and simultaneous receipt of three network WRITES.

Minimum access times occur when there is no contention. Maximum access times result when the worst possible case of contention occurs.

The **SCRAMNet+** cycle is initiated by the assertion of the Data Strobe (DS) and the de-assertion of the Data Strobe (DS) as the end of the cycle.

D.5.1 Typical Access Sequence

The following is the typical Host READ and Host WRITE sequence with no contention.

READ

- Address Strobe (AS) and Data Strobe fall (Start of cycle)
- Host Request (HREQ) asserted
- Host Request Pending (HREQ_PEND) asserted (DPRC accepts request)
- DPRC READ shared memory and pass the data to Host-Interface Logic
- De-assert HREQ_PEND
- Host Acknowledge (HACK)

- Release bus (DTACK)
- Requester/Host CPU de-assert DSx and AS (End of Cycle)

WRITE

- Address Strobe (AS) and Data Strobe fall (Start of cycle)
- Host Request (HREQ) asserted
- DPRC buffer data and address
- DPRC assert HREQ_PEND
- HACK asserted
- Release bus (DTACK)
- De-assert HREQ_PEND
- Requester/Host CPU de-assert DSx and AS (End of Cycle)
- DPRC WRITE to shared memory
- Host ACK (HACK)

D.5.2 Worst-Case Condition

The scenario described in all these cases considers the “worst case” operating condition. This is a very unlikely condition, and would not occur in an average configuration. The maximum timing values were generated in a laboratary with a “worst case” scenario emulation. Figure D-6 is a functional diagram of the “worst case” scenerio.

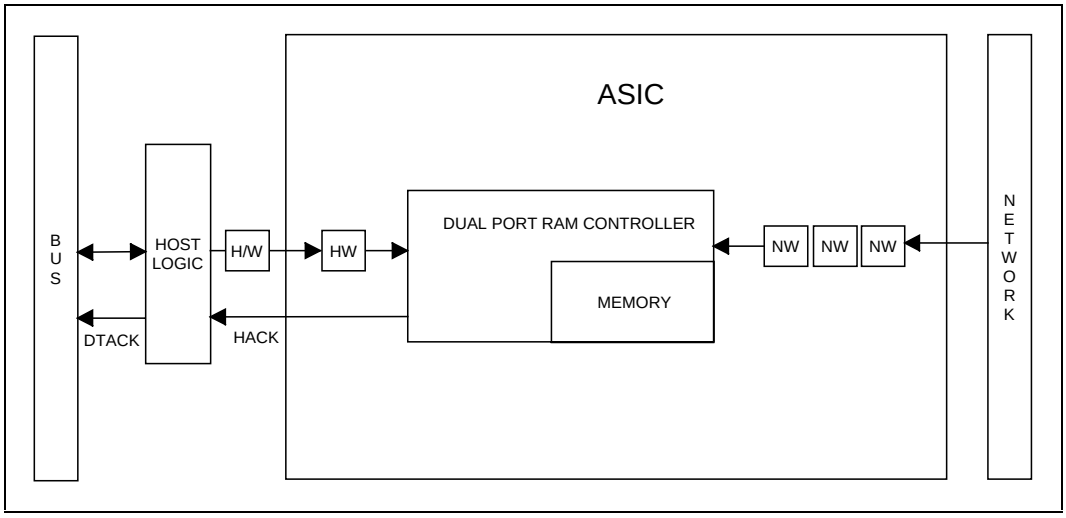


Figure D-6 Two Host WRITEs in Contention With Three Network WRITEs

CONFIGURATION

The test was conducted with two nodes on a ring and an extremely fast host CPU as a host data generator. This fast CPU is thoeretical since there is no CPU available today that is capable of maintaining the VME specification of 40 MB/sec. The two nodes were defined having their transmitters disabled, and their Interrupt FIFOs filled. The nodes were configured to transmit in Plus mode which translates to the **SCRAMNet+** maximum throughput of 16.6 MB/sec. A Plus mode packet has new data every 240 ns. The “worst case” condition, which is not a standard configuration, consists of releasing the FIFOs and having the ideal-host CPUs performing back-to-back host read and write operations.

GENERATED EFFECTS

The effect of the above setup will result in:

- Back-to-Back Host READs. In this case, the “worst case” timing would occur where two Network WRITE operations will stretch the second host READ operation. The third and fourth host READ operations will toggle between network WRITE operations. This pattern will repeat itself, if the “worst case” scenario continues, every fifth host READ.

  (i.e. HR1, NW1, NW2, HR2, NW3, HR3, NW4, NW5, HR4...)

- Back-to-Back Host WRITEs. In this case, the “worst case” timing would be where three Network WRITE operations will stretch the second host WRITE operation. The third and fourth host WRITE are also held off by two Network WRITE operations each. This pattern will repeat itself every fifth host WRITE if the “worst case” scenario continues.

  (i.e. HW1, NW1, NW2, NW3, HW2, NW4, NW5, HW3, NW6, NW7, HW4...)

The maximum time for a host READ or WRITE would occur with host back-to-back HREQs within 30 ns, combined with multiple Network WRITEs. Port two would be given priority so that the network would not be delayed by host activity to the shared memory.

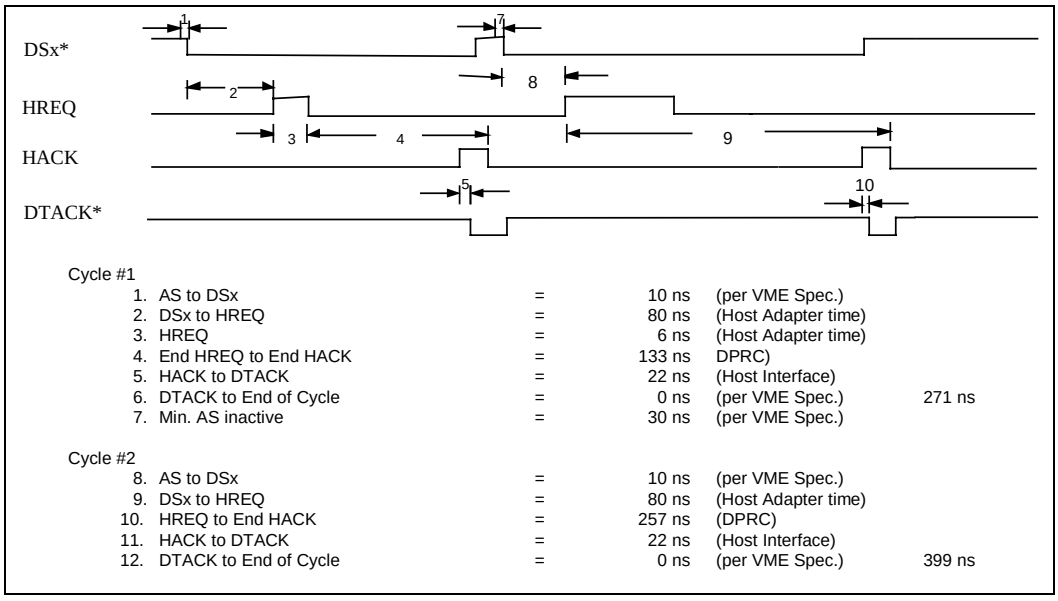
Maximum access times for the DPRC based on the “worst case” scenerio are shown in Table D-2 (These calculations do not include host timing):

Table D-2 Dual Port RAM Controller Maximum Access Times

Access Type	Cycle Time	Access Type	Cycle Time
Host READ	133 ns	Host WRITE	240 ns
2 Network WRITEs	+266 ns	3 Network WRITEs	+399 ns
+ Host READ	+133 ns	Host WRITE	+240 ns
Maximum Time	532 ns	Maximum Time	879 ns

D.5.3 Back-to-Back Host READs

Figure D-7 shows access timing for a Back-to-back VME read. All timing shown is based on the fastest possible VME master (according to VME specification C.1) driving the bus, and the maximum bandwidth on the **SCRAMNet+** Network ring.



Cycle #1

1.	AS to DSx	=	10 ns	(per VME Spec.)	
2.	DSx to HREQ	=	80 ns	(Host Adapter time)	
3.	HREQ	=	6 ns	(Host Adapter time)	
4.	End HREQ to End HACK	=	133 ns	DPRC)	
5.	HACK to DTACK	=	22 ns	(Host Interface)	
6.	DTACK to End of Cycle	=	0 ns	(per VME Spec.)	271 ns
7.	Min. AS inactive	=	30 ns	(per VME Spec.)	

Cycle #2

8.	AS to DSx	=	10 ns	(per VME Spec.)	
9.	DSx to HREQ	=	80 ns	(Host Adapter time)	
10.	HREQ to End HACK	=	257 ns	(DPRC)	
11.	HACK to DTACK	=	22 ns	(Host Interface)	
12.	DTACK to End of Cycle	=	0 ns	(per VME Spec.)	399 ns

Figure D-7 Back-to-Back Host READs

Figure D-8 illustrates how the DPRC must manage two inputs. The network WRITEs have the higher priority. Consequently, in this “worst case” scenerio, the Host READ cycle is stretched.

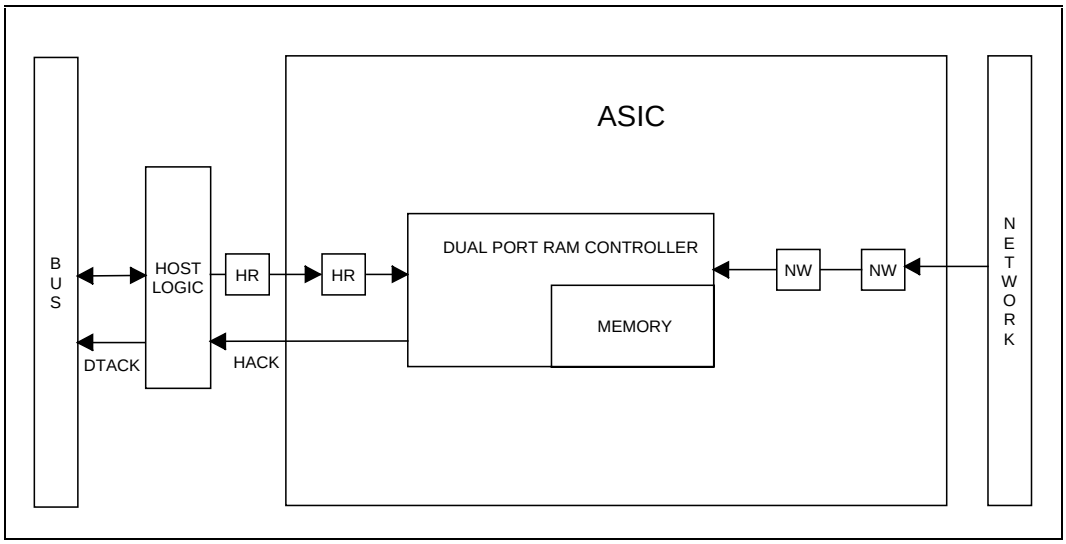


Figure D-8 Back-to-Back Host READs

FIRST CYCLE

1. AS, DSx fall - Start of Cycle (80 ns delay to decode address).
2. HREQ asserted - READ request to DPRC.
3. Host interface logic sleeps until HACK is received from DPRC (Maximum READ Time)
4. DPRC asserts HREQ_PEND Request accepted
   - DPRC is backed up with two Network WRITEs
   - DPRC performs READ for host.
5. HACK is issued to host interface logic.
6. DTACK is asserted - bus released.

7. DTACK is de-asserted; AS, DSx de-asserted - End of Cycle.

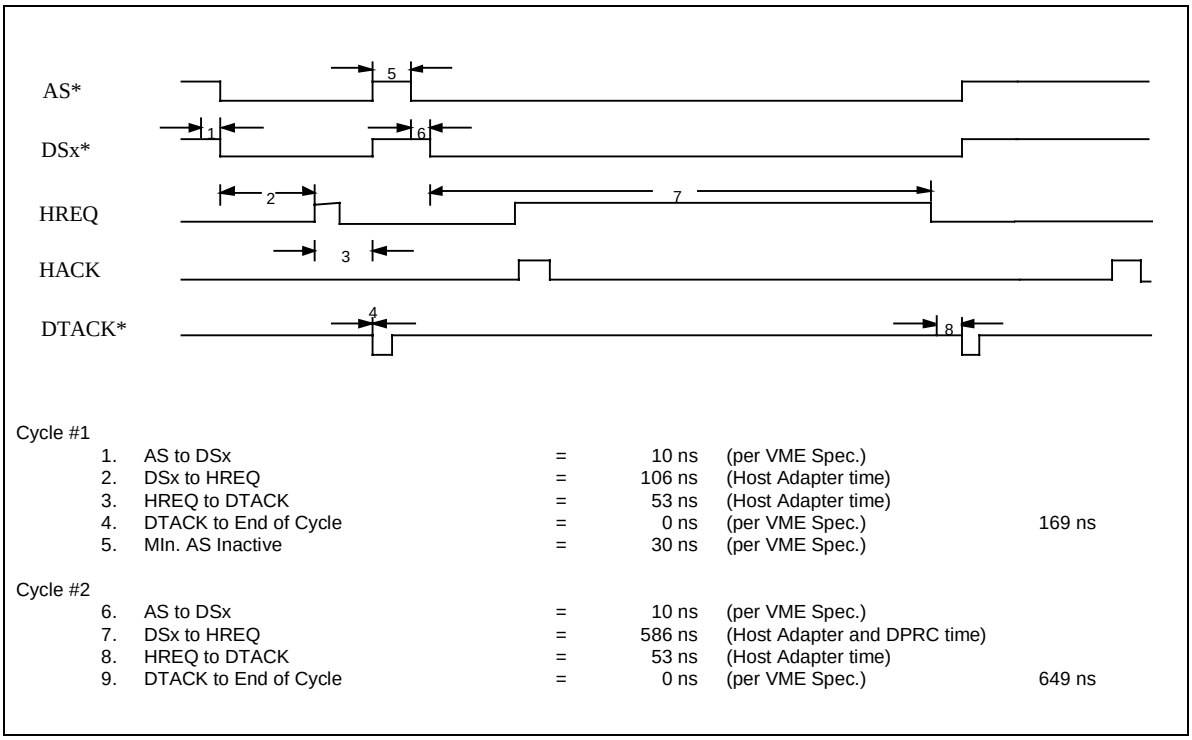
SECOND CYCLE

8. AS, DSx fall - Start of Cycle (80 ns delay to decode address).
9. HREQ asserted - READ request to DPRC.
10. Host interface logic sleeps until HACK is received from DPRC (Maximum READ Time) DPRC is processing two Network WRITEs.
11. HACK is issued to host interface logic; DTACK is de-asserted - bus released.
12. DTACK is asserted; AS, DSx de-asserted - End of Cycle.

D.5.4 Back-to-Back Host WRITEs

Figure D-9 shows access timing for two back-to-back VME WRITEs as fast as possible according to the VME specification revision C1. The first cycle shows the normal buffered WRITE cycle and the second cycle shows a stretched WRITE cycle (maximum hold). As in the VME READ, the host interface logic waits for the first Data Strobe to fall and then delays 106 ns to allow address decode and attain bus stability.

In the first VME cycle, the host logic raises a host WRITE request to the DPRC. The DPRC accepts the request, buffers the data, and issues a HACK to the Host Interface logic. The host interface logic responds by asserting DTACK 53 ns later. Under normal conditions this saves CPU cycles. By VME specification, as soon as the VME bus master detects DTACK on the bus, the AS and DSx may then be re-asserted to end the cycle.



Cycle #1					
1.	AS to DSx	=	10 ns	(per VME Spec.)	
2.	DSx to HREQ	=	106 ns	(Host Adapter time)	
3.	HREQ to DTACK	=	53 ns	(Host Adapter time)	
4.	DTACK to End of Cycle	=	0 ns	(per VME Spec.)	169 ns
5.	MIn. AS Inactive	=	30 ns	(per VME Spec.)	
Cycle #2					
6.	AS to DSx	=	10 ns	(per VME Spec.)	
7.	DSx to HREQ	=	586 ns	(Host Adapter and DPRC time)	
8.	HREQ to DTACK	=	53 ns	(Host Adapter time)	
9.	DTACK to End of Cycle	=	0 ns	(per VME Spec.)	649 ns

Figure D-9 Back-to-Back Host WRITEs

FIRST CYCLE

1. AS, DSx fall - Start of Cycle (106 ns delay to decode address).
2. HREQ asserted - WRITE request to DPRC
- DPRC asserts HREQ_PEND Request accepted
- DPRC buffers data
- DPRC asserts HACK.
3. Host linterface logic generates DTACK 53 ns later.
4. AS and DSx are de-asserted.
5. Minimum time AS is inactive is 30 ns.

SECOND CYCLE

The second cycle shows a second host WRITE is initiated 30 ns later according to minimum VME requirements. Under normal operating conditions it is unlikely the host will be able to turn around that quickly. An HREQ cannot be generated since the first one is still being serviced. Additionally, in this “worst case” scenerio, three network transactions must be processed before the previous host request is cleared.

Once the network transactions are cleared, the buffered data from the host WRITE can be written to shared memory which clears the WRITE buffer. This takes 240 ns from the request to the acknowledgement.

DPRC has not yet written to shared memory

6. AS, DSx fall - Start of second host cycle.
7. HREQ cannot be asserted since the previous request is still pending.
- DPRC receives 3 network WRITEs
- DPRC processes network WRITEs
- DPRC performs host WRITE to shared memory
8. HREQ_PEND de-asserted (No longer busy)
- DPRC asserts HREQ_PEND - WRITE request accepted
- DPRC buffers data
- DPRC asserts HACK
- Host Interface logic generates DTACK
9. AS and DSx are de-asserted

APPENDIX E
CONFIGURATION AIDS

TABLE OF CONTENTS

SCRAMNet+ CONTROL/STATUS REGISTERS REFERENCE SHEET

CSR 0

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

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

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

0	data intrpt vector
1	data intrpt vector
2	data intrpt vector
3	data intrpt vector
4	data intrpt vector
5	data intrpt vector
6	data intrpt vector
7	data intrpt vector
8	reserved
9	reserved
10	reserved
11	reserved
12	reserved
13	reserved
14	reserved
15	reserved

ACR

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

LED STATUS

G	INSERT
G	CARRIER DETECT

CSR 1 (READ RESET)

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

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

CSR 5

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

0	error intrpt vector
1	error intrpt vector
2	error intrpt vector
3	error intrpt vector
4	error intrpt vector
5	error intrpt vector
6	error intrpt vector
7	error intrpt vector
8	reserved
9	reserved
10	reserved
11	reserved
12	reserved
13	reserved
14	reserved
15	reserved

***** Write to CSR 1 to re-arm interrupts.

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

CSR 10

0	SM ACCESS ENB
1	reserved
2	reserved
3	reserved
4	reserved
5	reserved
6	reserved
7	reserved
8	reserved
9	reserved
10	reserved
11	reserved
12	SMA 12
13	SM ADDRESS
14	(LSW)
15	SMA 15

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

CSR 14

0	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
	reserved
15	reserved

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

CSR 11

0	SMA 16
1	
2	SHARED
3	MEMORY
4	ADDR
5	(MSW)
6	
7	SMA 23
8	reserved
9	reserved
10	reserved
11	reserved
12	reserved
13	reserved
14	reserved
15	reserved

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

CSR 15

0	not used
1	VME
2	IRQ
3	REGISTER
4	Levels
5	1-7
6	only
7	
8	reserved
9	reserved
10	reserved
11	reserved
12	reserved
13	reserved
14	reserved
15	reserved

CSR 16

0	HIPRO READ ENB
1	HIPRO READ ACR ENB
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 #

This page intentionally left blank.

APPENDIX F
ACRONYMS

ACK	Acknowledge or acknowledgment
ACR	Auxiliary Control RAM
AS#	Address Strobe number
BIST	Built-In Self Test
BLT	Block Transfer
CSR	Control/Status Register
DPRC	Dual Port RAM Controller
DS#	Data Strobe #
DTACK	Data Transfer Acknowledgment
DTB	Data Transfer Bus
ECM	Error Correction Mode
FDDI	Fiber Distributed Data Interface
FIFO	First-In, First-Out data buffer
FOB	Fiber Optic Bypass
GB	Gigabyte - 1024 Kilobytes (a billion bytes)
I/O PAGE	Block of address space which contains the CSRs for the processor and interface devices
HACK	Host Acknowledgment
HIPRO	High Performance
HREQ	Host Request
ISR	Interrupt Service Routine
KB	Kilobyte - 1024 bytes
LAN	Local Area Network
LSFR	Linear Feedback Shift Register
Longword	32-bit or 4-byte word
LSB	Least Significant Byte or Bit
LSHLW	Least Significant Half of a Longword
LSP	Least Significant Portion
LSW	Least Significant Word
MSB	Most Significant Byte or Bit
MSHLW	Most Significant Half of a Longword
MSP	Most Significant Portion
MSW	Most Significant Word
MTC	Missing Transition Code
MWT	SelBUS Memory Write Transfer cycle
NODE ID	Network Node Identification number (0-255)
ns, ms, μs	Nanosecond, millisecond, microsecond; respectively
PCB	Printed Circuit Board
PQFP	Plastic Quad Flat Pack
RAM	Random Access Memory
RMW	Read-Modify-Write
ROAK	Release On Interrupt Acknowledge cycle
SCM	Shared Common Memory
SCRAMNet	Shared Common Random Access Memory Network
SRAM	Static Random Access Memory
ST	AT&T Straight Tip type of fiber optic connector
UAT	Unaligned Transfer

This page intentionally left blank.

APPENDIX G
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.

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

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.

arbiter. A functional module that accepts bus requests from requester modules and grants control of the DTB to one requester at a time.

arbitration. The process of assigning control of the DTB to a requestor.

arbitration bus. One of the four buses provided by the backplane. This bus allows an arbiter module and several requester modules to coordinate use of the DTB.

arbitration cycle. An arbitration cycle begins when the arbiter senses a bus request. The arbiter grants the bus to a requester, which signals that the DTB is busy. The requester terminates the cycle by taking away the bus busy signal, which causes the arbiter to sample the bus requests again.

backplane. A printed circuit board (pcb) with 96-pin connectors and signal paths that bus the connector pins. Some systems have a single pcb, called the J1 backplane. It provides the signal paths needed for basic operations. Other systems also have a second pcb, called a J2 backplane. It provides the additional 96-pin connectors and signal paths needed for wider data and address transfers. Still others have a single pcb, called a J1/J2 backplane that provides the signal conductors and connectors of both the J1 and J2 backplanes.

backplane interconnect. An internal processor bus that allows I/O device controllers to communicate with main memory and the central processor, or they may be on a separate bus entirely. In the latter case, the I/O board enables and controls the communications between the I/O bus and the processor and memory.

backplane interface logic. Special logic that takes into account the characteristics of the backplane: its signal line impedance, propagation time, termination values, etc. The specification prescribes certain rules for the design of this logic based on the maximum length of the backplane and its maximum number of board slots.

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.) Then 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.

beginning of frame (BOF). A type of host-specific write to non-memory transfer which has the address and data for each transfer.

burst. A protocol where messages are transmitted without error correction to gain higher throughput.

burst+. Also **burst plus**. A variable packet size enhancement for the burst protocol. Maximum packet size may be set to either 256 bytes or 1024 bytes.

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.

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.

D08(O). (1) A slave that sends and receives data 8 bits at a time over D00-D07, or
(2) An interrupt handler that receives 8 bit status/ID over D00-D07, or
(3) An interrupter that sends 8-bit status/ID over D00-D07

D08(EO). (1) A master that sends or receives data 8 bits at a time over either D00-D07 or D08-D15,or
(2) A slave that sends and receives data 8 bits at a time over either D00-D07 or D08-D15

D16. (1) A master that sends and receives data 16 bits at a time over D00-D15, or
(2) A slave that sends and receives data 16 bits at a time over D00-D15, or
(3) An interrupt handler that receives 16-bit status/ID over D00-D15, or
(4) An interrupter that sends 16-bit status/ID over D00-D15

D32. (1) A master that sends and receives data 32 bits at a time over D00-D31, or
(2) A slave that sends and receives data 32 bits at a time over D00-D31, or

(3) An interrupt handler that receives 32-bit status/ID over D00-D31, or
(4) An interrupter that sends 32-bit

daisy-chain. A special type of signal line that is used to propagate a signal level from board to board, starting with the first slot and ending with the last slot. These are four bus grant daisy-chains and one interrupt acknowledge daisy-chain on the backplane.

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.

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.

IACK daisy-chain driver. A functional module that activates the interrupt acknowledge daisy-chain whenever an interrupt handler acknowledges an interrupt request. This daisy-chain ensures that only one interrupter will respond with its status/ID when more than one has generated an interrupt request on the same level.

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.

interrupt service routine (ISR). 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.

location monitor. A functional module that monitors data transfers over the DTB to detect accesses to the locations it has been assigned to watch. When an access occurs to one of these assigned locations, the location monitor generates an on-board signal.

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 tthat loops tthe 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 detection enabled.

platinum+. (Also platinum plus). A variable packet size enhancement for the platinum protocol. Maximum packet size may be set to either 256 bytes or 1024 bytes.

power monitor. A functional module that monitors the status of the primary power source to the system and signals when the power has strayed outside the limits required for reliable system operations. Since most systems are powered by an ac source, the power monitor is typically designed to detect drop-out or brown-out conditions on ax lines.

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.

programmed I/O (PIO) 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 overflow. A hardware failure condition reported when a second attempt to send a message around the network has failed due to another node corrupting data. This rare condition is most likely caused by dirty fiber optic cable ends, or perhaps the fiber cable is installed with too small a radius.

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.

serial clock driver. A functional module that provides a periodic timing signal that synchronizes the operation of the IEEE 1132* serial bus. Timing specifications for the serial clock driver of the IEEE 1132 are given in Appendix C. Two backplane signal lines are reserved for use by a serial bus. However, the protocols of the serial bus are completely independent of the IEEE 1014, and the inclusion of a serial bus is not a required feature of the IEEE 1014.

* IEEE 1132 may not yet be an approved standard.

shadow memory. See Auxiliary Control RAM (ACR)

shared memory (SM). **SCRAMNet** memory physically located on the network board. This dual-ported memory is accessable 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.

subrack. A rigid framework that provides mechanical support for boards inserted into the backplane, ensuring that the connectors mate properly and that adjacent boards do not contact each other. It also guides the cooling airflow through the system, and ensures that inserted boards do not disengage themselves from the backplane due to vibration or shock.

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

system controller board. A board that resides in slot 1 of the backplane and has a system clock driver, a DTB arbiter, and IACK daisy-chain driver, and a bus timer. Some also have a serial clock driver, a power monitor, or both.

time-out. Also network time-out. The time written to CSR5 that must elapse before a native meessage 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.

VME address space. The VME address space varies according to specific VME device and is identified as A16, A24, or A32 space. A32 is the largest address space; it allows up to 4 gigabytes of space using 32 bit addresses. A24 space uses 24 bit addresses, and A16 space uses 16 bit addresses.

VMEbus. A standard bus by which small computers and intelligent peripheral devices can be connected. The term VME stands for Versa Module Eurocard. This non-proprietary bus conforms to the American National IEEE Standard 1014 (ANSI/IEEE std 1014).

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.