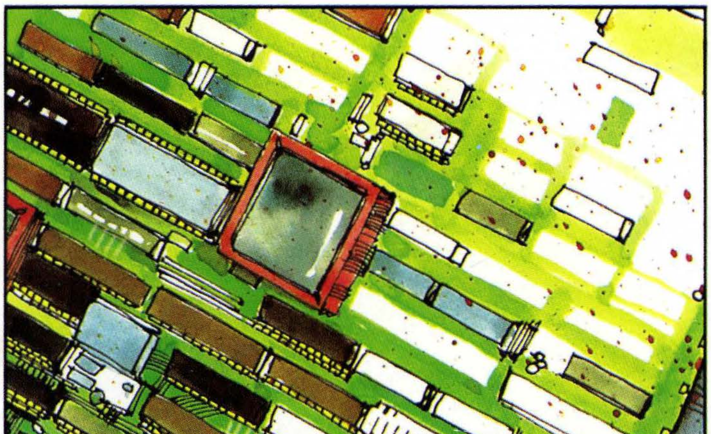
V/Ethernet 3207 Hawk

High-performance
VMEbus Ethernet
Communications
Controller



INTERPHASE
CORPORATION

V/Ethernet 3207 Hawk

High-performance
VMEbus Ethernet
Communications
Controller

V/Ethernet 3207 Hawk
LOCAL AREA NETWORK CONTROLLER FOR ETHERNET
USER'S GUIDE

Document Number................UG-0750-000-X0H
Revision...............................X0H
Follows Revision.......................X0G
Release Date.................November 20, 1987

COPYRIGHT NOTICE

FOR ASSISTANCE IN USING THE V/ETHERNET 3207 OR ANY OTHER INTERPHASE PRODUCT CALL:

INTERPHASE CUSTOMER SERVICE DEPARTMENT
(214) 919-9000

IN THE UNITED KINGDOM CALL:
(869) 321222

UPDATE

TO

V/ETHERNET 3207 USER'S GUIDE

(UG-0750-000-XOH, Rev. B)

March 28, 1989

The following 2-page update contains information which either supplements or supersedes information in the user's guide (UG-0750-000-XOH). The user's guide itself is located immediately following the update.

UPDATE

TO

V/Ethernet 3207 User's Guide

UG-0750-000-XOH, Rev. B

ADDENDUM - Released, March 28, 1989

The following information will be included in the next revision of the V/Ethernet 3207 Disk Controller User's Guide.

ACCESS TO THE DUAL PORT RAM

This section is a clarification of how jumpers JA4, JA8, JA9 and JA12 work in conjunction with each other to access the 256K dual port RAM.

The two functional blocks of the system that can access the 256K dual port RAM are the LANCE chip and VMEbus master. The following tables illustrates the jumpers to be set in order to detect parity errors. To configure the board properly, a selection for VMEbus and/or LANCE access generated errors must be made.1

VMEbus ACCESS GENERATED ERRORS

*Jumpers	Pins	Error Description
JA4 and JA9	1↔2	Parity error causes BERR
	2↔3	User detects parity error by polling BSR

LANCE ACCESS GENERATED ERRORS

*Jumpers	Pins	Error Description
JA8 and JA12	1↔2	Parity error on LANCE read causes LANCE "MERR" error and subsequent interrupt.
	2↔3	User detects parity error by polling BSR.

*Set both jumpers for proper configuration.

ERRATA

Pages 57 - 59. Factory Default jumper settings for JA4, JA8, JA9 and JA12 were miss labelled. The corrections are as follows:

JA4	1 - 2	DTACK timing for VMEbus initiated transfers is slow. This is for use when parity errors are reported to the VMEbus master as VMEbus error (BERR) (see JA8, JA12)--**Factory Default.**
	2 - 3	DTACK timing for VMEbus initiated transfers is fast. This is preferred when parity errors are sensed by polling board status register (BSR) (see JA8, JA12).
JA8	1 - 2	Enable special parity error reporting. LANCE cycle parity errors will cause a memory error (CSR0 bit 11) (See JA4, JA12)--**Factory Default.**
	2 - 3	Disable special parity error reporting (See JA4, JA12).
JA9	1 - 2	Enable VMEbus reporting of parity error using BERR signal--**Factory Default.**
	2 - 3	Disable VMEbus reporting of parity error using BERR signal. (NOTE: VMEbus BERR is asserted only on parity errors detected during V/Ethernet 3207 bus accesses. LANCE induced parity errors do not generate VMEbus BERRs.)
JA12	1 - 2	This is set if parity errors are being reported to the LANCE as memory errors (See JA4, JA8)--**Factory Default.**
	2 - 3	This setting is preferred if parity errors are sensed by polling board status register (See JA4, JA8).

Page 66. The title to Table A-2 was labelled incorrectly, it should read as followed:

"Table A-2: VMEbus Connector P2 Signal Descriptions (DASH 0 Only)*"

Page 67. The title to Table A-3 was labelled incorrectly, it should read as followed:

"Table A-3: VMEbus Connector P2 Signal Descriptions (DASH 1 Only)*"

TABLE OF CONTENTS

1 *INTRODUCTION TO THE V/Ethernet 3207 HAWK*

OVERVIEW 1
NETWORK PERFORMANCE FEATURES 1
LANCE Pipeline 2
OPERATION 3
SUMMARY OF FEATURES 4

2 *PROGRAMMING OF THE V/Ethernet 3207*

INTRODUCTION 7
ADDRESSING 8
COLLISION DETECTION 8
ERROR REPORTING/DIAGNOSTICS 9
BUFFER MANAGEMENT 9
INTERRUPT HANDLING 10
PROGRAMMING 11
Programming Sequence 12
Control and Status Registers 12
Accessing CSRs 13
Ring Access Mechanism in LANCE 13
SILO Operation 13
FOR FURTHER INFORMATION 14

3 *FUNCTIONAL DESCRIPTION OF THE V/Ethernet 3207*

INTRODUCTION 15
BOARD CONTROL REGISTER (BCR) 20
BOARD STATUS REGISTER (BSR) 22
INTERRUPT VECTOR REGISTER (IVR) 23
ETHERNET ADDRESS REGISTER (EAR) 24
The Serial Nonvolatile RAM 24
REGISTER DATA PORT (RDP) 29
REGISTER ADDRESS PORT (RAP) 29

TABLE OF CONTENTS (cont.)

3 *FUNCTIONAL DESCRIPTION OF THE V/Ethernet 3207 (Cont.)*

CONTROL AND STATUS REGISTERS 30
- Control and Status Register 0 (CSR0) 30
- Control and Status Register 1 (CSR1) 35
- Control and Status Register 2 (CSR2) 36
- Control and Status Register 3 (CSR3) 36

INITIALIZATION BLOCK 37
- Mode Register 38
- Physical Address Register (PADR) 40
- Logical Address Filter (LADRF) 41
- Receive Descriptor Ring Pointer (RDRP) 42
- Transmit Descriptor Ring Pointer (TDRP) 43

DESCRIPTOR RINGS 46
- Receive Message Descriptor 0 (RMD0) 46
- Receive Message Descriptor 1 (RMD1) 46
- Receive Message Descriptor 2 (RMD2) 48
- Receive Message Descriptor 3 (RMD3) 48
- Transmit Message Descriptor 0 (TMD0) 49
- Transmit Message Descriptor 1 (TMD1) 49
- Transmit Message Descriptor 2 (TMD2) 51
- Transmit Message Descriptor 3 (TMD3) 51

4 *INSTALLATION OF THE V/Ethernet 3207*

CABLING 53
- Dash 0 Version 53
- Dash 1 Version 53

SWITCH SETTINGS 53

JUMPER SETTINGS 57

INSTALLATION 59

TABLE OF CONTENTS (cont.)

5 *SPECIFICATIONS OF THE V/Ethernet 3207*

VMEbus SPECIFICATIONS 61
ENVIRONMENTAL SPECIFICATIONS 61
ELECTRICAL SPECIFICATIONS 61
MECHANICAL SPECIFICATIONS 61

APPENDICES

Appendix A

Connector Pin-outs 65
Connector P1 Signal Description 65
Connector P2 Signal Description (DASH 1) 66
Connector P2 Signal Discription (DASH 0) 67
Transceiver Connector Pin Assignments 68

Appendix B

V/Ethernet 3207 Board Layout for Artwork Rev. X0B 75
Jumper Configuration for Artwork Rev. X0B 76

GLOSSARY OF TERMS 71

INDEX 79

ILLUSTRATIONS

FIGURE 1 - CONTROL/STATUS & DATA STRUCTURES.17/18
FIGURE 2 - V/Ethernet 3207 BLOCK DIAGRAM.......19
FIGURE 3 - BOARD CONTROL REGISTER FORMAT.......20
FIGURE 4 - BOARD STATUS REGISTER FORMAT........22
FIGURE 5 - INTERRUPT VECTOR REGISTER FORMAT....23
FIGURE 6 - CONTROL STATUS REGISTER FORMAT......30
FIGURE 7 - MODE REGISTER FORMAT................38
FIGURE 8 - RMD 1 FORMAT........................46
FIGURE 9 - TMD 1 FORMAT........................49
FIGURE 10 - TMD 3 FORMAT.......................51
FIGURE 11 - BOARD LAYOUT.......................54
FIGURE 12 - SAMPLE SWITCH BLOCK SETTINGS........55
FIGURE 13 - SWITCH BLOCK RECORD................56

TABLES

TABLE 1 - PROGRAMMABLE REGISTERS...............15
TABLE 2 - ADDITIONAL LANCE REGISTERS...........16
TABLE 3 - INTERRUPT VECTOR SETTINGS............21
TABLE 4 - SERIAL NOVRAM INSTRUCTION SET........25
TABLE 5 - INITIALIZATION BLOCK.................37
TABLE A1 - VMEbus P1 CONNECTOR SIGNALS..........65
TABLE A2 - P2 CONNECTOR SIGNALS (DASH 1)........66
TABLE A3 - P2 CONNECTOR SIGNALS (DASH 0).......67
TABLE A4 - TRANSCEIVER CONNECTOR SIGNALS.......68

PREFACE

This User's Guide is intended to be a reference document for users who already have a general understanding of Ethernet and IEEE 802.3 compatible local area networks and a general knowledge of the VME standard. The V/Ethernet 3207 Hawk uses the Am7990 LANCE chip from Advanced Micro Devices to control many of the data transfer functions on the network. This document does not attempt to fully explain all of the LANCE. Instead, where more information may be necessary, the user may wish to consult the Am7990 LANCE and the Am7992B SIA (Serial Interface Adapter) data sheets from Advanced Micro Devices.

The following information is provided in this user's guide:

Section 1 - Introduction to the V/Ethernet 3207

Section 2 - Programming of the V/Ethernet 3207

Section 3 - Functional Description of the V/Ethernet 3207

Section 4 - Installation of the V/Ethernet 3207

Section 5 - Specifications of the V/Ethernet 3207

Appendix A - Connector Pin-outs

Glossary

Index

For more detailed explanations of specific operations or features of the V/Ethernet 3207 Hawk, call us at INTERPHASE. We would be happy to answer any questions!

INTERPHASE CORPORATION
APPLICATIONS ENGINEERING DEPARTMENT
(214) 350-9000

IN THE U.K. CALL (0296) 435661

SECTION 1

INTRODUCTION TO THE V/ETHERNET 3207 HAWK

OVERVIEW

The V/Ethernet 3207 Hawk is a VMEbus Local Area Network Controller for Ethernet and other IEEE 802.3 compatible networks. Designed to be used primarily in a single user, workstation environment, the V/Ethernet 3207 Hawk provides cost effective control of layers one and two of the seven-layer ISO/OSI model of data communications. Layers three and above of the protocol are referred to software running on the host CPU. While this minimal approach does use some host CPU computing power, it also creates several significant advantages over the direct effect of a low cost Ethernet connection.

This approach gives the system designer great latitude in implementing the optimum tradeoff between protocol sophistication and network efficiency. The continuing evolution of various implementations of the OSI model, plus the need to support existing TCP/IP, XNS, and ISO-IP low-level protocols as well as NFS and RFS applications, requires the flexibility and maintainability only found in host resident software. Indeed, any disadvantage of referring the upper layers of the protocol to the host becomes increasingly less significant as host CPUs become faster and more powerful. Thus, by not having a general purpose microprocessor on-board, the V/Ethernet 3207 Hawk provides both a degree of flexibility and a price performance ratio not found in other approaches.

NETWORK PERFORMANCE FEATURES

The V/Ethernet 3207 Hawk utilizes the combination of an Am7990 Local Area Network Controller for IEEE-802.3/Ethernet (LANCE), an Am7992B Serial Interface Adapter (SIA) and 256 Kbytes of dual ported RAM. The dual ported RAM is configured as transmit and receive ring buffers, with the number of buffers in each ring under programmatic control via Ring Descriptor Tables.

Ring Descriptor Tables also identify each buffer (transmit or receive) and indicate the buffer owner (the LANCE or the host CPU). Both the host CPU (via the VMEbus) and the LANCE have access to the full 256 Kbyte buffer. The host CPU sends data across the network by writing it into the RAM and receives data by reading it from the RAM. To the host CPU, the dual ported RAM appears as fast access, 16- or 32-bit wide, system memory. The inherent queuing mechanism of the transmit and receive ring buffers decouples system timing from network timing and vice versa. Thus, VMEbus timing cannot degrade network performance and neither can host CPU response time.

LANCE Pipeline

The Ethernet interface on the V/Ethernet 3207 Hawk consists of the LANCE chip and the SIA (Serial Interface Adapter) from AMD. The LANCE is a 10 Mbit/sec device, optimized to perform the link-level Ethernet protocol. This includes the CSMA/CD network access, on-chip DMA, error reporting, packet handling and local bus interface functions. The SIA provides Manchester II encoding/decoding of the serial bit stream and translation of the TTL I/O of the LANCE into differential I/O for the transceiver.

The operation of the SIA is controlled by the LANCE and the LANCE is controlled by the host. The LANCE is programmed through registers in Short I/O space of the V/Ethernet 3207 Hawk and through data and control structures in the 256 Kbytes of dual ported RAM. The LANCE has an internal 48-byte data SILO which it services from the dual port RAM (by filling from the RAM or emptying into the RAM). The LANCE does this with DMA bursts of up to eight words. In conventional architectures, the LANCE is tightly coupled either to the local bus (local memory) or to the system bus interface. The result is that it monopolizes the bus (either the local bus or the system bus, depending upon the architecture) during the entire eight word transfer. This takes a minimum of 4.8 microseconds even with zero wait state (200 nanosecond or better) memory.

The V/Ethernet 3207 Hawk resolves this problem by including a LANCE PIPELINE between the LANCE and the dual port memory. The LANCE PIPELINE enables the V/Ethernet 3207 Hawk to interleave VMEbus accesses to the dual port RAM (and refresh cycles) with LANCE accesses, even though the LANCE "thinks" it has total control of the memory for eight cycles. Thus, there is no critical timing between the VMEbus and the LANCE. And there is no monopolizing of the dual port RAM by the comparatively slow DMA of the LANCE. The data and control structures that reside in the 256 Kbytes of dual port RAM can be accessed by the host over the VMEbus at any time.

The data structures include ownership bits to indicate which device (host or LANCE) can write to a structure, so this does not change the software interface. But it does have a profound effect on the way the system performs because it decouples the LANCE and the internal bus (and thus it decouples the LANCE and the VMEbus), and it minimizes the bus bandwidth requirements of the LANCE. These are two of the important performance considerations that give the V/Ethernet 3207 Hawk such an outstanding price performance ratio.

OPERATION

The 256 Kbytes of on-board buffer RAM contain both the Transmit and Receive Data Buffers as well as the Transmit and Receive Ring Descriptor Tables for the LANCE chip. For each entry in either the Receive Ring Descriptor Table (RRDT) or the Transmit Ring Descriptor Table (TRDT) there is a corresponding Receive or Transmit Data Buffer. Each entry in either table includes an owner bit to indicate whether the data buffer is currently under control of the host CPU or the LANCE.

To send an Ethernet frame, or sequence of frames, the host simply puts the data into the first available buffer in the transmit ring (i.e. the first transmit buffer that is owned by the host). The host then writes the packet byte count to that buffer's entry in the TRDT and sets the owner bit in the TRDT to indicate to the LANCE chip that the buffer is ready to be transmitted.

When the LANCE finds that it owns this transmit buffer, it formats the data into a standard Ethernet packet and transmits it over the network. At the completion of the transmit operation, the LANCE chip clears the owner bit in the TRDT and generates an interrupt that gets passed to the host CPU. After reading a register in the LANCE chip to determine the cause of the interrupt, the host accesses the TRDT to determine the status of the transmit operation.

The low level details of a receive operation are similarly attended to by the hardware on the V/Ethernet 3207 Hawk. When a packet with this node's Ethernet Node Address is received over the network, data is transferred into the on-board buffer and an interrupt is generated on the VMEbus.

The host CPU reads a set of status registers in the LANCE chip to determine the cause of the interrupt. If the interrupt was caused by the completion of a packet reception, then the host accesses that buffer's entry in the RRDT to determine the status of the operation and to obtain the packet byte count for the received packet. Once the host CPU has moved the data, it resets the owner bit in the RRDT to free the buffer for the reception of another packet.

SUMMARY OF FEATURES

- Supports IEEE 802.3 Ethernet or Cheapernet Transceivers
- Network Access Independent of Host Response Time
- Network Performance Independent of VMEbus Timing
- Ethernet Node Address Stored in Nonvolatile Memory
- 256 Kbytes of Fast Access Dual Ported Memory accessible to both the LANCE and the VMEbus
- LANCE PIPELINE maximizes VMEbus performance
- Software Configurable Transmit Ring Buffer (from 2 to 128 buffers) for Automatic Transmit Queuing
- Software Configurable Receive Ring Buffer (from 2 to 128 buffers) for Automatic Receipt of Multiple Frames
- Software Programmable Interrupt Levels (1 of 7)
- Software programmable interrupt vector

- UNIX® Driver support
- Software Drivers on a Variety of Distribution Media
- Provides Flexibility in Implementing the Optimum Tradeoff between Protocol Sophistication and Network Efficiency
- Slave Data Transfers 16- and 32-bit data, 24- and 32-bit address, address modifiers: supervisory or non-privileged, standard or extended data access (3D, 39, 0D, 09)
- Slave Control/Status Transfers (Short I/O Space) 16-bit data, 16-bit address, address modifiers: supervisor or non-privileged, short access (2D, 29)
- Single Double-height VMEbus Board
- Superior Price Performance Ratio

UNIX is a registered trademark of AT&T Bell Laboratories.

SECTION 2

PROGRAMMING OF THE V/ETHERNET 3207 HAWK

INTRODUCTION

This section provides a summary of V/Ethernet 3207 programming considerations that will help you optimize your Ethernet activity when you write a driver. The Am7990 Ethernet controller chip (LANCE) is responsible for much of the V/Ethernet 3207 activity, as such, it is highly recommended that the user also read the LANCE technical manual. To receive a copy of it, call the Interphase Applications Engineering Department at (214) 350-9000.

The LANCE has four internal control and status register (CSR0, CSR1, CSR2, and CSR3) which are used for various functions such as loading the initialization block address, selecting different programming modes, and indicating status conditions. The host communicates with the LANCE during the initialization phase by loading the starting address of the Initialization Block into two of these internal control registers. During normal operation, the host communicates directly with the LANCE for demand transmission and periodically to read the status bits following interrupts. All other transfers to/from the dual ported RAM are automatically handled by the LANCE as DMA and are controlled indirectly by the host through the transmit and receive descriptor ring pointers.

Interrupts to the host are generated by the LANCE upon six different events:

- Completion of its initialization routine
- Reception of a packet
- Transmission of a packet
- Transmitter timeout error
- Missed packet
- Memory error

The cause of the interrupt is ascertained by reading CSR0. Bit six of CSR0 (INEA) enables or disables interrupts to the host. The method in which the V/Ethernet 3207 handles interrupts is explained in Section 3 of this user's guide.

The basic operation of the LANCE consists of two distinct modes: transmit and receive. In the transmit mode, the LANCE directly accesses data in the transmit buffer. It prefaces the data with a preamble and a sync pattern. It also calculates and appends a 32-bit CRC. This packet is then ready for transmission to the Am7992B SIA. On transmission, the first byte of data loads into the 48-byte SILO. The LANCE then begins to transmit preamble while simultaneously loading the rest of the packet into the SILO for transmission.

In the receive mode, packets are sent via the SIA to the LANCE. the packets are loaded into the 48-byte SILO for preparation of automatic downloading into the buffer memory. A CRC is calculated and compared with the CRC appended to the data packet. If the calculated CRC does not agree with the packet CRC, an error bit is set.

ADDRESSING

Packets can be received using three different destination addressing schemes: physical, logical, and promiscuous.

The physical scheme does a full comparison of the 48-bit destination address in the packet with the node address that was programmed into the LANCE during initialization.

There are two types of logical address. One is a group-type mask where the 48-bit address in the packet is put through a hash filter to map the 48-bit physical address into one of 64 logical groups. If any of the 64 groups have been preselected as the logical address, then the 48-bit address is stored in main memory. At this time, a look up is performed to compare the 48-bit incoming address with the prestored 48-bit logical address. This mode can be useful if sending packets to all of a particular type of device simultaneously (e.g., send a packet to all file servers or all printer servers). The second logical address is a broadcast address where all nodes on the network receive the packet.

The last receive mode of operation is the promiscuous mode. In this mode a node will accept all packets on the coax regardless of destination address.

COLLISION DETECTION

The Ethernet CSMA/CD network access algorithm is implemented completely within the LANCE. In addition to listening for a clear coax before transmitting, Ethernet handles collisions in a predetermined way. Should two or more transmitters attempt to access the coax at the same time, they will collide and the data on the coax will be garbled. The transmitting nodes listen while they transmit, detect the collision, and then continue to transmit for a predetermined length of time to "jam" the network and ensure that all nodes have recognized the collision. The transmitting nodes then delay a random amount of time according to the Ethernet "truncated binary backoff algorithm," so the colliding nodes do not try to repeatedly access the network at the same time. Up to 16 attempts to access the network are made by the LANCE before reporting back an error due to excessive collisions.

ERROR REPORTING/DIAGNOSTICS

Extensive error reporting is provided by the LANCE. Error conditions reported relate either to the network as a whole or to data packets. Network-related errors are recorded as flags in the CSRs and are examined by the host CPU following an interrupt. Network-related errors include:

- Babbling transmitter
  - Transmitter attempting to transmit more than 1518 data bytes
- Collision
  - Collision detection circuitry nonfunctional
- Missed packet
  - Insufficient buffer space
- Memory timeout
  - Memory response failure

Packet-related errors are written into descriptor entries corresponding to the packet. Packet-related errors include:

- CRC
  - Invalid data
- Framing
  - Packet did not end on a byte boundary
- Overflow/Underflow
  - Abnormal latency in servicing a DMA request
- Buffer
  - Insufficient buffer space

The LANCE performs several diagnostic routines which enhance the reliability and integrity of the system. These include a CRC logic check and two loopback modes (internal and external). Errors may be introduced into the system to check error detection logic. A Time Domain Reflectometer (TDR) is incorporated in the LANCE to help system designers locate faults in the Ethernet cable. Shorts and opens manifest themselves in reflections which are sensed by the TDR.

BUFFER MANAGEMENT

A key feature of the V/Ethernet 3207 is the flexibility and speed of communication between the LANCE and the host through the 256 kbytes of dual ported RAM. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings.

There are separate descriptor rings to describe transmit and receive operations. up to 128 tasks can be queued up on a descriptor ring awaiting execution by the LANCE. Each entry in a descriptor ring holds a pointer to a data memory buffer and an entry for the length of the data buffer. Data buffers can be chained or cascaded to handle a long packet in multiple data buffer areas. The LANCE searches the descriptor rings in a "lookahead" manner to determine the next empty buffer in order to chain buffers together or to handle back-to-back packets. As each buffer is filled, an "own" bit is reset, allowing the host processor to process the data in the buffer.

Buffer management is accomplished through message descriptors organized in ring structures in memory. Each message descriptor entry is four words long. There are two rings allocated for the LANCE: a Receive Ring and a Transmit Ring. The LANCE is capable of polling each ring for buffers to either empty or fill with packets to or from the channel. The LANCE is also capable of entering status information in the descriptor entry with which it is currently working.

The location and length of the descriptor rings are specified in the Initialization Block, accessed by the LANCE during the initialization procedure. Writing a '1' into the STRT bit of CSR0 will cause the LANCE to start accessing the descriptor rings and enable it to send and receive packets.

The LANCE communicates with a host device through the ring structures in memory. Each entry in the ring is either "owned" by the LANCE or the host. There is an ownership bit (OWN) in the message descriptor entry. Mutual exclusion is accomplished by a protocol which states that each device can only relinquish ownership of the descriptor entry to the other device; it can never take ownership and no device can change the state of any field in any entry after it has relinquished ownership.

INTERRUPT HANDLING

The V/Ethernet 3207 hardware can generate an interrupt request on any one of the seven possible VMEbus interrupt request lines. The interrupt level (i.e., interrupt request line) is selected by setting bits in the V/Ethernet 3207 Board Control Register (BCR). The interrupt request bits should, however, only be changed only when an interrupt request condition is not imminent. To ensure that the interrupt request condition does not occur, LANCE interrupts can be disabled by clearing the INEA bit in CSR0 (bit 6).

The LANCE can generate a VMEbus interrupt for any of six different occurrences: packet reception, packet transmission, initialization done, memory error, babbling transmitter, and missed packet. When any one of these situations occurs, the Interrupt Flag and the Interrupt Enable bit are both set in CSR0, the LANCE interrupt logic will assert the external interrupt output and the host will then initiates an IACK cycle on the VMEbus.

Typical interrupt handling procedures will clear the condition that originally causes an internal LANCE interrupt. But if a second interrupting condition occurs while the first condition is being serviced, the internal LANCE hardware does not ensure that every attention condition will generate a high-to-low transition on the LANCE interrupt output; thus not every interrupt will be serviced. Therefore, additional V/Ethernet 3207 hardware is used to handle simultaneous interrupts.

Between the LANCE interrupt output and the VMEbus request line is an enable, or mask, function. This interrupt mask function performs two tasks. First, after an interrupt request is presented to the VMEbus interrupt logic, subsequent interrupt requests are not presented because of the original LANCE attention situation. Second, if more than one interrupting condition occurs simultaneously within the LANCE, the V/Ethernet 3207 is able to present multiple interrupt requests to the VMEbus interrupt logic.

Once the LANCE asserts its interrupt output, a single request is passed on to the VMEbus request logic. After the interrupt handling procedures, including clearing of the LANCE internal interrupt source are complete, the V/Ethernet 3207 interrupt mask must be re-enabled before any subsequent interrupts from the LANCE are passed to the VMEbus interrupt request logic.

The Interrupt Enable bit in the Board Control Register resets the on-board interrupt circuitry after each interrupting condition. Therefore, this bit should be the last bit set before returning from the interrupt handling routine.

PROGRAMMING

The LANCE operates in an environment that includes close coupling with the 256 Kbytes of dual ported RAM and the host. The LANCE is programmed by a combination of registers and data structures resident within the LANCE and in memory. There are four Control and Status Registers (CSRs) within the LANCE which are programmed by the HOST device. Once enabled, the LANCE has the ability to access memory locations to acquire additional operating parameters.

The LANCE has the ability to do independent buffer management as well as transfer data packets to and from the Ethernet. There are three memory structures that reside within the dual ported RAM that are accessed by the LANCE.

The Initialization Block: it occupies 12 words in contiguous memory starting on a word boundary. It contains the operating parameters necessary for device operation. The Initialization Block is comprised of six entries:

- Mode of operation
- Physical address
- Logical address mask
- Location of Receive and Transmit Descriptor Rings
- Number of entries in Receive and Transmit Descriptor Rings

Receive and Transmit Descriptor Rings: these are two ring-structures, one each for incoming and outgoing packets. Each entry in each ring is four words long, and each entry must start on a longword boundary. The descriptor rings are comprised of the following information:

- Address of a data buffer
- Length of a data buffer
- Status information associated with the data buffer

Data Buffer: it is made up of contiguous portions of memory reserved for packet buffering. Data buffers may begin on arbitrary byte boundaries.

Programming Sequence

In general, the programming sequence of the LANCE may be summarized as:

1. Programming the LANCE CSRs by a host device to locate an Initialization Block in memory. The byte control, byte addressing, and address latch enable modes are defined here also.

2. The LANCE loads itself with the information contained in the Initialization Block.

3. The LANCE accesses the descriptor rings for packet handling.

CONTROL AND STATUS REGISTERS

There are four Control and Status Registers (CSRs) resident within the LANCE. The CSRs are accessed through two registers in Short I/O space, address port (RAP) and data port (RDP).

Accessing the Control and Status Registers

The CSRs are read (or written) in a two-step operation. The address of the CSR to be accessed is written into the address port (RAP). During a subsequent transaction, the data being read from (or written to) the data port (RDP) is read from (or written to) the CSR selected in the RAP.

RING ACCESS MECHANISM IN THE LANCE

Once the LANCE is initialized through the Initialization Block and started, the Host and the LANCE communicate via Transmit and Receive rings for packet transmission and reception.

These are two sets of pointers into the dual ported RAM (four 16-bit registers per set, corresponding to the four entries in each descriptor) maintained in the LANCE. The first set points to the current buffer, and they are working registers which are used for transferring the data for the packet. The second set contains the pointers to the next buffer in the ring which the LANCE obtained from the lookahead operation.

There are three types of ring access in the LANCE. The first type is when the LANCE polls the rings to own a buffer. The second type is a lookahead operation. The LANCE does a lookahead between the time that it is transferring data to and from the SILO; this lookahead is only done once. The third type of ring access is a data chaining operation. The LANCE tries to own the next descriptor in the ring when it clears the OWN bit for the current buffer.

SILO Operation

The SILO provides temporary buffer storage for data being transferred between the parallel bus I/O pins and the serial bus I/O pins of the LANCE. The capacity of the SILO is 48 bytes.

For transmit operations, data is loaded into the SILO under LANCE internal microprogram control. The SILO has to have at least 16 bytes available before the LANCE requests the dual port RAM access. The LANCE will start sending the preamble (if the line is idle) as soon as the first byte is loaded into the SILO from memory. Should the transmitter be required to back off, there could be up to 32 bytes of data in the SILO ready for transmission. Reception has priority over transmission during the time that the transmitter is backing off.

For receive operations, data is loaded into the SILO from the serial input shift register during reception. Data leaves the SILO under microprogram control. The LANCE microcode will wait until there are at least 16 bytes of data in the SILO before initiating a transfer into dual port RAM. The preamble (including sync) is not loaded into the SILO.

PARITY CHECKING

The V/Ethernet 3207 has a 32-bit wide internal DRAM array that has four possible sources of parity errors (i.e., one parity error for each byte of DRAM). When a parity error occurs, a bit in a latch is set and subsequently read through the Board Status Register. And by periodically polling the bits in the latch, parity errors can be easily reported.

Also, the V/Ethernet 3207 can be configured (via hardware jumpers) to modify the manner in which the DRAM read cycle is terminated. Depending on the source of the read cycle, the parity error is terminated in one of two ways. For a VMEbus-originated DRAM read cycle, the cycle is terminated by returning BERR instead of DTACK. For a LANCE-originated DRAM read cycle, the cycle is terminated by forcing a Memory Error (MERR).

One important aspect of parity error reporting that is easy to overlook is the time it takes to check parity. If the V/Ethernet 3207 is configured such that the DTACK and READY signals are blocked (refer to Section 4 for specific hardware jumper configuration), the generation of DTACK or READY is delayed for approximately 30 nanoseconds to allow the parity check process to complete. The reporting of parity errors to VMEbus and LANCE may be programmed independently of each other.

The PERR CLR bit in the Board Control Register is used to clear any parity errors, regardless of origination, that are "remembered" in the Board Status Register.

FOR FURTHER INFORMATION

The following topics are discussed in more detail in the Am7990 LANCE data sheet from Advanced Micro Devices, Inc.:

Transmit Ring Buffer Management
Receive Ring Buffer Management
Frame Formatting
Framing Errors (Dribbling Bits)
Collision Detection and Collision JAM
Receive-based Collisions
Transmit-based Collisions
Collision/Microcode Interaction
Time Domain Reflectometry
Heartbeat
Cyclic Redundancy Check (CRC)
Loopback
Serial Transmission
Serial Reception

The LANCE performs these functions on-board the V/Ethernet 3207 Hawk as outlined in the data sheet. Since the V/Ethernet 3207 is designed around the Am7990 Local Area Network Controller from Advanced Micro Devices, it is highly recommended that the user also read the LANCE technical manual. Call the Interphase Applications Engineering Department at (214) 350-9000 for a copy.

SECTION 3

FUNCTIONAL DESCRIPTION OF THE V/Ethernet 3207 HAWK

INTRODUCTION

The V/Ethernet 3207 Hawk is a VMEbus slave device whose control and status registers are accessed through five locations in the Hawk's 256 bytes of Short I/O space. Data and extended control/status information not available in the registers is passed through 256 Kbytes of dual ported RAM that appears as fast access system memory.

There are six registers that are accessed through five contiguous word locations in Short I/O space. They are listed below:

Table 1 - Programmable Registers

Register Name	Mnemonic	Offset into Short I/O	Type
Board Control Register	(BCR)	0	Write Only
Board Status Register	(BSR)	0	Read Only
Interrupt Vector Register	(IVR)	2	Write Only
Register Data Port *	(RDP)	4	Read/Write
Register Address Port *	(RAP)	6	Read/Write
Ethernet Address Register	(EAR)	8	Read/Write

* these are LANCE registers

There are four additional registers inside the LANCE itself that are accessed by setting the register number in the RAP and then reading or writing to the RDP. They are as follows:

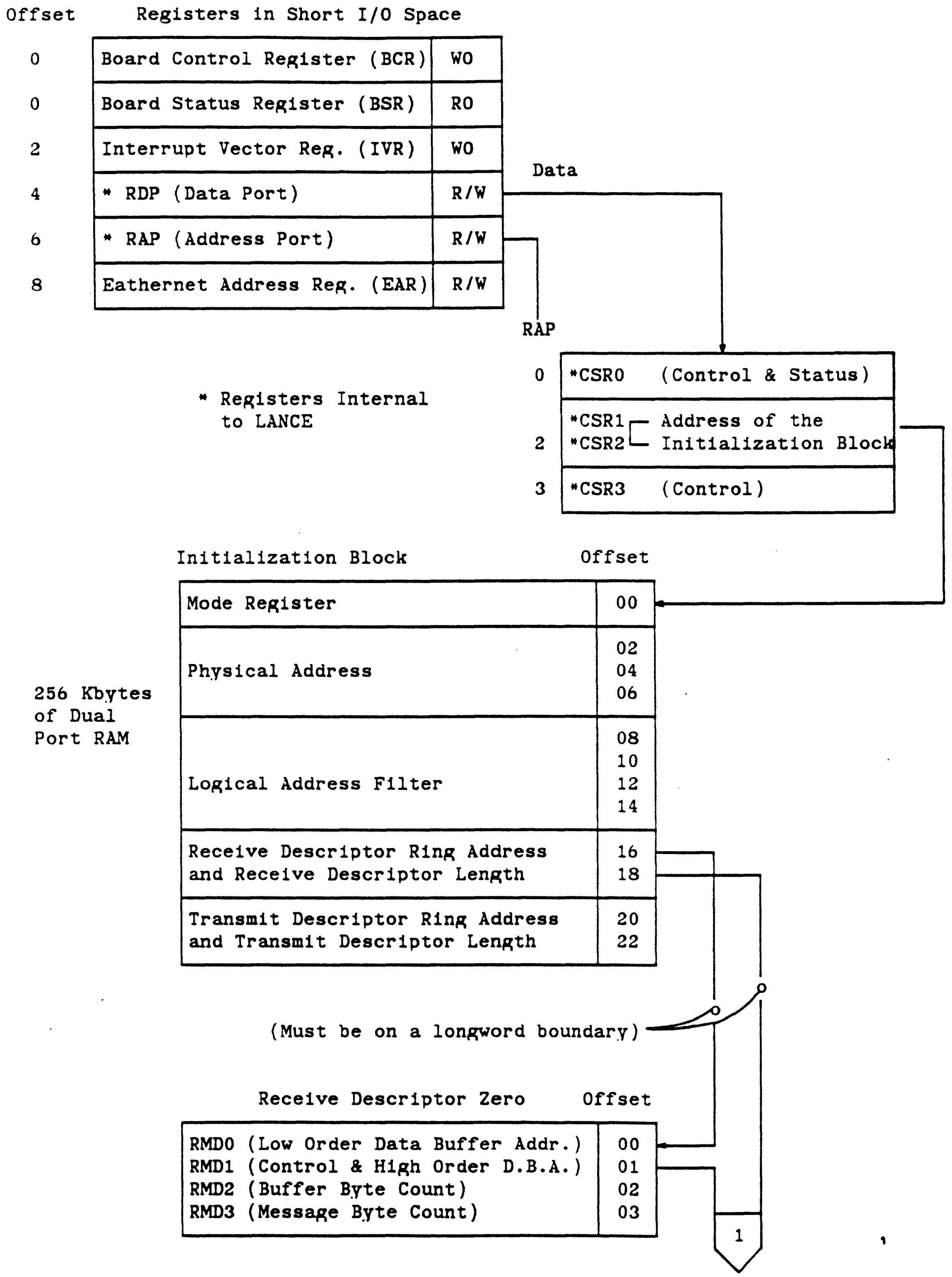
Table 2 - Additional LANCE Registers

Register Name/Mnemonic - register usage	Register Number
CSR0 - (control/status)	0
CSR1 - (initialization block address)	1
CSR2 - (initialization block address)	2
CSR3 - (control)	3

The remainder of the control/status information and all of the data structures reside within the 256 Kbytes of dual port, on-board RAM. The Initialization Block Address in CSR1 and CSR2 point to the 24-byte Initialization Block in RAM. It in turn points to a Receive Ring Descriptor Table (RRDT) and to a Transmit Ring Descriptor Table (TRDT). Individual entries in the RRDT then point to individual Receive Data Buffers and entries in the TRDT point to Transmit Data Buffers. Once initialized, the LANCE maintains the two sets of Ring Descriptor Tables that keep track of where the data buffers are and what their status (ownership) is.

Figure 1 is a Control/Status and Data Structure Diagram showing the relationship of the various hardware registers and the data structures within the 256 Kbytes of dual ported RAM.

Figure 2 is a simplified Functional Block Diagram of the V/Ethernet 3207 Hawk. It shows the relationship of the hardware blocks and the major address and data paths.



Offset	Registers in Short I/O Space	
0	Board Control Register (BCR)	WO
0	Board Status Register (BSR)	RO
2	Interrupt Vector Reg. (IVR)	WO
4	* RDP (Data Port)	R/W
6	* RAP (Address Port)	R/W
8	Eathernet Address Reg. (EAR)	R/W

RAP	
0	*CSR0 (Control & Status)
 2	*CSR1 Address of the *CSR2 Initialization Block
3	*CSR3 (Control)

* Registers Internal to LANCE

256 Kbytes of Dual Port RAM

Initialization Block	Offset
Mode Register	00
Physical Address	02 04 06
Logical Address Filter	08 10 12 14
Receive Descriptor Ring Address and Receive Descriptor Length	16 18
Transmit Descriptor Ring Address and Transmit Descriptor Length	20 22

Receive Descriptor Zero	Offset
RMD0 (Low Order Data Buffer Addr.)	00
RMD1 (Control & High Order D.B.A.)	01
RMD2 (Buffer Byte Count)	02
RMD3 (Message Byte Count)	03

Figure 1. V/Ethernet 3207 Data Structures
(Continued on next page.)

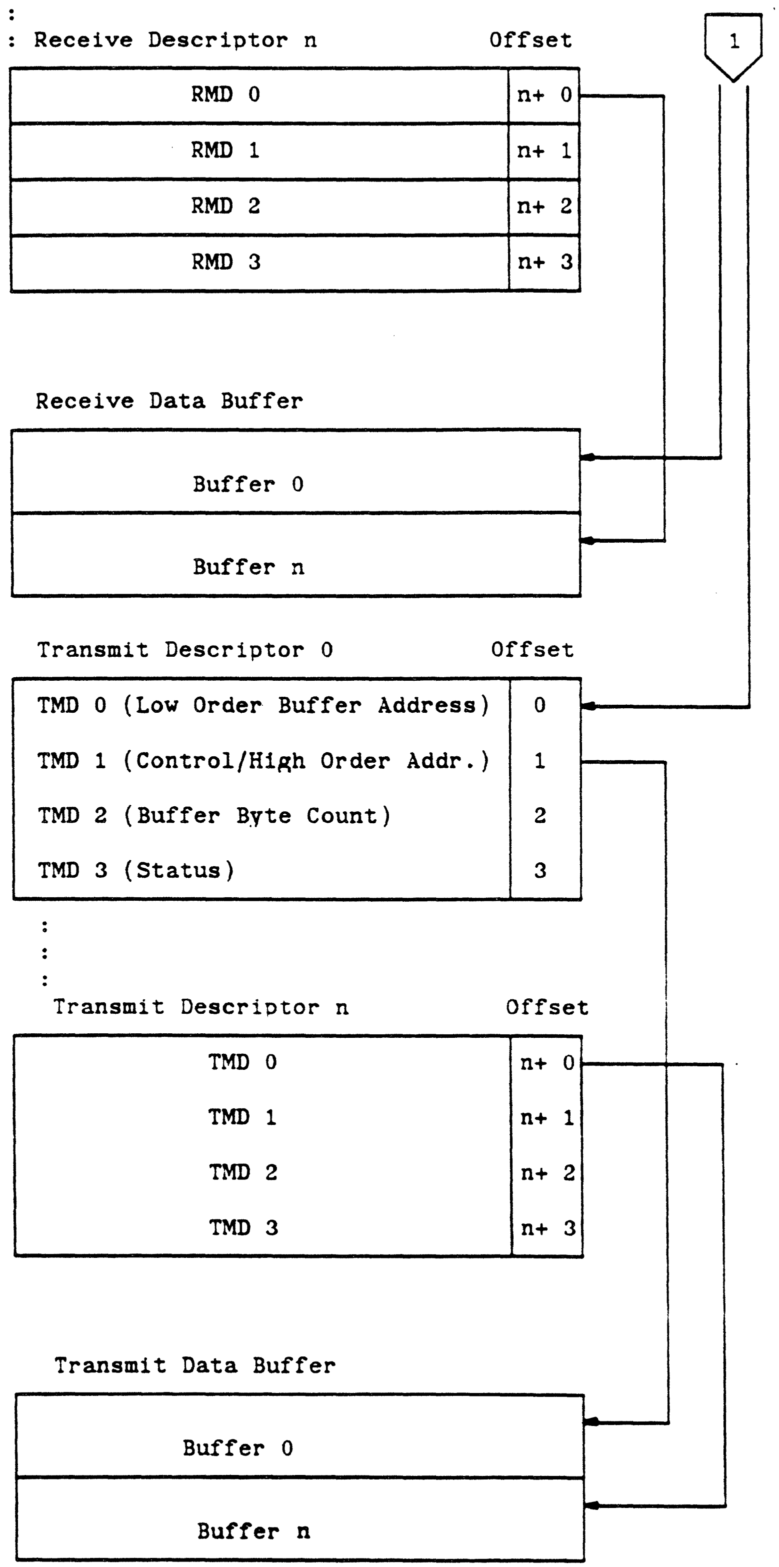
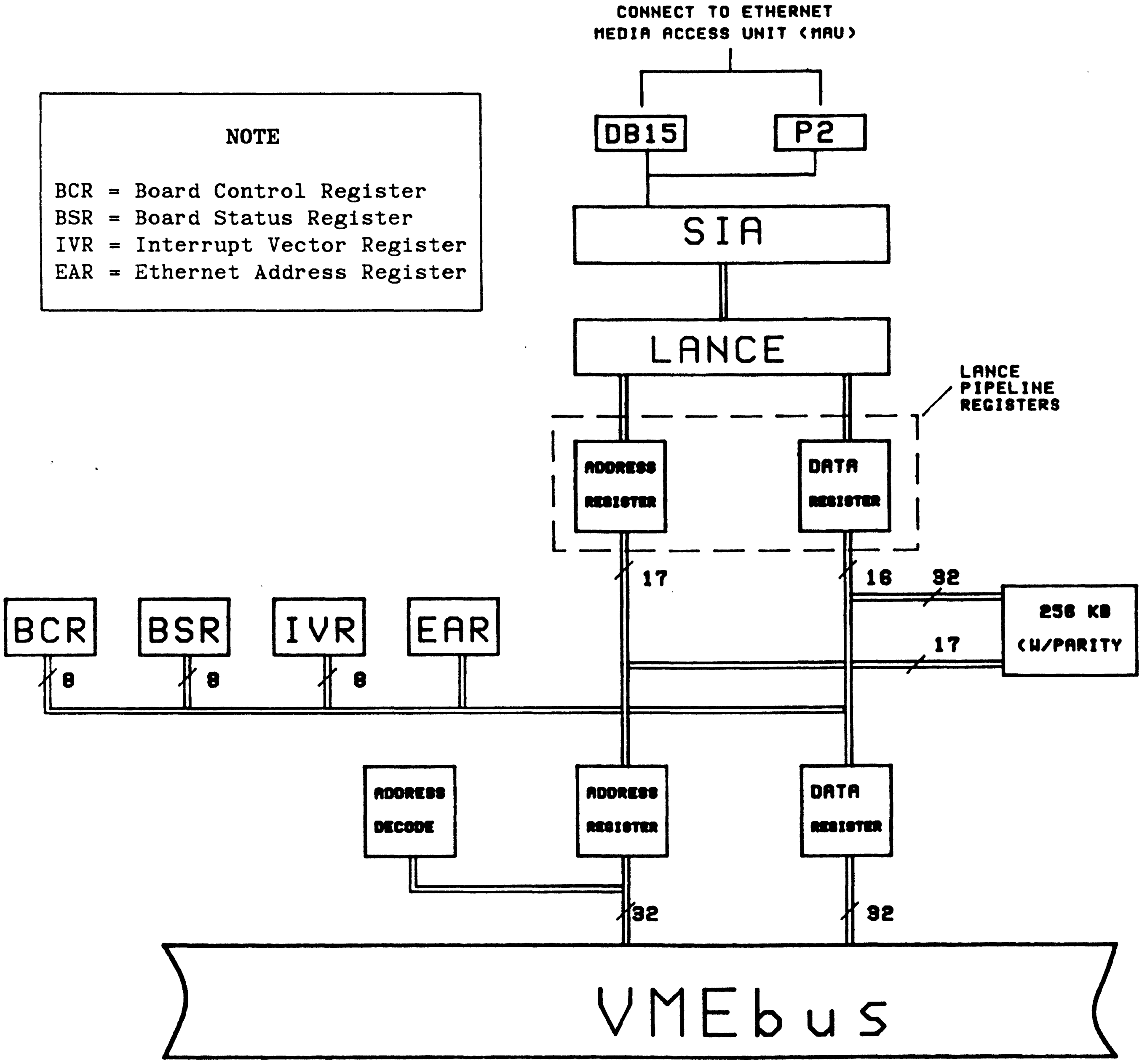


Figure 1. V/Ethernet 3207 Hawk Data Structures (cont.)



NOTE

BCR = Board Control Register
BSR = Board Status Register
IVR = Interrupt Vector Register
EAR = Ethernet Address Register

Figure 2. V/Ethernet 3207 Simplified Block Diagram

BOARD CONTROL REGISTER - BCR - (Write Only)

The Board Control Register contains the bits that are used by the host CPU to initiate commands to the V/Ethernet 3207. It also contains the bits required to indicate the condition of the V/Ethernet 3207 to the CPU. The Board Control Register is a write only register in Short I/O space with an address offset of 000. The format of the Board Control Register is shown in Figure 3 below.

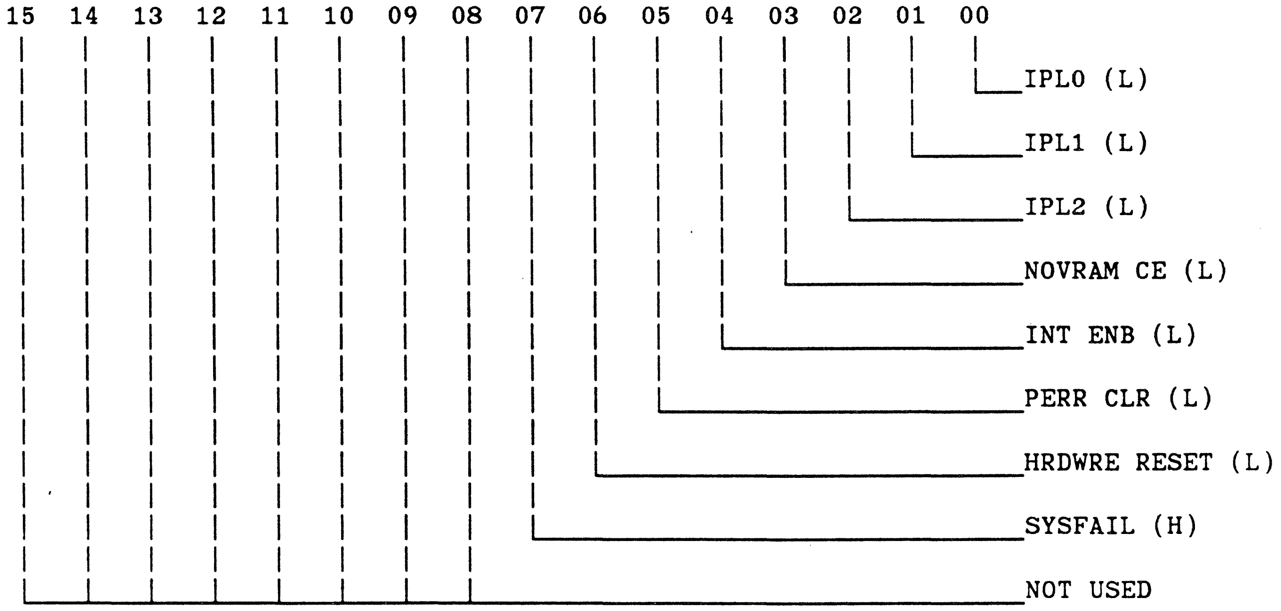


Figure 3. Board Control Register Format

Bits 15-8: Not Used.

Bit 7: **SYSFAIL** - SYSTEM FAILURE: This bit is used to drive the SYSFAIL* signal on the VMEbus (if enabled by hardware jumper JA 11) during a hardware or software reset. If this bit is set to'1', the VMEbus SYSFAIL line is set and the status indicator on the front panel is red, if this bit is cleared to '0', the SYSFAIL line is cleared and the status indicator is green. The V/Ethernet 3207 sets this bit to '1' when System Reset or Hardware Reset is performed.

Bit 6: **HRDWRE RESET** - HARDWARE RESET: When this bit is cleared to '0', it causes a hardware reset of the V/Ethernet 3207. This is the equivalent of driving the SYSRESET signal on the VMEbus. Once this bit is cleared, the board will be busy for 15 microseconds. There is no need to set this bit to '1' once it is cleared.

Bit 5: **PERR CLR** - PARITY ERROR CLEAR: This bit is cleared to '0' to reset the parity error bits in the BSR after a parity error has been detected. In addition, this bit should be cleared to '0' during initialization to clear the parity bits in the BSR. There is no need to set this bit to '1' once it is cleared.

Bit 4: **INT ENB*** - INTERRUPT ENABLE: This bit is cleared to '0' to reset the on-board interrupt circuitry after an interrupting condition. This enables the next interrupt to be posted. This bit should be cleared as the last action of the interrupt service routine. In addition, this bit should be cleared to '0' during initialization. There is no need to set this bit to '1' once it is cleared.

Bit 3: **NOVRAM CE*** - NONVOLATILE RAM CHIP ENABLE: This bit must be cleared to '0' when reading from or writing to the V/Ethernet 3207 nonvolatile RAM. It must remain clear ('0') during the entire read or write cycle. Upon completion of the read/write, this bit must be set to '1'. The V/Ethernet 3207 sets this bit to '1' when System Reset or Hardware Reset is performed..

Bits 2-0: **IPL0-2*** - INTERRUPT VECTOR LEVEL: These three bits set the bus interrupt level (level one through seven). Table 3 below shows the bit settings and the corresponding interrupt level that is indicated.

Table 3 - Interrupt Vector Settings

	IPL2	IPL1	IPL0
Level 7	0	0	0
Level 6	0	0	1
Level 5	0	1	0
Level 4	0	1	1
Level 3	1	0	0
Level 2	1	0	1
Level 1	1	1	0

Level 0 (1,1,1) indicates that no interrupts are specified. The V/Ethernet 3207 sets IPLO, IPL1 and IPL2 to '1' (interrupt level zero) when System Reset or Hardware Reset is performed.

BOARD STATUS REGISTER - BSR - (Read Only)

The Board Status Register is used to indicate when a parity error has occurred in the on-board RAM. This register is a read only register in Short I/O space with an address offset of 000. The following figure illustrates the format of the Board Status Register.

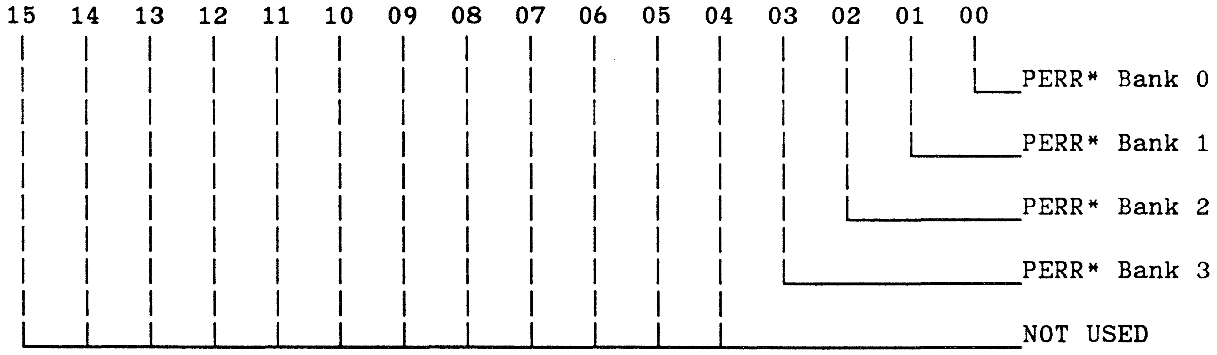


Figure 4. Board Status Register Format

Bits 15-4: Not Used.

Bits 3-0: **PERR - PARITY ERROR.** These are active low signals. A ‘1’ indicates no error and a ‘0’ indicates an error.

NOTE

The Board Status Register indicates any parity errors that occur on the V/Ethernet 3207. It cannot differentiate between parity errors that occur during a VMEbus access and parity errors that occur as a result of a LANCE access.

INTERRUPT VECTOR REGISTER - IVR - (Write Only)

The Interrupt Vector Register contains the interrupt vector used to respond to the VMEbus Interrupt Acknowledge cycle. It consists of a unique 8-bit user defined vector. This register is a write only register in Short I/O space with an address offset of 002 (hex). The following figure illustrates the format of the Interrupt Vector Register.

```
15 to 08    07 to 00
   |           |
   |           |____________IVEC
   |
   |________________________RESERVED
```

Figure 5. Interrupt Vector Register Format

Bits 15-8: Reserved (Clear to '0'.)

Bits 7-0: **IVEC - INTERRUPT VECTOR.**

ETHERNET ADDRESS REGISTER - EAR - (Read/Write)

The Ethernet Address Register is used for reading and writing to the 256-bit (sixteen 16-bit words) nonvolatile RAM. The EAR is a one-word register with the least significant bit (D0) reserved for reading and writing to the 256-bit serial nonvolatile RAM. The EAR is a read/write register in Short I/O space with an address offset of 008 (hex).

The serial 256-bit nonvolatile RAM is composed of a 16 X 16 static RAM, overlaid bit for bit with a nonvolatile EEPROM array. The serial nonvolatile RAM is controlled through an 8-bit instruction register that is also accessed serially through the Ethernet Address Register.

The Serial Nonvolatile RAM

To enable the serial nonvolatile RAM for any data accesses, bit 3 in the Board Control Register (NOVRAM CE - active low) must be set to a '0'. And then before a write operation can be performed, the Write Enable Latch (internal to the nonvolatile RAM) must be set by shifting in a Set Write Enable Latch (WREN) instruction. Then the Write Data into RAM (WRITE) instruction is shifted in, one bit at a time, starting with the MSB. Finally, the 16 bits of data are shifted in. (Data should also be shifted in the MSB first.)

After the data is shifted into the RAM, a Store RAM Data EEPROM (STO) instruction must be shifted in to move the data from the RAM to the EEPROM. This complete cycle must be repeated three times to store the 48-bit Ethernet Node Address in the EEPROM. After the last WRITE operation has been performed, a Reset Write Enable Latch (WRDS) instruction must be shifted in, and then the NOVRAM CE bit in the BCR must be set to a '1'. This disables further write operations.

It should be noted that the Ethernet Address Register is preprogrammed at the factory with an Ethernet node address (assigned by Xerox), so unless the user wants to change that address, there is no need to write to this register.

Once the Ethernet Node Address is in the EEPROM, it will be retained for more than 100 years, or until it is overwritten. To read any data back from the 256-bit serial nonvolatile RAM, the NOVRAM CE bit in the BCR must be cleared to '0'. The data must then be read from the EEPROM to the RAM with a Recall EEPROM Data into RAM (RCL) instruction. Once the data is in the RAM, it can then be shifted out, one bit at a time, through the EAR with a Read Data from RAM (READ) instruction. After the last READ operation has been performed, the NOVRAM CE bit in the BCR must be set to a '1'. This disables further READ operations.

Because of a data/shift pulse anomaly, the data does not shift out in the same order that it was shifted in. If data was shifted in MSB first, then the data will be shifted out correctly ordered but rotated to the left one position (see below).

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

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

It is the responsibility of the host CPU not only to shift the data in and out, but also to order the data correctly before attempting to use it.

Table 4: Serial Nonvolatile RAM Instruction Set

Instruction	Format MSB LSB	Operation
WRDS	0XXXX111	Reset Write Enable Latch (Disables WRITE and STO)
STO	0XXXX110	Store RAM Data in EEPROM
WRITE	0AAAA100	Write Data into RAM Address AAAA
WREN	0XXXX011	Set Write Enable Latch (Enables WRITE and STO)
RCL	0XXXX010	Recall EEPROM Data into RAM
READ	0AAAA00X	Read Data from RAM Address AAAA

X = Don't Care
A = Address Bit

Note: The NOVRAM instruction data is sent by shifting the MSB first.

The V/Ethernet 3207 NOVRAM occupies 256 bytes. The first three words (48 bits) are used for the Physical Ethernet Address. (Bit zero of the physical address is always forced to '0'). The following four words (64 bits) are used for the Logical Ethernet Address. (Bit zero of the logical address is always forced to '1'). The remaining nine words (144 bits) of the NOVRAM space is reserved.

The sample code on the following three pages illustrates a typical implementation of NOVRAM access using the "C" programming language.

```
#define HBC_NOV_RAM             0x0008          /* novram enable bit in BLCSR */

/*
 * Novram "Opcode" masks (inverted due to 74AS646's)
 */
#define WRD_NOVRAM              0xE0            /* Reset write enable latch   */
#define STO_NOVRAM              0x60            /* Store ram to eeprom        */
#define SLP_NOVRAM              0xA0            /* Novram into low power mode */
#define WRT_NOVRAM              0x20            /* Writes word from location x*/
#define WRE_NOVRAM              0xC0            /* Set write enable latch     */
#define RCL_NOVRAM              0x40            /* Recall eeprom data into ram*/
#define RDE_NOVRAM              0x00            /* Reads word from location x */

/*
 * BOARD Register Access Macros
 */
#define WRT_BDCSR_OR(x)             hwk->hwk_hwkaddr->board_csr=board_csr_image| =x
#define WRT_BDCSR_AND(x)            hwk->hwk_hwkaddr->board_csr=board_csr_image&=x
#define DELAY(x)                    for (counter = x; counter >0 ;counter--)
#define NOVRAM_CMD(c,a)             for(i=0;i<8;hwk_novram=((c| (a<<1))>>i),i++);
DELAY(HW_DELAY)


/*
 * reset write enable latch (disables writes and stores)
 */
nr_wtdis()
{
    WRT_BDCSR_AND  ( ~HBC_NOV_RAM );            /* set novram chip enable bit  */
    NOVRAM_CMD(WRD_NOVRAM,0);                   /* send opcode                 */
    WRT_BDCSR_OR   ( HBC_NOV_RAM );             /* clear novram chip enable bit*/
    return(OK);
}

/*
 * store novram ram data into eeprom
 */
nr_store()
{
    WRT_BDCSR_AND  ( ~HBC_NOV_RAM );            /* set novram chip enable bit  */
    NOVRAM_CMD(STO_NOVRAM,0);                   /* send opcode                 */
    WRT_BDCSR_OR   ( HBC_NOV_RAM );             /* clear novram chip enable bit*/
    return(OK);
}
```

```
/*
 *  places novram into SLEEP lower power mode
 */
nr_sleep()
{
    WRT_BDCSR_AND  ( ~HBC_NOV_RAM );        /* set novram chip enable bit  */
    NOVRAM_CMD(SLP_NOVRAM,0);               /* send opcode                 */
    WRT_BDCSR_OR  ( HBC_NOV_RAM );          /* clear novram chip enable bit*/
    return(OK);
}

/*
 * writes 16 bits to the novram at location x
 */
nr_write(addr,data)

register u_short addr,data;
{
    register unsigned short val = 0,
                            mask = 0x8000;
    register unsigned int   wbit;

        /*
         *  Write novram
         */
        WRT_BDCSR_AND  ( ~HBC_NOV_RAM );        /* enable novram with image*/
        NOVRAM_CMD     ( WRT_NOVRAM, addr );    /* send write novram opcode*/
        /*
         *  Put 16 bits one at a time to the novram
         */
        for (wbit=0; wbit<16; wbit++) (         /* put 16 bits             */
                (mask&data) ? (hwk_novram=1) : (hwk_novram=0);
                mask = mask >>1;                /* one bit at a time       */
                DELAY( HW_DELAY );
        }                                       /* until done              */
        WRT_BDCSR_OR   ( HBC_NOV_RAM );         /* disable novram with image*
}
/*
 *  set write enable latch (enables writes and stores)
 */
```

```
nr_wten()
{
    WRT_BDCSR_AND  ( ~HBC_NOV_RAM );        /* set novram chip enable bit  */
    NOVRAM_CMD(WRE_NOVRAM,0);               /* send opcode                 */
    WRT_BDCSR_OR   ( HBC_NOV_RAM );         /* clear novram chip enable bit*/
    return(OK);
}

/*
 *  recall eeprom data into ram
 */
nr_recall()
{
    WRT_BDCSR_AND  ( ~HBC_NOV_RAM );        /* set novram chip enable bit  */
    NOVRAM_CMD(RCL_NOVRAM,0);               /* send opcode                 */
    WRT_BDCSR_OR   ( HBC_NOV_RAM );         /* clear novram chip enable bit*/
    return(OK);
}

/*
 * reads 16 bits from the novram at location x
 */
nr_read(addr)

register u_short addr;
{
        register unsigned short val = 0, mask = 0x04000;
        register unsigned int wbit;

        WRT_BDCSR_AND ( •HBC_NOV_RAM );         /* enable novram with image*/
        NOVRAM_CMD    ( RDE_NOVRAM, addr );     /* send read opcode        */
        /*
         *  Get 16 bits one at a time from the novram
         *  We wrote bit 15 first, so we read bit 14 first, bit 15 last.
         */
        for (wbit=0; wbit<15; wbit++) {         /* get bits 14 thru 0      */
                                                /* and build data          */
                (hwk_novram & 0x01) ? (val=(val| mask)):(val=(val&(~mask)));
                mask = mask>>1;
                DELAY( HW_DELAY );
        }                                       /* until done              */
                                                /* get bit 15              */
        (hwk_novram & 0x01) ? (val=(val| 0x8000)):(val=(val&0x7FFF));
        DELAY         ( HW_DELAY );
        WRT_BDCSR_OR  ( HBC_NOV_RAM );          /* disable novram with image*/
        return (val);                           /* return word value read   */
}
```

REGISTER DATA PORT - RDP - (Read/Write)

The Register Data Port is used to read or write data to or from the four Control/Status Register (CSR0--CSR3) in the LANCE. It is a 16-bit Read/Write register in Short I/O space with an address offset of 004 (hex).

The particular CSR connected to the RDP is determined by the register number in the Register Address Port (RAP). CSR1, CSR2, and CSR3 are accessible only when the STOP bit of CSR0 (bit 02) is set. If the STOP bit is not set while attempting to Read CSR1, CSR2, CSR3, undefined data will be returned. If it is not set on a Write, the operation will be ignored. RAP is cleared by a HRDWRE RESET or a VMEbus SYSRESET.

REGISTER ADDRESS PORT - RAP - (Read/Write)

The Register Address Port is used to specify which of the four Control/Status Registers (CSRs) in the LANCE is connected to the RDP. The RAP is a Read/Write register in Short I/O space with an address offset of 006 (hex)

Bits 15 through two of the Register Address Port are reserved and must be set to zero. Bits one and zero are used to select the CSR number which is accessed through the RDP. The CSRs are selected as follows:

Bit 1	Bit 0	CSR Selected
0	0	CSR0
0	1	CSR1
1	0	CSR2
1	1	CSR3

To access a CSR, its address (number) must be written into the RAP during a bus slave transaction. During a subsequent bus slave transaction, the data being read from (or written to) the RDP is read from (or written to) the CSR selected in the RAP. Once written, the address in RAP remains unchanged until it is rewritten or cleared by HRDWRE RESET or VMEbus SYSRESET.

NOTE

All four Control and Status Registers in the LANCE (CSR0, CSR1, CSR2, CSR3) are Read/Write.

CONTROL AND STATUS REGISTERS

Control and Status Register Zero - CSR0 (Read/Write)

Each bit of CSR0 is delineated in the following subsection. The format of the register is detailed in Figure 6 below.

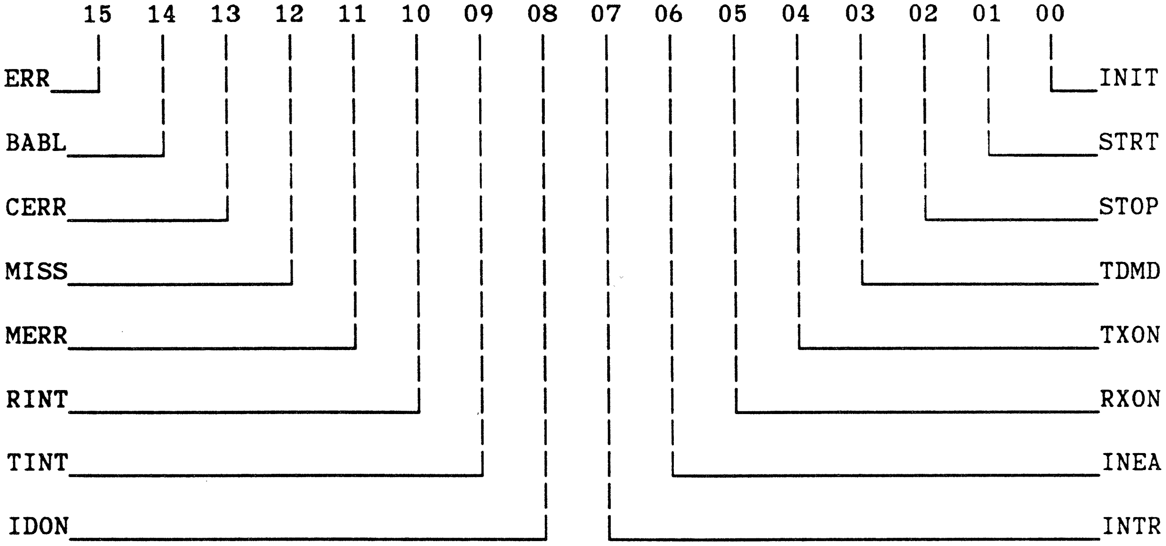


Figure 6. Control Status Register Zero (CSR0)

Bit 15 **ERR:** ERROR summary is set by the "OR" of BABL, CERR, MISS, and MERR. ERR remains set as long as any of the error flags are true. ERR is read only, writing to it has no effect. It is cleared by HRDWRE RESET or VMEbus SYSRESET, by setting the STOP bit, or by clearing the individual error flags.

NOTE

Babble (bit 14), Missed Packet (bit 12), Memory Error (bit 11), Receive Interrupt (bit 10), Transmit Interrupt (bit 9), and Initialization Done (bit 8) all create a VMEbus interrupt if and only if the following conditions are met:

INTR (CSR0 bit 7) is set INEA (CSR0 bit 6) is set	Interrupt Vector Level is greater than zero (i.e., BCR bits 2..0 are not 1,1,1)

Bit 14 **BABL:** BABBLE is a transmitter timeout error. It indicates that the transmitter has been on the channel longer than the time required to send a packet of the maximum length. It will be set after 1519 data bytes have been transmitted. The V/Ethernet 3207 will continue to transmit until the whole packet is transmitted or until there is a failure before the whole packet is transmitted. A VMEbus interrupt is generated as specified in the note on page 30. After the interrupting condition has been cleared, the VMEbus interrupt must be enabled by writing a '0' to INT ENB* (BCR bit 4).

BABL is Read/Clear only and is set by the LANCE and cleared by the host writing a '1' into the bit; writing a '0' has no effect. It may also be cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit.

Bit 13 **CERR:** COLLISION ERROR indicates that the collision input to the LANCE failed to activate within two microseconds after a chip-initiated transmission was completed. The collision after transmission is a transceiver test feature. This function is also known as a "heartbeat" or SQE (Signal Quality Error) test.

CERR is Read/Clear only and is set by the LANCE. It is cleared by the host writing a '1' into the bit; writing a '0' has no effect. It is also cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit. A CERR error will not cause an interrupt to be generated.

Bit 12 **MISS:** MISSED PACKET is set to indicate a loss of data when the receiver loses a packet because it does not own any receive buffers. SILO overflow is not reported since there is no receive ring entry in which to write the status. MISS is invalid in internal loopback mode. A VMEbus interrupt is generated as specified in the note on page 30. After the interrupting condition has been cleared, the VMEbus interrupt must be enabled by writing a '0' to INT ENB* (BCR bit 4).

MISS is a Read/Clear only bit and is set by the LANCE and is cleared by the host writing a '1' to the bit; writing a '0' has no effect. It is also cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit.

Bit 11 **MERR:** MEMORY ERROR is set when the LANCE is the internal bus master and has not received Ready within 25.6 microseconds after asserting the address on the DAL (Data/Address Lines of the LANCE) lines. When a Memory Error is detected, the receiver and transmitter are turned off (CSR0, TXON = 0, RXON = 0). A VMEbus interrupt is generated as specified in the note on page 30. After the interrupting condition has been cleared, the VMEbus interrupt must be enabled by writing a '0' to INT ENB* (BCR bit 4).

During normal operation, the buffers between the LANCE and the 256 Kbyte RAM prevent the LANCE from ever generating a Memory Error. If however, Memory Parity is enabled (hardware jumper JA8), and if there is a parity error on a LANCE memory read, then the hardware will force the LANCE to generate a Memory Error. This in turn can generate an interrupt to the host, et cetera. This is the only condition that should ever generate a Memory Error.

MERR is a Read/Clear only bit and is set by the LANCE and is cleared by the host writing a '1' to the bit; writing a '0' has no effect. It is also cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit.

Bit 10 **RINT:** RECEIVER INTERRUPT is set when the LANCE updates an entry in the Receive Descriptor Ring for the last buffer received or reception is stopped due to a failure. A VMEbus interrupt is generated as specified in the note on page 30. After the interrupting condition has been cleared, the VMEbus interrupt must be enabled by writing a '0' to INT ENB* (BCR bit 4).

RINT is a Read/Clear only bit and is set by the LANCE and is cleared by the host writing a '1' to the bit; writing a '0' has no effect. It is also cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit.

Bit 9 **TINT:** TRANSMITTER INTERRUPT is set when the LANCE updates an entry in the Transmit Descriptor Ring for the last buffer sent or transmission is stopped due to a failure. A VMEbus interrupt is generated as specified in the note on page 30. After the interrupting condition has been cleared, the VMEbus interrupt must be enabled by writing a '0' to INT ENB* (BCR bit 4).

TINT is a Read/Clear only bit and is set by the LANCE and is cleared by the host writing a '1' to the bit; writing a '0' has no effect. It is also cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit.

Bit 8 **IDON:** INITIALIZATION DONE indicates that the LANCE has completed the initialization procedure started by setting the INIT bit. When IDON is set, the LANCE has read the Initialization Block from memory and stored the parameters. A VMEbus interrupt is generated as specified in the note on page 30. After the interrupting condition has been cleared, the VMEbus interrupt must be enabled by writing a '0' to INT ENB* (BCR bit 4).

IDON is a Read/Clear only bit and is set by the LANCE and is cleared by the host writing a '1' to the bit; writing a '0' has no effect. It is also cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit.

Bit 7 **INTR:** INTERRUPT FLAG is set by the "OR" of BABL, MISS, MERR, RINT, TINT, and IDON.

A VMEbus interrupt will be generated if INTR is set, if INEA (CSR0, bit 6) is '1', and if the BCR (bits 2--0) contains an Interrupt Vector Level other than 0 (1,1,1). INTR is Read only; writing to this bit has no effect. INTR is cleared by HRDWRE RESET or VMEbus SYSRESET, by setting the STOP bit, or by clearing the condition that caused the interrupt.

After the interrupting condition (BABL, MISS, MERR, RINT, TINT, or IDON) has been cleared, the VMEbus interrupt must be enabled by writing a '0' to INT ENB* (BCR bit 4).

Bit 6 **INEA:** INTERRUPT ENABLE allows the LANCE to generate a VMEbus interrupt when the interrupt flag is set. INEA is Read/Write and is automatically cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit. If INEA is '1', if the BCR (bits 2--0) contains an Interrupt Vector Level other than level 0 (1,1,1), and if interrupts have been enabled via the INT ENB* (BCR bit 4) when the INTR (CSR0 bit 7) is set, a VMEbus interrupt will be generated.

Bit 5 **RXON:** RECEIVER ON indicates that the receiver is enabled. RXON is set when STRT is set if DRX is '0' in the Mode word of the Initialization block. (The LANCE must be initialized before RXON can be set.) RXON is cleared in two circumstances: when IDON is set via the INIT bit and DRX is '1' in the Mode word of the Initialization block, or when a memory error (MERR) occurs. RXON is Read only; writing this bit has no effect. RXON is also cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit.

Bit 4 **TXON:** TRANSMITTER ON indicates that the transmitter is enabled. TXON is set when STRT is set if DTX is '0' (DTX is in the Mode word of the Initialization block) and the INIT bit has been set. TXON is cleared when IDON is set and DTX is '1' in the Mode word or an error such as MERR, CFLO, or BUFF has occurred during transmission.

TXON is Read only; writing to this bit has no effect. TXON is cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit.

Bit 3 **TDMD:** TRANSMIT DEMAND, when set, causes the LANCE to access the Transmit Descriptor Ring without waiting for the polltime interval to elapse. TDMD does not have to be set to transmit a packet; it merely hastens the V/Ethernet 3207's response to a Transmit Descriptor Ring entry insertion by the host.

TDMD is Write with '1' only and cleared by the microcode after it is used. It may read as a '1' for a short time after it is written because the microcode may have been busy when TDMD was set. It is also cleared by HRDWRE RESET or VMEbus SYSRESET or by setting the STOP bit. Writing a '0' to this bit has no effect.

Bit 2 **STOP:** STOP disables the LANCE from all external activity when set. STOP also clears the LANCE internal logic. Setting STOP is the equivalent of asserting Bus Reset on the LANCE. Bus Reset is a hardware reset pin on the LANCE. STOP is asserted directly by the host by HRDWRE RESET (BCR bit 6) or by the VMEbus SYSRESET signal. The LANCE remains inactive and STOP remains set until the STRT or the INIT bit is set. If STRT, INIT, and STOP are all set simultaneously, STOP will override the other bits and only STOP will be asserted.

STOP is Read/Write with '1' only and is also set by Reset. Writing a '0' to this bit has no effect. STOP is cleared by setting either the INIT or STRT bits. If STOP has been set, CSR1, CSR2, and CSR3 should be reloaded prior to setting STRT.

Bit 1 **STRT:** START enables the LANCE to send and receive packets, perform local memory access and manage the buffers. The STOP bit must be set prior to setting the STRT bit. Setting STRT clears the STOP bit (CSR1, CSR2, and CSR3 should be reloaded prior to setting STRT). If STRT and INIT are set together, the INIT function will be executed first.

STRT is Read/Write with '1' only. Writing a '0' into this bit has no effect. STRT is cleared by HRDWRE RESET or VMEbus SYSRESET, or by setting the STOP bit.

Bit 0 **INIT:** INITIALIZE, when set, causes the LANCE to begin the initialization procedure and access the Initialization Block. The STOP bit must be set prior to setting the INIT bit. Setting INIT clears STOP. If STRT and INIT are set together, the INIT function will be executed first.

INIT is Read/Write with '1' only. Writing a '0' into this bit has no effect. INIT is cleared by HRDWRE RESET or VMEbus SYSRESET, or by setting the STOP bit.

Control and Status Register 1 - CSR1 (Read/Write)

CSR1 is selected when RAP bits one and zero are '0,1' respectively. It is Read/Write, and accessible only when the STOP bit of CSR0 is '1'. The contents of CSR1 are not preserved after a HRDWRE RESET or VMEbus SYSRESET, or when the STOP bit (CSR0 bit 2) is set.

Bits 15 through one of CSR1 indicate the low order 16 bits of the address of the first word (lowest address) of the Initialization Block. Bit zero of CSR1 is reserved and must be '0'.

Control and Status Register 2 - CSR2 (Read/Write)

CSR2 is selected when RAP bits one and zero are '1,0' respectively. It is Read/Write, and accessible only when the STOP bit of CSR0 is '1'. The contents of CSR2 are not preserved after a HRDWRE RESET or VMEbus SYSRESET, or when the STOP bit (CSR0 bit 2) is set.

Bits 15 through eight of CSR2 are reserved. Bits seven through zero indicate the high order eight bits of the address of the first word (lowest address) of the Initialization block.

Control and Status Register 3 - CSR3 (Read/Write)

CSR3 is selected when RAP bits one and zero are '1,1'. It is the mechanism through which the LANCE defines the Bus Master Interface. CSR3 is Read/Write and accessible only when the STOP bit of CSR0 is '1'. The contents of CSR3 are not preserved after a HRDWRE RESET or VMEbus SYSRESET, or when the STOP bit (CSR0 bit 2) is set.

Bits 15--3 **RESERVED** - These bits are reserved and must be set to '0'.

Bit 2 **BSWP** - FOR THE V/Ethernet 3207 TO FUNCTION PROPERLY, THIS BIT MUST BE SET TO '1'.

Bit 1: **ACON** - FOR THE V/Ethernet 3207 TO FUNCTION PROPERLY, THIS BIT MUST BE SET TO '0'.

Bit 0: **BCON** - FOR THE V/Ethernet 3207 TO FUNCTION PROPERLY, THIS BIT MUST BE SET TO '0'.

INITIALIZATION BLOCK

Operating parameters are defined during the initialization procedure which includes reading the Initialization Block in memory. The 24-bit address of the Initialization Block must be written into CSR1 and CSR2 prior to setting INIT. The Initialization Block is read by the LANCE when INIT (CSR0 bit 00) is set. The INIT bit should be set before, or concurrent with, the STRT bit to insure proper initialization, and subsequently, proper operation. After the LANCE has read the Initialization Block, IDON (Initialization Done) is set, and a VMEbus interrupt is generated if INTR is set, if INEA (CSR0 bit 6) is '1', and if the BCR (bits 2--0) contains an Interrupt Vector Level other than 0 (i.e., 1,1,1). After the interrupting condition has been cleared, the VMEbus interrupt must be enabled by writing a '0' to INT ENB* (BCR bit 4).

Table 5 shows the format of the Initialization Block. Each entry is described in more detail following the table.

Table 5 - Initialization Block

Address	(hex) Offset		Contents
IADR	+00	--->	MODE REGISTER
IADR	+02	--->	PADR BITS 00 THROUGH 15
IADR	+04	--->	PADR BITS 16 THROUGH 31
IADR	+06	--->	PADR BITS 32 THROUGH 47
IADR	+08	--->	LADRF BITS 00 THROUGH 15
IADR	+0A	--->	LADRF BITS 16 THROUGH 31
IADR	+0C	--->	LADRF BITS 32 THROUGH 47
IADR	+0E	--->	LADRF BITS 48 THROUGH 63
IADR	+10	--->	RDRA BITS 00 THROUGH 15
IADR	+12	--->	RLEN & RDRA BITS 16 THROUGH 23
IADR	+14	--->	TDRA BITS 00 THROUGH 15
IADR	+16	--->	TLEN & TDRA BITS 16 THROUGH 23

Following the description of the Initialization Block, a sample program listing is provided to further illustrate the structure and use of the Initialization Block.

MODE REGISTER

The Mode Register allows the V/Ethernet 3207's operating parameters to be altered. Normally, the Mode Register is clear (all zeros).

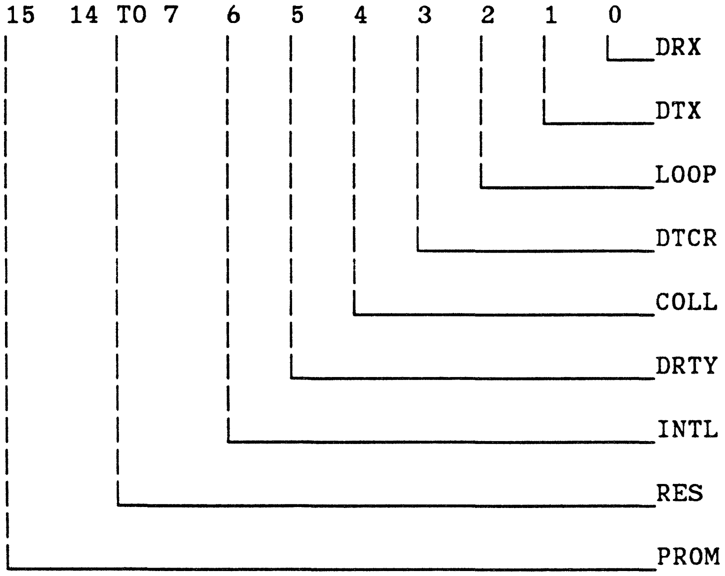


Figure 7. Mode Register Format

Bit 15 **PROM:** PROMISCUOUS mode is selected by setting this bit to '1'. In this mode, all incoming packets are accepted, regardless of the Ethernet node address.

Bits 14--7 **RES:** RESERVED. These bits are reserved.

Bit 6 **INTL:** INTERNAL LOOPBACK is used with the LOOP bit to determine where the loopback is to be done. If this bit is '1' and LOOP is '1', internal loopback is selected; if this bit is '0' and LOOP is '1', external loopback is selected. Internal loopback allows the V/Ethernet 3207 to receive its own transmitted packet. Since this represents full duplex operation, the packet size is limited to between eight and 32 bytes. Internal loopback in the LANCE is operational only when the packets are addressed to the node itself.

The LANCE will not receive any packets externally when it is in internal loopback mode.

External loopback allows the LANCE to transmit a packet through the SIA transceiver cable out to the Ethernet coax. Packet sizes must be between eight and 32 bytes (inclusive) long for valid external loopback operation. External loopback is used to determine the operability of all circuitry and connections between the LANCE and the coaxial cable. Multicast addressing in external loopback is valid only when DTCR is '1' (user must append the four-byte CRC). In external loopback mode, the LANCE also receives packets from other nodes. INTL is only valid if LOOP is '1', otherwise it is ignored.

Bit 5 **DRTY:** DISABLE RETRY, when set to '1', specifies that only one attempt will be made to transmit a packet. If there is a collision on the first transmission attempt, a Retry Error (RTRY) will be reported in Transmit Descriptor 3 (TMD3).

Bit 4 **COLL:** FORCE COLLISION allows the collision logic to be tested. The V/Ethernet 3207 must be in internal loopback mode for COLL to be valid. If COLL is '1', a collision will be forced during the subsequent transmission attempt. This will result in 16 total transmission attempts with a Retry Error (RTRY) reported in TMD3.

Bit 3 **DTCR:** DISABLE TRANSMIT CRC, when '0', will cause the transmitter to generate and append a CRC to the transmitted packet. When DTCR is '1', the CRC logic is allocated to the receiver and no CRC is generated.

During loopback, if DTCR is '0', a CRC is generated on the transmitted packet but it will not be checked by the receiver since the CRC logic is shared and cannot generate and check CRC at the same time. The generated CRC will be written into memory with the data and can be checked by the host software.

If DTCR is '1' during loopback, the host software must append a CRC value to the transmit data. The receiver will check the CRC on the received data and report any errors.

Bit 2 **LOOP:** LOOPBACK allows the V/Ethernet 3207 to operate in full duplex mode for test purposes. The packet size must be between eight and 32 bytes (inclusive). The received packet can be up to 36 bytes (32 bytes plus four bytes of CRC) when DTCR is '0'. During loopback, the runt packet filter is disabled because the maximum packet is forced to be smaller than the minimum size Ethernet packet (64 bytes).

When LOOP is '1', simultaneous transmission and reception is allowed for a message constrained to fit within the silo. The V/Ethernet 3207 waits until the entire message is in the silo before serial transmission begins. The incoming data stream fills the silo from behind as it is being emptied. Moving the received message out of the silo to memory does not begin until reception has ended.

In loopback mode, transmit data chaining is not possible. This means that once the end of packet is sensed, the LANCE will not attempt to use the next despriptor data buffer as it does in normal transmit operations. Receive data chaining is possible if receive buffers are 32 bytes long to allow for lookahead.

Bit 1 **DTX:** DISABLE TRANSMITTER causes the LANCE to not access the Transmitter Descriptor Ring and therefore no transmissions are attempted. When DTX is '1', the TXON bit in CSR0 is cleared once initialization is complete.

Bit 0 **DRX:** DISABLE THE RECEIVER causes the LANCE to reject all incoming packets and not access the Receive Descriptor Ring. When DRX is '1', the RXON bit in CSR0 is cleared once initialization is complete.

PHYSICAL ADDRESS REGISTER (PADR)

The Physical Address (PADR) is the unique 48-bit physical address assigned to the V/Ethernet 3207. Bit zero of PADR is reserved and must be zero.

The three-word (48-bit) Ethernet node address must be written into the PADR section of the Initialization Block before the LANCE is instructed to INIT (CSR0 bit zero).

The nonvolatile RAM connected to the EAR is provided as a convenient place to store the Ethernet node address so that the V/Ethernet 3207 Hawk can "remember" its address, but it is the host that must read the EAR and write the address into the PADR prior to LANCE initialization.

The nonvolatile RAM connected to the EAR is 256 Kbits (sixteen 16-bit words), so it can store more than just the 48-bit Ethernet node address. The standard V/Ethernet 3207 Hawk drivers from Interphase use words zero, one, and two for the Ethernet node address. The remainder is available as a handy 13 words of nonvolatile RAM.

LOGICAL ADDRESS FILTER (LADRF)

The Logical Address Filter (LADRF) is a 64-bit mask used by the V/Ethernet 3207 to implement multicast addressing.

If the destination address field of an incoming address (i.e., the LSB of the received address) is a '1', the address is identified as logical and is passed through the address filter.

This filter is a 64-bit mask composed of four 16-bit registers. The incoming address is sent through the CRC circuit. After all 48 bits of the address have gone though the CRC circuit, the high order six bits of the resultant CRC (32-bit CRC) are strobed into a register. This register is used to select one of the 64-bit positions in the Logical Address Filter. If the selected filter is a '1', the address is accepted and the packet will be put in memory. The Logical Address Filter only assures that there is a possibility that the incoming logical address belongs to the node. To determine if it belongs to the node, the incoming logical address that is stored in main memory is compared by software to the list of logical addresses to be accepted by this node.

The task of mapping a logical address to one of 64 bit positions requires a simple computer program which uses the same CRC algorithm (defined by Ethernet).

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is always enabled. If the Logical Address Filter is loaded with all zeros, all incoming logical addresses except broadcast will be rejected. The multicast addressing in external loopback is operational only when DTCR in the Mode Register is set to '1'.

RECEIVE DESCRIPTOR RING POINTER (RDRP)

The Receive Descriptor Ring Pointer (RDRP) contains the address of the Receive Descriptor Ring and specifies its length.

Bits 31--29 **RLEN:** RECEIVE DESCRIPTOR RING LENGTH bits are used to specify the length of the Receive Descriptor Ring. The Receive Ring Length is the number of entries in the receive ring expressed as a power of two.

RLEN	Number of 4-Word Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Bits 28--24 **RES:** These bits are reserved and must be set to '0'.

Bits 23--00 **RDRA:** RECEIVE DESCRIPTOR RING ADDRESS bits are used to indicate the base address (lowest address) of the Receive Descriptor Ring. Bits 02..00 of this field act as an offset to ensure that the Receive Descriptor Rings are aligned on longword boundaries. Therefore, these three bits must be set to '0'.

TRANSMIT DESCRIPTOR RING POINTER (TDRP)

The Transmit Descriptor Ring Pointer (TDRP) contains the address of the Transmit Descriptor Ring and specifies its length.

Bits 31--29 **TLEN:** TRANSMIT DESCRIPTOR RING LENGTH bits are used to specify the length of the Transmit Descriptor Ring. The Transmit Descriptor Ring Length is the number of entries in the receive ring expressed as a power of two.

TLEN	Number of 4-Word Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Bits 28--24 **RES:** These bits are reserved.

Bits 23--00 **TDRA:** TRANSMIT DESCRIPTOR RING ADDRESS bits are used to indicate the base address (lowest address) of the Transmit Descriptor Ring. Bits 02..00 of this field act as an offset to ensure that the Transmit Descriptor Rings are aligned on longword boundaries. Therefore, these three bits must be set to '0'.

```
struct ether_addr {
   u_char   ether_addr_octet[6];
} hawk_paddr;

/* The address contained in this structure must be longword aligned      */
struct le_drp {                         /* Descriptor Ring Pointer       */
        u_short  drp_laddr;             /* Low 16 bits of ring address   */
        u_short  drp_haddr_len;         /* High 8 bits of ring address - exponent*/
};

struct le_iblock {
        u_short ib_mode;                /* Chip's operating parameters   */
        /*
         * The bytes must be swapped within the word, so that, for example,
         * the address 8:0:20:1:25:5a is written in the order
         *              0 8 1 20 5a 25
         */
        u_char  ib_padr[6];
        u_char  ib_ladrf[8];
        struct  le_drp ib_rdrp;         /* Receive Descriptor Ring Pointer  */
        struct  le_drp ib_tdrp;         /* Transmit Descriptor Ring Pointer */
} hawk_ib;

                              /*********************************************/
   hawk_paddr(0) = 0x10;      /* hawk_paddr(0) corresponds to paddr bits    */
   hawk_paddr(1) = 0x32;      /* 00..07.  hawk_paddr(5) likewise corresponds*/
   hawk_paddr(2) = 0x54;      /* to paddr bits 40..47.  Note the            */
   hawk_paddr(3) = 0x76;      /* requirement that paddr bit 00 MUST be set  */
   hawk_paddr(4) = 0x98;      /* to ZERO, thus hawk_paddr(0) always is an   */
   hawk_paddr(5) = 0xba;      /* even number.                               */
                              /*********************************************/
/*
 * typical setup of LANCE initialization block
 */
   hawk_ib.ib_mode     = MODE_LOOP| MODE_INTL;    /* i.e. internal loopback    */
   hawk_ib.ib_padr[0] = hawk_paddr(1);
   hawk_ib.ib_padr[1] = hawk_paddr(0);
   hawk_ib.ib_padr[2] = hawk_paddr(3);
   hawk_ib.ib_padr[3] = hawk_paddr(2);
   hawk_ib.ib_padr[4] = hawk_paddr(5);
   hawk_ib.ib_padr[5] = hawk_paddr(4);
   hawk_ib.ib_ladrf[0]= 0;
   hawk_ib.ib_ladrf[1]= 0;
   hawk_ib.ib_ladrf[2]= 0;
   hawk_ib.ib_ladrf[3]= 0;
   hawk_ib.ib_ladrf[4]= 0;
   hawk_ib.ib_ladrf[5]= 0;
   hawk_ib.ib_ladrf[6]= 0;
   hawk_ib.ib_ladrf[7]= 0;
   hawk_ib.ib_rdrp.drp_laddr =
        (unsigned short)(&rx_message_descriptor_ring[0]);
    if ( (n_rings = (unsigned char)g2s_exp(NUMBER_OF_RXRINGS)) == 8  ) {
        printf("\nBad number %x of receive rings",NUMBER_OF_RXRINGS);
    exit(NOK);
    }
    hawk_ib.ib_rdrp.drp_haddr_len =
        (unsigned short)((&rx_message_descriptor_ring[0])>>16)+(n_rings<<13));
    hawk_ib.ib_tdrp.drp_laddr =
        (unsigned short)(&tx_message_descriptor_ring[0]);
    if ( (n_rings = (unsigned char)g2s_exp(NUMBER_OF_TXRINGS)) == 8  ) {
        printf("\nBad number %x of transmit rings",NUMBER_OF_TXRINGS);
        exit(NOK);
    }
    hawk_ib.ib_tdrp.drp_haddr_len =
                                                 (unsigned
short)((&tx_message_descriptor_ring[0])>>16)+(n_rings<<13));

/*
 * g2s_exp - Get the power of two of the number.
 *           Returns (0 - 7) for valid number of rings,
 *           Returns (8) for invalid number of rings.
 */
g2s_exp(num)

    u_char num;

{
    switch(num) {
    case 128:
         return(7);
    case  64:
         return(6);
    case  32:
         return(5);
    case  16:
         return(4);
    case   8:
         return(3);
    case   4:
         return(2);
    case   2:
         return(1);
    case   1:
         return(0);
    default :
         return(8);
    }
}
```

DESCRIPTOR RINGS

Each entry in the descriptor ring is four words long (RMD0, RMD1, RMD2, RMD3 or TMD0, TMD1, TMD2, TMD3). The Receive Descriptor Ring and Transmit Descriptor Ring entries have a similar format. The descriptor ring entries contain the address of data buffers, control and status information, and the buffer byte count.

RECEIVE MESSAGE DESCRIPTOR ZERO (RMD0)

RMD0 contains the low order 16 address bits of the data buffer for this descriptor. This word is written by the host and is unchanged by the LANCE.

RECEIVE MESSAGE DESCRIPTOR ONE (RMD1)

Figure 8 below shows the format of RMD1. A detailed explanation of the bits follows the figure.

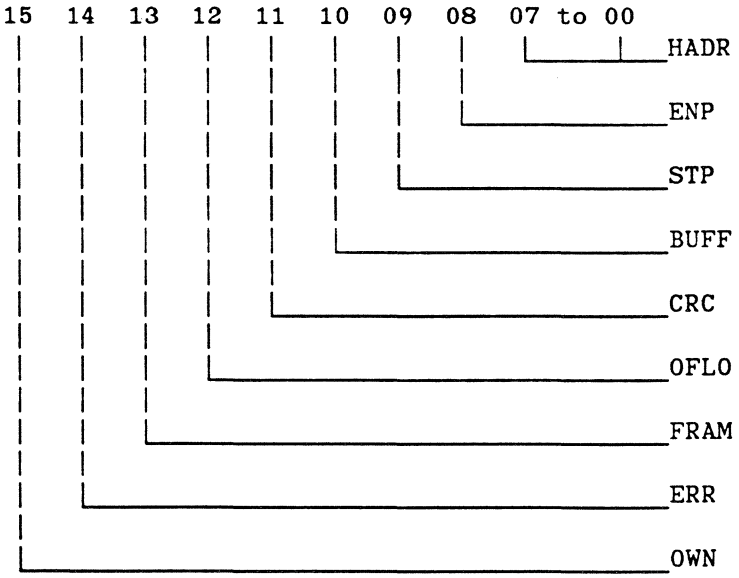


Figure 8. RMD1 Format

Bit 15 **OWN:** The OWN bit indicates that the descriptor entry is owned by the host (OWN is '0') or by the LANCE (OWN is '1'). The LANCE clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the LANCE or host has relinquished ownership of a buffer, it must not change any field in the four words that comprise the descriptor entry.

Bit 14 **ERR:** ERROR summary is the logical "OR" of FRAM, OFLO, CRC, and BUFF (RMD1 bits 10..13).

Bit 13 **FRAM:** FRAMING ERROR indicates that the incoming packet contained a noninteger multiple of eight bits and there was a CRC error. If there was not a CRC error on the incoming packet, the FRAM will not be set even if there is a noninteger multiple of eight bits in the packet. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not set.

Bit 12 **OFLO:** OVERFLOW error indicates that the receiver has lost all or part of the incoming packet due to an inability to store the packet in a memory buffer before the internal silo overflowed. OFLO is valid only when ENP is not set.

Bit 11 **CRC:** CRC indicates that the receiver has detected a CRC error on the incoming packet. CRC is valid only when ENP is set and OFLO is not set.

Bit 10 **BUFF:** BUFFER ERROR is set any time the LANCE does not own the next buffer while data chaining a received packet.

If a Buffer Error occurs, an Overflow Error may also occur internally in the SILO but will not be reported in the descriptor status entry unless both the BUFF and OFLO errors occur at the same time.

Bit 9 **STP:** START OF PACKET indicates that this is the first buffer used by the LANCE for this packet. It is used for data chaining buffers.

Bit 8 **ENP:** END OF PACKET indicates that this is the last buffer used by the LANCE for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fit into one buffer and there was no data chaining.

Bit 7--0 **HADR:** The HIGH ORDER eight address bits of the data buffer are pointed to by this descriptor. This field is written by the host and unchanged by the LANCE.

RECEIVE MESSAGE DESCRIPTOR TWO (RMD2)

Bits 15 through 12 of this word MUST BE ONES. This field is written by the host and unchanged by the LANCE.

The remaining bits, bits 11 through zero, contain the Buffer Byte Count (BCNT). The Buffer Byte Count is the length of the buffer pointed to by this descriptor, expressed as a two's complement number. This field is written by the host and unchanged by the LANCE. The first buffer in the packet has to be at least 64 bytes long.

RECEIVE MESSAGE DESCRIPTOR THREE (RMD3)

Bits 15 through 12 of this word are reserved and MUST BE ZEROS.

The remaining bits, bits 11 through zero, contain the Message Byte Count (MCNT). The Message Byte Count is the length in bytes of the received message. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the LANCE and cleared by the host.

TRANSMIT MESSAGE DESCRIPTOR ZERO (TMD0)

TMD0 contains the low order 16 address bits of the data buffer for this descriptor. This word is written by the host and is unchanged by the LANCE.

TRANSMIT MESSAGE DESCRIPTOR ONE (TMD1)

Figure 9 below shows the format of TMD1. A detailed explanation of the bits follows the figure.

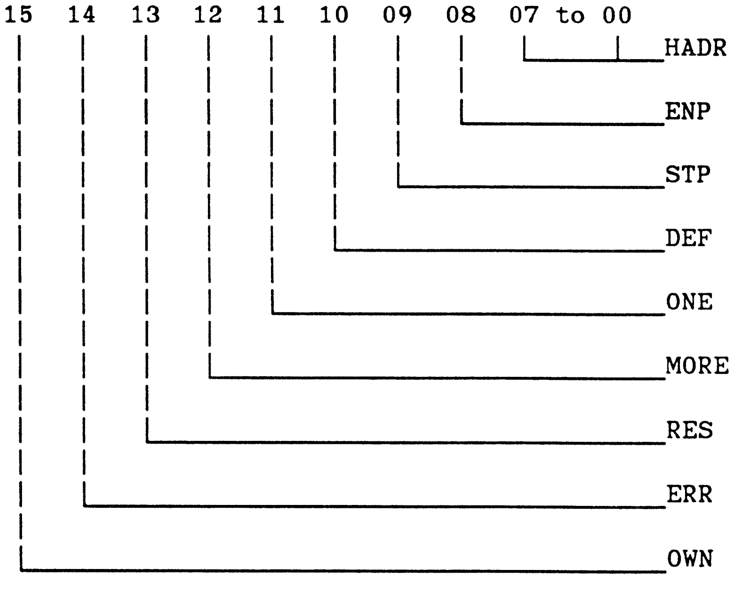


Figure 9. TMD1 Format

Bit 15 **OWN:** The OWN bit indicates that the descriptor entry is owned by the host (OWN is '0') or by the LANCE (OWN is '1'). The host sets the OWN bit after filling the buffer pointed to by this descriptor. The LANCE clears the OWN bit after transmitting the contents of the buffer. Once the LANCE or host has relinquished ownership of a buffer, it must not change any field in the four words that comprise the descriptor entry.

Bit 14 **ERR:** ERROR summary is the logical "OR" of LCOL, LCAR, UFLO, and RTRY. (TMD3 bits 12, 11, 14, and 10 respectively.)

Bit 13 **RES:** RESERVED. This bit is reserved and the LANCE will write a '0' to this bit.

Bit 12 **MORE:** MORE indicates that more than one retry was needed to transmit a packet.

Bit 11 **ONE:** ONE indicates that exactly one retry was needed to transmit a packet. ONE flag is not valid when LCOL is set in TMD3.

Bit 10 **DEF:** DEFERRED indicates that the LANCE had to defer while trying to transmit a packet. This condition occurs if the channel is busy when the LANCE is ready to transmit.

Bit 09 **STP:** START OF PACKET indicates that this is the first buffer to be used by the LANCE for this packet. It is used for data chaining buffers. STP is set by the host and unchanged by the V/Ethernet 3207. The STP bit must be set in the first buffer of the packet. The LANCE will otherwise skip over this descriptor, and poll the next descriptor(s) until both OWN and STP are set.

Bit 08 **ENP:** END OF PACKET indicates that this is the last buffer to be used by the V/Ethernet 3207 for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fit into one buffer and there was no data chaining. ENP is set by the host and unchanged by the LANCE.

Bit 7--0 **HADR:** The HIGH ORDER eight address bits of the data buffer are pointed to by this descriptor. This field is written by the host and unchanged by the LANCE.

TRANSMIT MESSAGE DESCRIPTOR TWO (TMD2)

Bits 15 through 12 of this word MUST BE ONES. This field is written by the host and unchanged by the LANCE.

The remaining bits, bits 11 through zero, contain the Buffer Byte Count (BCNT). The Buffer Byte Count is the usable length of the buffer pointed to by this descriptor, expressed as a two's complement number. This is the number of bytes from this buffer that will be transmitted by the V/Ethernet 3207. This field is written by the host and unchanged by the LANCE. The first buffer in the packet has to be at least 100 bytes long when data chaining, and 64 bytes (DTCR is '1', transmit CRC enabled) or 60 bytes (DTCR is '0', transmit CRC disabled) when not data chaining.

TRANSMIT MESSAGE DESCRIPTOR THREE (TMD3)

The format of Transmit Message Descriptor 3 (TMD3) is depicted in Figure 10 below. A detailed description of the bits follows the figure.

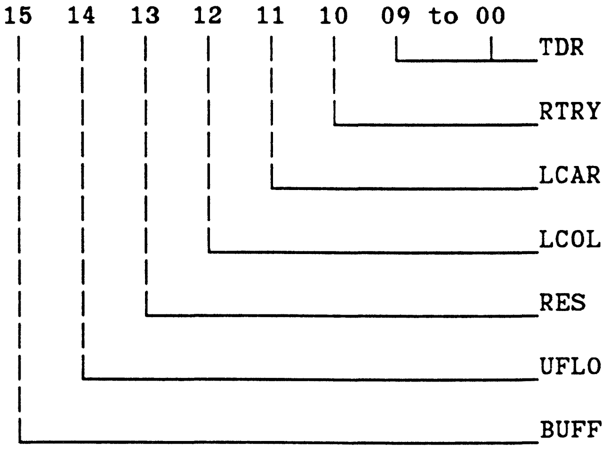


Figure 10. TMD3 Format

Bit 15 **BUFF:** BUFFER ERROR is set by the LANCE during transmission when the LANCE does not find the ENP flag in the current buffer and does not own the next buffer. BUFF is set by the LANCE and cleared by the host. BUFF error will turn off the transmitter, (TXON in CSR0 goes to '0').

If a Buffer Error occurs, an Underflow Error will also occur. BUFF is not valid when LCOL or RTRY is set during TX data chaining.

Bit 14 **UFLO:** UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the SILO has emptied before the end of the packet was reached. On a UFLO error, the transmitter is turned off (TXON in CSR0 goes to '0').

Bit 13 **RES:** RESERVED. The LANCE will write a '0' to this bit.

Bit 12 **LCOL:** LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The LANCE does not retry on late collisions.

Bit 11 **LCAR:** LOSS OF CARRIER is set when the carrier input (RENA) to the LANCE goes false during a chip-initiated transmission. The LANCE does not retry upon loss of carrier. It will continue to transmit the entire packet until it is finished. LCAR is not valid in Internal Loopback Mode.

Bit 10 **RTRY:** RETRY ERROR indicates that the transmitter has failed in 16 attempts to successfully transmit a message due to repeated collisions on the medium. If DRTY is '1', in the Mode Register of the Initialization Block, RTRY will set after one failed transmission attempt.

Bits 9--0 **TDR:** TIME DOMAIN REFLECTOMETRY reflects the state of an internal chip counter that counts from the start of a transmission to the occurrence of a collision. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the LANCE and is valid only if RTRY is set.

SECTION 4

INSTALLATION OF THE V/Ethernet 3207 HAWK

CABLING

There are two versions of the V/Ethernet 3207, Dash 0 and Dash 1, that dictate the cabling options that are available. The Dash 1 board is the standard configuration; it allows Ethernet I/O signals to be directed out of the V/Ethernet 3207 front panel only. The Dash 0 board is the nonstandard configuration; it allows Ethernet I/O signals to be directed out of either the front panel or off of the VMEbus P2 connector. The dual version system was developed to enable the V/Ethernet 3207 to work in the widest range of systems. The specific hardware configuration for each version is detailed below.

DASH 0

- Supports front panel Ethernet I/O
- Supports VMEbus P2 Ethernet I/O
- VMEbus P2 connector utilizes Rows A, B and C
- Transformer in position TX1 for front panel I/O
- Jumper JA1 in position 1--2 for front panel I/O
- Transformer in position TX2 for VMEbus P2 I/O
- Jumper JA1 in position 2--3 for VMEbus P2 I/O

DASH 1

- Supports front panel Ethernet I/O only
- VMEbus P2 connector utilizes Row B only
- Transformer must be seated in TX1 position only
- Jumper JA1 must be in position 1--2 only

Figure 11 on the following page shows the position of the cable connections, the jumpers, and the location of the option switches on the V/Ethernet 3207 printed circuit card. These positions are standard regardless of the board version. Please refer to the diagram for information as indicated in the remainder of this section of the user's guide.

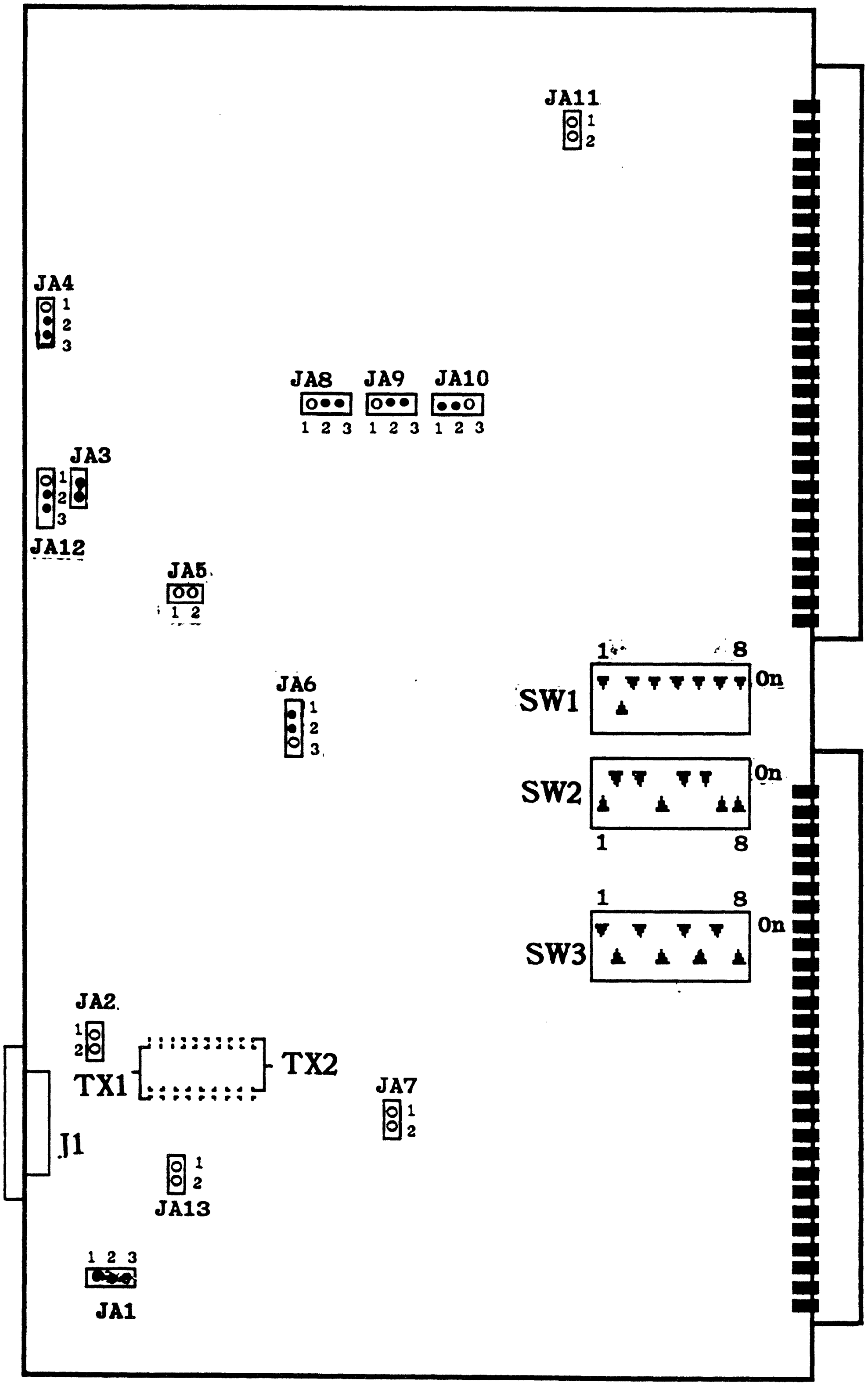


Figure 11. Board Layout

SWITCH SETTINGS

The V/Ethernet 3207 has three switch blocks (SW1, SW2, SW3) that contain eight switches (numbered 1-8) each. (See Figure 11 on the following page for their locations on the board.) These switches are used to select the VMEbus base address of the controller RAM, the base address for Short I/O space, and the allowable address modifiers.

SW1

ADDRESS BIT	A15	A14	A13	A12	A11	A10	A09	SUP
SWITCH #	1	2	3	4	5	6	7	8
POSITION	ON	OFF	ON	ON	ON	ON	ON	ON

SW2

ADDRESS BIT	A23	A22	A21	A20	A19	A18	32-bit	24-bit
SWITCH #	1	2	3	4	5	6	7	8
POSITION	OFF	ON	ON	OFF	ON	ON	OFF	OFF

SW3

ADDRESS BIT	A31	A30	A29	A28	A27	A26	A25	A24
SWITCH #	1	2	3	4	5	6	7	8
POSITION	ON	OFF	ON	OFF	ON	OFF	ON	OFF

Figure 12. Sample Switch Block Setting

Switches SW2 and SW3 set the base VMEbus address of the 256 Kbytes of dual ported RAM on the V/Ethernet 3207. SW2-1 through SW2-6 set address bits A23-A18 respectively. SW3-1 through SW3-8 set address bits A31-A24 respectively. An OFF switch has a value of '1', and an ON switch has a value of '0'. The base address for the Short I/O space of the V/Ethernet 3207 Hawk is set by SW1. SW1-1 through SW1-7 set address bit A15-A09 respectively. An ON switch has a value of '0' and an OFF switch has a value of '1'. SW2-7 selects the address modifiers that are permitted in the extended 32-bit address space. SW2-8 selects the address modifiers that are permitted in the standard 24-bit address space. SW1-8 selects the modifiers that are permitted in the Short I/O 16-bit space.

		SWITCH POSITION	
		ON	OFF
ADDRESS SPACE	SWITCH	ADDRESS MODIFIERS PERMITTED (Sup. Only)	
Extended	SW2-7	0D	0D and 09
Standard	SW2-8	3D	3D and 39
Short I/O	SW1-8	2D	2D and 29

The standard factory settings, shown in Figure 12, show a base VMEbus address of 0x900000 for standard 24-bit addressing, with access allowed by address modifiers 0D and 09 for the extended 32-bit address space and by address modifier 3D and 39 for the standard 24-bit address space. The base address of Short I/O space is set at 4000, and access is restricted to supervisor only (address modifier 2D).

The following figure contains a set of blank switch blocks so that the user can maintain a record of the desired switch settings for his/her application.

SW1

ADDRESS BIT	A15	A14	A13	A12	A11	A10	A09	SUP
SWITCH #	1	2	3	4	5	6	7	8
POSITION	___	___	___	___	___	___	___	___

SW2

ADDRESS BIT	A23	A22	A21	A20	A19	A18	32-bit	24-bit
SWITCH #	1	2	3	4	5	6	7	8
POSITION	___	___	___	___	___	___	___	___

SW3

ADDRESS BIT	A31	A30	A29	A28	A27	A26	A25	A24
SWITCH #	1	2	3	4	5	6	7	8
POSITION	___	___	___	___	___	___	___	___

Figure 13. Switch Block Record

JUMPER SETTINGS

There are 13 hardware jumpers on the V/Ethernet 3207 Hawk. They allow the user to alter many of the operating characteristics of the board, thereby custom configuring the board for use in specific system. Several of the jumpers are set at the factory and should not be altered by the user (these jumpers are noted in the following text). Refer to Figure 11 for the location of each jumper on the board.

JA1 1 - 2 +12V (fused) transceiver power to AUI connection at J1 (front panel) connector--**Factory Default**)

2 - 3 +12V (fused) transceiver power to AUI connection at VMEbus P2 connector

JA2 Reserved for future use. This jumper should not be modified by the user.

JA3 IN Parity check on data bits zero through 31--**Factory Default, Do Not Modify**

OUT Parity check on data bits zero through 15

JA4 1 - 2 DTACK timing for VMEbus initiated transfers is slow. This is for use when parity errors are reported to the VMEbus master as VMEbus error (BERR) (see JA8, JA12).

2 - 3 DTACK timing for VMEbus initiated transfers is fast. This is preferred when parity errors are sensed by polling board level status register (BSLR) (see JA8, JA12)--**Factory Default.**

JA5 Reserved for future use. This jumper should not be modified by the user.

JA6	1 - 2	Source of DRAM is set at localbus address line 17. (This must be used with 32-bit wide DRAM array.)--**Factory Default, User Should Not Modify.**
	2 - 3	Source of DRAM is set at localbus address line 1. (This must be used with 16-bit wide DRAM array.)
JA7	- IN	Factory test point. Normal operation
	- OUT	Factory test point. Interrupt test position--**Factory Default.**
JA8	1 - 2	Enable special parity error reporting. The V/Ethernet 3207 will report VMEbus cycle parity errors by VMEbus error signal (BERR) and LANCE cycle parity errors will cause a memory error (CSR0 bit 11) (See JA4, JA12).
	2 - 3	Disable special parity error reporting (see JA4, JA12.--**Factory Default.**
JA9	1 - 2	Enable VMEbus reporting of parity error using BERR signal.
	2 - 3	Disable VMEbus reporting of parity error using BERR signal. (NOTE: VMEbus BERR is asserted only on parity errors detected during V/Ethernet 3207 bus accesses. LANCE induced parity errors do not generate VMEbus BERRs.)--**Factory Default.**
JA10	1 - 2	SYSCLOCK signal supplied by VMEbus (16 MHz typical).--**Factory Default.**
	2 - 3	SYSCLOCK signal supplied on-board the V/Ethernet 3207. the V/Ethernet 3207 can operate in systems that do not provide SYSCLOCK, but there is a slight performance penalty for Short I/O and IACK cycles.
JA11	IN	SYSFAIL to VMEbus is enabled.
	OUT	SYSFAIL to VMEbus disabled.--**Factory Default.**

JA12 1 - 2 This is set if parity errors are being reported to the LANCE as memory errors (see JA4, JA8).

2 - 3 This setting is preferred if parity errors are sensed by polling board level status register (see JA4, JA8).--**Factory Default.**

JA13 IN Selects full-step signaling AUI interface (for Ethernet version 1.0 applications).

OUT Selects half-step signaling AUI interface (for Ethernet version 2.0 and IEEE 802.3 applications).--**Factory Default.**

INSTALLATION

The V/Ethernet 3207 is designed to ensure easy installation into the VME system. Upon receipt of the board, check to make sure that no damage has occurred during shipping. Usually, a thorough visual inspection is sufficient since each board is thoroughly checked at Interphase just prior to shipment. However, please note that the V/Ethernet 3207 is sensitive to electrostatic discharge (ESD) and the board could be damaged if handled improperly. Interphase ships the board enclosed in a special anti-static bag. Upon receipt of the board, take the proper measures to eliminate board damage due to ESD (i.e., wear a wrist ground strap or other grounding device).

WARNING

Do NOT install or apply power to a damaged board. Failure to observe this warning could result in extensive damage to the board and/or system.

If the board is undamaged and all parts are accounted for, proceed with the installation.

1. The first step is to set all on-board switches so that the V/Ethernet 3207 is properly configured for operation within your system. Those settings are discussed at the beginning of this section, and should be reviewed before continuing.

2. Once the board is configured, ensure that the system power is OFF.

WARNING

System power must be OFF before the V/Ethernet 3207 can be installed. Failure to do so may result in severe damage to the board and/or system.

3. When the power is off, connect the transceiver cable to the transceiver, making sure that the pins are properly oriented. Transceiver cable pin-outs are detailed in Appendix A.

4. Carefully slide the board into the VME card slot. It should slide all the way in without any difficulty.

5. Once the board is properly installed in the slot, tighten the captive mounting screws on each end of the board.

When the board is installed, run a complete test on the system to ensure system integrity.

NOTE

The +12V power that is provided to the transceiver is fused. If the transceiver draws excessive current the fuse will "trip" causing an inability to transmit or receive. If this occurs, disconnect the transceiver cable (from either P2 or J1) and let the heat sensitive fuse cool for approximately one minute. When cool, the fuse will automatically reset and the cable may be re-attached.

SECTION 5

SPECIFICATIONS OF THE V/Ethernet 3207 HAWK

VMEbus SPECIFICATIONS

DTB Slave	A24, A32, D8, D16, or D32 for Data Transfers A16, D16 for Command/Status
Interrupter	Any of I(1-7) Dynamic

ENVIRONMENTAL SPECIFICATIONS

Operating Temperature	32-131 degrees F. (0-55 degrees C.)
Maximum Humidity	10-90% noncondensing

ELECTRICAL SPECIFICATIONS

Power	5.7 Amps max. and 4.2 Amps typical @ +5V DC ±5% 500 mA max @ +12V DC ±5% (transceiver power)

MECHANICAL SPECIFICATIONS

Configuration	EXP, Double Height VMEbus card 6.3" x 9.2" (160mm x 233mm)
Cables	DB15 female connector w/slide latch as specified in IEEE 802.3 specification

APPENDIX A

APPENDIX A - CONNECTOR PIN-OUTS

Table A-1: VMEbus Connector P1 Signal Descriptions

PIN NUMBER	ROW A	ROW B	ROW C
1	Data 00		Data 08
2	Data 01		Data 09
3	Data 02		Data 10
4	Data 03	Bg 0 in-	Data 11
5	Data 04	Bg 0 out-	Data 12
6	Data 05	Bg 1 in-	Data 13
7	Data 06	Bg 1 out-	Data 14
8	Data 07	Bg 2 in-	Data 15
9	Ground	Bg 2 out-	Ground
10	SYSCLOCK	Bg 3 in-	SYSFAIL-
11	Ground	Bg 3 out-	BERR-
12	DS 1-		SYSRESET-
13	DS 0-		LWORD-
14	WRITE-		AM 05
15	Ground		Address 23
16	DTACK-	AM 00	Address 22
17	Ground	AM 01	Address 21
18	AS-	AM 02	Address 20
19	Ground	AM 03	Address 19
20	IACK	Ground	Address 18
21	IACKIN-		Address 17
22	IACKOUT-		Address 16
23	AM 04	Ground	Address 15
24	Address 07	IRQ 7-	Address 14
25	Address 06	IRQ 6-	Address 13
26	Address 05	IRQ 5-	Address 12
27	Address 04	IRQ 4-	Address 11
28	Address 03	IRQ 3-	Address 10
29	Address 02	IRQ 2-	Address 09
30	Address 01	IRQ 1-	Address 08
31			+12V DC
32	+5V DC	+5V DC	+5V DC

If no signal is referenced, then the V/Ethernet 3207 does not use that pin. (A "-" symbol denotes an active low signal.)

Table A-2: VMEbus Connector P2 Signal Descriptions (DASH 1 Only)*

PIN NUMBER	ROW A	ROW B	ROW C
1		+5V DC	
2		Ground	
3			
4		Address 24	
5		Address 25	
6		Address 26	
7		Address 27	
8		Address 28	
9		Address 29	
10		Address 30	
11		Address 31	
12		Ground	
13		+5V DC	
14	Ground	Data 16	Ground
15	Reserved	Data 17	Reserved
16	Ground	Data 18	Ground
17	Reserved	Data 19	Reserved
18	Ground	Data 20	Ground
19	Reserved	Data 21	Reserved
20	Ground	Data 22	Ground
21	Reserved	Data 23	Reserved
22	Ground	Ground	Ground
23	Collision+	Data 24	Collision-
24	Ground	Data 25	Ground
25	Transmit+	Data 26	Transmit-
26	Ground	Data 27	Ground
27	Receive+	Data 28	Receive-
28	Ground	Data 29	Ground
29	Fused +12V DC	Data 30	Fused +12V
30	Ground	Data 31	Ground
31	Reserved	Ground	Reserved
32	+5V DC	+5V DC	+5V DC

If no signal is referenced, then the V/Ethernet 3207 does not use that pin. (A "-" symbol denotes an active low signal.)

* There are two versions of the V/Ethernet 3207, DASH 0 and DASH 1, that require separate P2 connector configurations.

Table A-3: VMEbus Connector P2 Signal Descriptions (DASH 0 Only)*

PIN NUMBER	ROW A	ROW B	ROW C
1		+5V DC	
2		Ground	
3			
4		Address 24	
5		Address 25	
6		Address 26	
7		Address 27	
8		Address 28	
9		Address 29	
10		Address 30	
11		Address 31	
12		Ground	
13		+5V DC	
14		Data 16	
15		Data 17	
16		Data 18	
17		Data 19	
18		Data 20	
19		Data 21	
20		Data 22	
21		Data 23	
22		Ground	
23		Data 24	
24		Data 25	
25		Data 26	
26		Data 27	
27		Data 28	
28		Data 29	
29		Data 30	
30		Data 31	
31		Ground	
32		+5V DC	

If no signal is referenced, then the V/Ethernet 3207 does not use that pin. (A "-" symbol denotes an active low signal.)

* There are two versions of the V/Ethernet 3207, DASH 0 and DASH 1, that require separate P2 connector configurations.

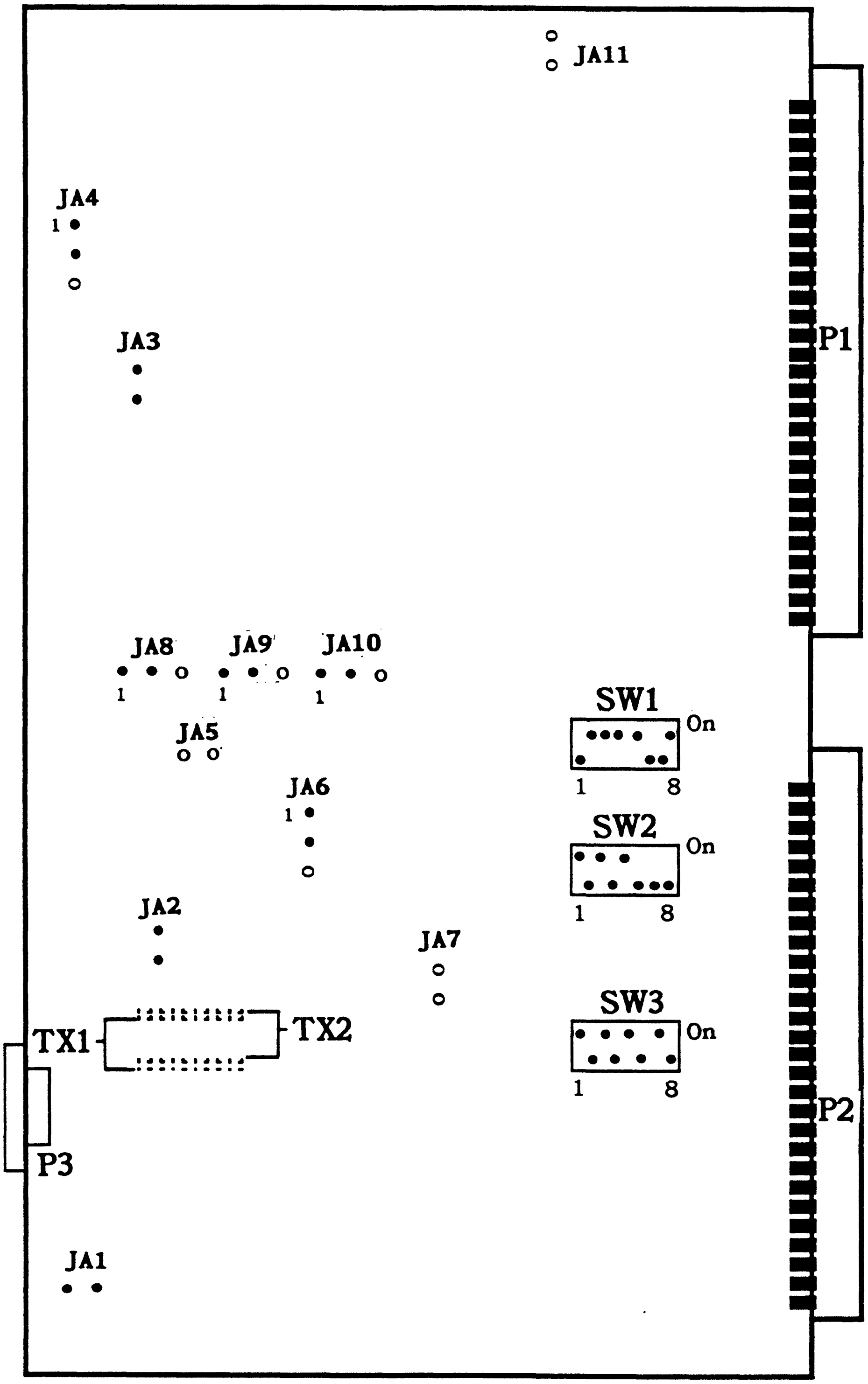
Table A-4: Transceiver Connector (J1) Pin Assignments

PIN NUMBER	SIGNAL
1	Ground (Control IN Shield)
2	Control IN Circuit A (Collision+)
3	Data OUT Circuit A (Transmit+)
4	Ground (Data IN Shield)
5	Data IN Circuit A (Receive+)
6	Ground (Power-)
7	
8	Ground (Control OUT Shield)
9	Control IN Circuit B (Collision-)
10	Data OUT Circuit B (Transmit-)
11	Ground (Data OUT Shield)
12	Data IN Circuit B (Receive-)
13	Fused +12V DC
14	Ground (Power Shield)
15	

If no signal is referenced, then the V/Ethernet 3207 does not use that pin. (A "-" symbol denotes an active low signal.)

APPENDIX B

APPENDIX B - HARDWARE JUMPERS FOR V/Ethernet 3207 BOARDS WITH ARTWORK REVISION XOB



JUMPER SETTINGS

The jumper settings detailed in this appendix refer to V/Ethernet 3207 boards with artwork Revision of X0B. There are eleven jumpers on-board the V/Ethernet 3207; however many of the jumpers should not be modified by the user. The user selectable jumpers are as follows: JA1, JA4, JA9, JA10 and JA11; all others are set at the factory and should not be altered. A complete description of each of the jumpers is provided below.

JA1 IN Provides fused +12V to Ethernet transceiver (**factory default**).
OUT No power is provided to the transceiver.

JA2 IN Factor default.

JA3 IN Factory default.

JA4 1--2 Parity enabled (slow cycle) (**factory default**)
2--3 Parity disabled (fast cycle).

JA5 OUT Factory default.

JA6 1--2 Factory default.

JA7 OUT Factory default.

JA8 1--2 Factory default.

JA9 1--2 When parity is enabled (JA4 is 1--2), PERR will result in a VMEbus error instead of a DTACK (**factory default**).

2--3 When parity is enabled (JA4 is 1--2), parity will still be checked on all reads of the dual port memory. If the parity error was on the VMEbus read, the parity error will be flagged in the Board Status Register. If the parity error occurred on a LANCE read, the interrupt will still be flagged in the BSR. In addition, a memory error (MERR) in the LANCE CSR0 will be set and the host wil be interrupted.

The internal memory appears to run approximately 30 nanoseconds slower when this jumper is set at 1--2 because the memory parity must be checked before DTACK can be asserted. Therefore, it is recommended that the V/Ethernet 3207 be run with this jumper in the 2--3 position. The host will be interrupted on any parity error from the LANCE read, and parity errors on host reads can be detected by checking the BSR after a DMA transfer.

JA10 1--2 VMEbus signal SYSCLCK is available (**factory default**).
2--3 VMEbus signal SYSCLK is not available.

JA11 IN SYSFAIL to VMEbus is enabled.
OUT SYSFAIL to VMEbus is disabled (**factory default**).

GLOSSARY OF TERMS

Address

A number which identifies a specific location. It can refer to a node in the network, a packet of data, or a location in computer memory.

AUI

AUI (Attachment Unit Interface) is the transceiver interface, as specified in the IEEE 802.3 specification.

Address Modifier

A VMEbus-specific 6-bit code used to modify the way a VMEbus address is interpreted. The V/Ethernet 3207 uses address modifiers 29 and 2D to specify Short I/O access.

Cheapernet

An IEEE 802.3 standard network characterized by the type of cable used to connect the nodes. The cable is generally less expensive than the cable used in a standard Ethernet network, thus the name.

CSMA/CD

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) is a method whereby a number of nodes can gain access to a network. Each node waits until no other signal is present on the line before beginning transmission. If two nodes try to access the line simultaneously, a collision is detected. The nodes will quit transmitting and then, after a preset delay, try to re-access the network. Since the delay period is different for each node, another collision is unlikely.

Cyclical Redundancy Check (CRC)

An error detection technique where error information is appended to transmitted data and then checked by the receiving node to insure that no errors occurred in transmission.

Data Chaining

When a packet is transmitted or received, the data is automatically "chained" together as a function of the descriptor rings. That is, if the current buffer fills/empties before the end of packet is sensed, the LANCE will attempt to use the next descriptor/data buffer.

Data Link Layer

The data link layer is the second level of the OSI model of protocol software for data communications. It is responsible for organizing messages and coordinating their flow.

Data Strobe

A signal from a bus master to a bus slave that indicates that a data transfer is occurring.

Data Transfer Acknowledge (DTACK)

An affirmative acknowledgment sent across the VMEbus to a master to indicate that either the data has been received or that the slave has presented data.

DMA (Direct Memory Access)

Direct memory access is an activity that transfers data directly from a peripheral into system memory without requiring the system CPU.

DTB (Data Transfer Bus)

Data transfer bus as used in the VME standard.

Ethernet

Ethernet is a CSMA/CD-based system that uses coaxial cable to connect workstations in the Local Area Network.

EXP (Expanded Double Bus)

Expanded double bus VME board as used in the VME standard.

Firmware

A computer program written onto a storage device such that it cannot be accidentally erased, (i.e., it is stored in Read Only Memory [ROM]).

Header

A header is placed at the beginning of each data packet being transmitted over the network. It contains information concerning destination address, source address, message numbering and other descriptive information.

Host

Refers to the primary or controlling processor in a system.

Interrupt

The interrupt capability of the VMEbus provides a means by which devices can interrupt normal bus activity. These interrupts are prioritized (priority interrupts) into seven levels. Priority interrupts utilize signals IACK, IACKIN, and IACKOUT to acknowledge that the interrupt has been generated.

Layer

A layer refers to several networking functions that are grouped together in a hierarchical manner as outlined in the OSI model of data communications.

Logical Addressing

Most operating systems store data in logical addresses, and in order to access or move the data, the host CPU must translate the logical addresses into physical addresses.

Loopback

Loopback is a diagnostics test that routes a transmitted signal through the network and returns it to the sending device. In this way, it can be used to pinpoint the location of specific problems in the signal's path.

Memory Type

The memory type specifies either 8-, 16- or 32-bit data transfers, or 16-, 24- or 32-bit addressing.

Multitasking

Refers to the ability of a device to perform two or more tasks concurrently.

Multicast Addressing

Refers to a type of Ethernet/Cheapernet addressing whereby a packet is transmitted to every device/node that fits a specified parameter (e.g., all file servers).

Node

A node is any point on a network where a networking communications device resides. It is typically a workstation.

OSI Model

OSI (Open System Interconnect) model is a standard networking framework developed by the International Standards Organization. The framework consists of a seven layer hierarchical model that groups networking functions into logical sections. The seven layers are (from lowest to highest): Physical, Data Link, Network, Transport, Session, Presentation, and Application.

Packet

A packet is the standard format in which data is transmitted over the network. It consists of a header, an address, control information, data elements, error information, and start and stop bits.

Packet Buffer

As the name would imply, a packet buffer is memory space allocated for storing a packet that is ready to be transmitted or has just been received.

Physical Layer

The physical layer is the lowest level of the OSI networking model.

Short I/O

Short I/O is a VMEbus-specified block of memory used to facilitate processing of certain information. On the V/Ethernet 3207, Short I/O is a 512-byte block of memory set aside for handling command parameters, control information, and statuses that occur when the host CPU is acting as the bus master. It is called Short I/O space because the upper 16 VMEbus address lines are ignored, and only the lower 16 are used for transactions that take place in Short I/O.

SILO

A type of First-in-First-out (FIFO) register.

Transceiver

An active device that connects the V/Ethernet 3207 to either the Ethernet cable or Cheapernet cable.

VMEbus

VMEbus is an industry standard high-performance 32-bit bus designed with an open bus architecture.

INDEX

INDEX

B

Babble, 31
Base Address
    Setting of, 55
Block Diagram, 17
Board Control Register, 20
Board Status Register, 22
Buffer Byte Count
    Receive, 48
    Transmit, 51
Buffer Error
    Receive, 47
    Transmit, 52

C

Cabling Options, 53
Collision Error, 31
CRC
    Receive, 47
CSR 0, 30
    BABL, 31
    CERR, 31
    ERR, 30
    IDON, 33
    INEA, 33
    INIT, 35
    INTR, 33
    MERR, 31
    MISS, 31
    RINT, 32
    RXON, 33
    STOP, 34
    STRT, 35
    TDMD, 34
    TINT, 32
    TXON, 34
CSR 1, 35
CSR 2, 36
CSR 3, 36

D

DASH 0, 53, 67
DASH 1, 53, 66
Deferred
    Transmit, 50
Descriptor Rings, 46
Disable Receiver, 40
Disable Retry, 39
Disable Transmit CRC, 39
Disable Transmitter, 40

E

End of Packet
    Receive, 48
    Transmit, 50
Error, 30
    Receive, 47
    Transmit, 50
Ethernet Address Register, 24
External Loopback, 39

F

Framing Error
    Receive, 47

H

Hardware Reset
    HRDWRE RESET, 21

I

Initialization, 37
Initialization Done, 33
Initialize, 35
Installation,
    Board, 59
Internal Loopback, 38
Interrupt Clear
    INT CLR, 21
Interrupt Enable, 33
Interrupt Flag, 33
Interrupt Vector
    Settiing of,, 21
Interrupt Vector Register, 23

IPL, 21

J

Jumper Settings, 57

L

Late Collision, 52
Logical Address Filter, 41
Loopback, 40
Loss of Carrier, 52

M

Memory Error, 31
Message Byte Count
    Receive, 49
Missed Packet, 31
Mode Register, 38
    DRTY, 39
    DRX, 40
    DTCR, 39
    DTX, 40
    INTL, 38
    Loop, 40
    PROM, 38
More
    Transmit, 50

N

Nonvolatile RAM Clear
    NOVRAM CE, 21
NOVRAM, 24
    Instructions, 25

O

Option Switches
    Setting of, 53
Overflow
    Receive, 47
Own Bit
    Receive, 47
    Transmit, 49

P

P1 Connector, 65
P2 Connector, 66, 67
Parity Error, 22
Parity Error Clear
    PERR CLR, 21
Physical Address, 40
Promiscuous Mode, 38

R

Receive Descriptor Ring
    Pointer, 42
Receive Message Descriptor, 46
Receiver Interrupt, 32
Receiver On, 33
Register Address Port, 29
Register Data Port, 29
Retry
    Disabling, 39
Retry Error
    Transmit, 52

S

Start, 35
Start of Packet
    Receive, 48
    Transmit, 50
Stop, 34
System Fail
    SYSFAIL, 20

T

Time Domain Reflectometry, 52
Transceiver
    Connector, 68
Transmit Demand, 34
Transmit Descriptor Ring
    Pointer, 43
Transmit Message Descriptors, 49
Transmitter Interrupt, 32
Transmitter On, 34

U
Underflow Error
    Transmit, 52

INTERPHASE
CORPORATION

OPEN SYSTEMS CONTROLLERS™
Disk • Tape • Networking

13800 Senlac • Dallas, Texas 75234 • (214) 919-9000 • FAX: (214) 919-9200 • NASDAQ-NMS:INPH
Interphase International
Astral House, Granville Way • Bicester, Oxon OX6 OJT • (01144) 869 321222 • FAX (01144) 869-247720