NI1010A UNIBUS ETHERNET COMMUNICATIONS CONTROLLER USER'S MANUAL

UM-NI1010A (950-0002-AA) (Rev. AA)

MICOM-Interlan, Incorporated
155 Swanson Road
Boxborough, MA 01719

REV.	REVISION HISTORY	DATE
00	Original.	11/82

MICOM-INTERLAN
DOCUMENTATION CHANGES

Please note the following changes to this document:

Name

The company's name should be changed to MICOM-Interlan throughout the document.

Address

The company's address should be changed throughout the document to the following:

MICOM-Interlan, Inc.
155 Swanson Rd.,
Boxborough, MA 01719

Telephone (617) 263-9929 Telex 95-1909

Service and Support

Any information on service and support in this document should be changed to the following:

MICOM-Interlan provides a variety of support services for this product, including:

- Technical Assistance Centers, for technical support via the telephone
- A Repair Center, providing warranty or out-of-warranty repairs

On-site support for certain products and systems is also available.

For information on service and support for this product, please contact MICOM Customer Service at one of the following locations.

Eastern Area

MICOM Technical Assistance Center
Boxborough, Massachusetts

1-800-LAN-TALK
1-800-TEL-LLAN (in Massachusetts)

Western Area

MICOM Technical Assistance Center
Simi Valley, California

1-800-833-DATA
1-805-583-8600 (in California)

Warranty

Any information on warranty or return materials authorization in this document may no longer be in effect. For the latest information, contact your local MICOM-Interlan representative.

PREFACE

This manual is intended for the engineer, programmer, or technician who will install, program, and maintain the NI1010A UNIBUS Ethernet Communications Controller.

The manual is divided into the following chapters:

- CHAPTER 1, INTRODUCTION
  Provides a general description of the NI1010A's features, specifications, and related accessories.

- CHAPTER 2, INSTALLATION
  Provides information on how to unpack, configure, install, cable-up, and verify operation of the NI1010A.

- CHAPTER 3, PROGRAMMING INFORMATION
  Describes the NI1010A's programming interface and command operations.

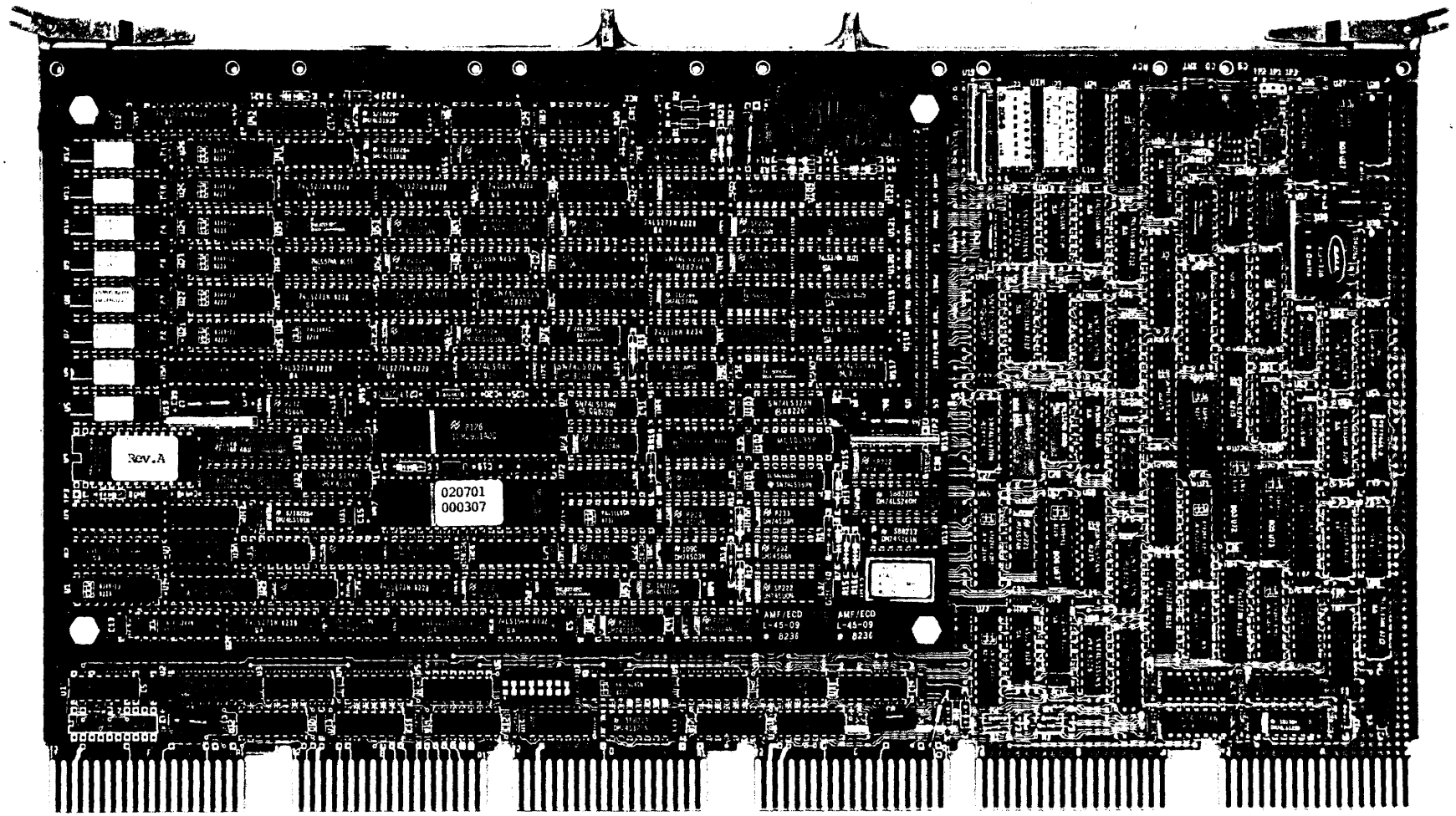
- CHAPTER 4, FUNCTIONAL DESCRIPTION
  Provides a description of the NI1010A's architecture, and functional operation during transmission, reception, and execution of the onboard diagnostics.

- CHAPTER 5, MAINTENANCE
  Supplies information on how to exercise and troubleshoot the NI1010A with the standalone PDP-11 diagnostic.

- APPENDIX A, ETHERNET NETWORK PLANNING, INSTALLATION, AND TEST GUIDELINES
  Provides some helpful hints on how to plan, install, and commission an Ethernet coaxial cable installation.

Supporting documentation on the Ethernet is available in:
THE ETHERNET, A LOCAL AREA NETWORK DATA LINK AND PHYSICAL LAYER SPECIFICATION V1.0; September 30, 1980, Xerox/Intel/Digital.

TABLE OF CONTENTS


Rev.A
020701
000307

CHAPTER ONE

INTRODUCTION

This chapter provides a general description of the NI1010A's features, summarizes the NI1010A product specifications, and describes hardware and software accessory products available for the NI1010A.

1.0 GENERAL DESCRIPTION

The NI1010A UNIBUS Ethernet Communications Controller is a single hex-height board that contains all the data communications controller logic required for interfacing Digital Equipment Corporation's family of VAX-11 and UNIBUS PDP-11 minicomputers to an Ethernet local area network. The controller board complies in full with the Xerox/Intel/Digital Ethernet V1.0 Specification by performing the specified CSMA/CD (Carrier Sense, Multiple Access with Collision Detect) data link and physical channel functions.

As shown in Figure 1-1, the NI1010A, when attached to a transceiver unit, provides the host UNIBUS system a complete connection onto the Ethernet baseband coaxial cable local area network, permitting 10 Mbit per second transmissions over distances up to 2500 meters.

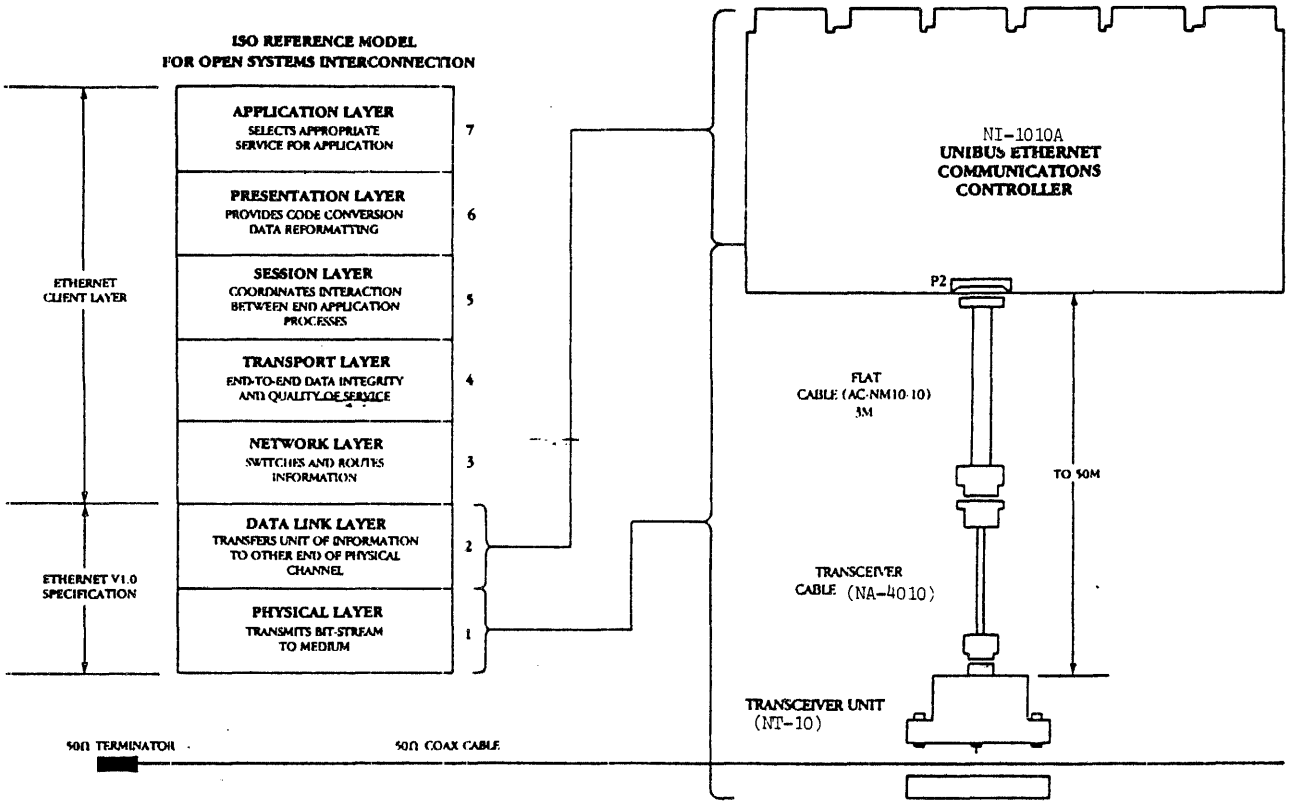


Figure 1-1. Ethernet Architecture and NI1010A Implementation

1.1 FEATURES

Implements Ethernet V1.0 Specification:

The NI1010A fully complies with Xerox/Intel/Digital Ethernet V1.0 Specification.

Implements Ethernet Data Link Layer Functions:

The NI1010A formats frames and performs the CSMA/CD transmit link management functions required to successfully deliver frames onto the network. When not transmitting a frame, the NI1010A continuously listens to the network for frame traffic intended for it. Only frames with a matching address are accepted by the controller for subsequent transfer to the host UNIBUS system. The controller performs Physical, Multicast-Group (up to 63), and Broadcast address recognition. CRC generation and checking is also performed.

Implements Ethernet Physical Channel Functions:

The NI1010A transmits and receives 10 Mbit per second bit-streams with electrical and timing specifications compatible for direct connection to an Ethernet transceiver unit. The controller performs the required frame synchronization functions, and Manchester encoding/decoding of the bit-streams.

Supports High Station Performance:

The NI1010A has been designed to offer high network performance while minimizing the service load placed upon the host UNIBUS system. Serving to buffer the host from the unpredictable interarrival times characteristic of network traffic, the board has a receive FIFO (first-in, first-out) memory which can store up to 13.5 KBytes of received frames. For transmit buffering, the NI1010A has a 1.5K Byte FIFO from which all frame retransmissions are made. All data transfers between the NI1010A and host UNIBUS memory are performed by the NI1010A's DMA controller. The DMA controller may be preloaded by the host with up to 16 receive buffer descriptors.

Extensive Diagnostic Features:

The NI1010A controller offers comprehensive network and board-level diagnostic capabilities. LED indicators mounted on the edge of the board provide a visual indication of whether or not the host is communicating onto the network. For a comprehensive station diagnosis, the NI1010A may be operated in three different types of data loopback. On power-up, or by host command, the controller performs a confidence test on itself. A LED indicator shows the pass/fail operational state of the board.

Collects Network Statistics:

The NI1010A tallies statistical values on various network traffic and error conditions observed.

One Hex-Height UNIBUS Board:

The NI1010A fits into one UNIBUS SPC slot. The controller is mechanically, electrically, and architecturally compatible with Digital Equipment Corporation's UNIBUS specifications.

1.2 SPECIFICATIONS

1.2.1 NI1010A Network Specifications

- 10 million bit per second data transmission rate
- Coaxial cable segments up to 500 meters (1640 feet) in length
- Up to 100 transceiver connections per coaxial cable segment
- Up to 2 repeaters in path between any two stations
- Up to 1500 meters (4920 feet) of coaxial cable between any two stations
- Up to 50 meters (165 feet) of transceiver cable between an NI1010A controller and its transceiver
- Up to 2500 meters (1.55 miles) maximum station separation
- Point-to-point links up to 1000 meters (3280 feet) in length
- Up to 1024 stations per network

1.2.2 NI1010A UNIBUS Specifications

Power Requirements:	+5Vdc +- 5% @ 7.5 A typical, 8.2 A maximum. +15Vdc +- 5% @ 0.5 A maximum (for transceiver only).
Register Addresses:	The NI1010A has three I/O page registers; BA+0, Command and Status Register (CSR), BA+2, Buffer Address Register (BAR), BA+4, Byte Count Register (BCR). The base address (BA) is switch selectable from 760000 to 777760 in increments of 20 (octal).
Interrupt Vectors:	The NI1010A has two interrupt vectors; VA+0, Receive DMA Done (RCV DONE), VA+2, Command Done (CMD DONE). The vector base address (VA) is switch selectable from 000 to 770 in increments of 10 (octal).
Interrupt Priority Levels:	Both NI1010A interrupts reside at the same priority level, BR5. The interrupt priority level may be altered to BR4.
UNIBUS Data Transfers:	All UNIBUS data transfers are via non-processor request (NPR) direct memory

access (DMA). Transfers are 4 word burst.

UNIBUS Loading: The NI1010A represents 1 AC and 1 DC load to the UNIBUS.

1.2.3 Transceiver Interface

All NI1010A transceiver interface (P2) signals are Ethernet V1.0 Specification compatible.

Mating Connectors: 16-pin Berg 65846-01, or 3M 3452, or equivalent.

1.2.4 Environmental Specifications

Operating Temperature: 0 to 50 degrees Celsius (32 to 122 degrees Fahrenheit).

Relative Humidity: to 90%, non-condensing.

1.3 NI1010A PRODUCTS AND ACCESSORIES

The following NI1010A products are available from Interlan:

MODEL NUMBER	DESCRIPTION
BD-NI1010A	NI1010A UNIBUS Ethernet Communications Controller board
DS-NI1010-RX01	NI1010A Standalone PDP-11 Diagnostic on RX01 floppy disk
DS-NI1010-RX02	NI1010A Standalone PDP-11 Diagnostic on RX02 floppy disk
UM-NI1010A	NI1010A User Manual
AC-NM10-10	Flat Cable with Connectors; 10 feet long (3 meters)

The following operating system software support is available for the NI1010A (consult the factory for latest details):

MODEL NUMBER	DESCRIPTION
NS2010	RSX-11M/S Device Driver for the NI1010A controller board
NS2020	RT-11(SJ,FB,XM) Device Handler for NI1010A controller board
NS2030	VMS Device Driver and VAX-11 Diagnostic Program for NI1010A controller board.

In addition, Interlan supplies the following network accessories:

MODEL NUMBER	DESCRIPTION
UN-NA1010	Ethernet Transceiver Unit
AC-NA1010-xxx	Ethernet Transceiver Cable with connectors; available in lengths of 10, 50, and 150 feet
NA1020-xxx	Ethernet 50 ohm Coaxial Cable; available in lengths of 77, 230, and 385 feet.

CHAPTER TWO

INSTALLATION

The NI1010A is a hex-height board that has been designed to be mechanically, electrically, and architecturally compatible with Digital Equipment Corporation's standards for UNIBUS compatibilty. The board may be installed in any system which uses the UNIBUS architecture, including all of DEC's UNIBUS-based PDP-11 and VAX-11 minicomputers.

Chapter 2 provides the necessary information on how to configure the NI1010A, install it into a UNIBUS system, connect it to an Ethernet, and verify its operation.

2.1 UNPACKING AND INSPECTION

To protect against damage during shipment each NI1010A is packed in a special carton. The carton may be be opened at either end to remove the contents. The NI1010A board, and other enclosed material, should be carefully inspected for any visible signs of damage that could have resulted during shipment. If damage appears to exist, the carrier responsible for the shipment, and Interlan Customer Service, should be immediately notified.

It is suggested that all salvageable shipping cartons and packing material be saved for future use in the event the product must be reshipped.

2.2 NI1010A ETHERNET CONSIDERATIONS

The NI1010A is completely compatible with the DEC/Intel/Xerox Version 1.0 Ethernet Specification and may be connected to any 10Mbps Ethernet installation. Appendix A at the back of this manual provides some helpful information about installing and testing an Ethernet coaxial cable transmission system.

2.2.1 Transceiver Placement

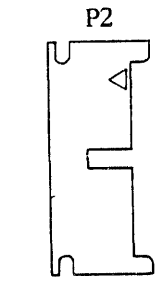
All transceiver connections to the Ethernet transmission cable introduce a finite bridging impedance that causes some, albeit small, amount of signal reflection to occur on the cable. To insure that reflections from transceivers do not add to cause transmission errors, the placement of tansceivers along the cable must be controlled.

Approved Ethernet coaxial cable is marked with annular rings at 2.5 meter intervals. Wherever possible, transceivers should only be placed at one of these rings, minimizing the likelihood of having transceiver reflections with phase angles that add.

The total number of transceivers on a cable segment must not exceed 100.

2.2.2 NI1010A Transceiver Interface and Cable Requirements

The NI1010A connects to the Ethernet via the board's P2 flat cable connector. The connector's pinout is shown below:



BERG 65496-043
OR EQUIV.

Pin		Pin	
1	N/C	2	Collision Presence —
3	Collision Presence +	4	Transmit—
5	Transmit +	6	N/C
7	N/C	8	Receive —
9	Receive +	10	Transceiver Power
11	Transceiver Power Return	12	N/C
13	N/C	14	N/C
15	N/C	16	N/C

Figure 2-1. NI1010A Transceiver Cable Connector

Mating connectors for P2 are 16-pin Berg 65846-01, 3M 3452, or an equivalent.

The total length of cable between the NI1010A and its transceiver must not exceed 50 meters (165 feet).

To assist in transceiver connection, Interlan offers an accessory 10 foot flat cable (Interlan part number AC-NM10-10) which is designed to interconnect twisted-pair Ethernet transceiver round-cable to the NI1010A's P2 connector. The flat cable, intended for flexible inter-connection in an internal cabinet environment, has greater loss than an equivalent length of the twisted-pair transceiver cable so the length of flat cable used should not exceed 3 meters (10 feet).

The transceiver end of the AC-NM10-10 Flat Cable has a 15-pin D subminiature female connector with a slide lock assembly (Cinch type DA 51220-1). Per the Ethernet Specification, pin assignments for the connector are as follows:

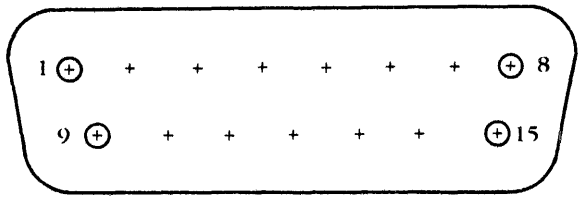


Figure 2-2. AC-NM10-10 Transceiver Cable Connector (Transceiver End)

1	Shield (see note)		
2	Collision Presence +	9	Collision Presence
3	Transmit +	10	Transmit -
4	Reserved (N/C)	11	Reserved (N/C)
5	Receive +	12	Receive -
6	Power Return	13	Power
7	Reserved (N/C)	14	Reserved (N/C)
8	Reserved (N/C)	15	Reserved (N/C)

N/C = no connection

NOTE: For proper electrical integrity and safety, the shield of the transceiver cable must be connected to the frame of the equipment enclosure, and the frame of the equipment enclosure must be connected to the safety ground (third wire) of the AC power line. The shield of the transceiver cable should not be connected to the NI1010A's logic ground, only to the frame of the cabinet.

2.2.3 NI1010A Ethernet Addresses

Each NI1010A controller has been assigned a physical address burned into an on-board programmable part. The user may change the physical address by issuing a "Load Physical Address" (see Chapter 3, Command Descriptions).

The NI1010A may be set up to allow for the reception of none, all, or up to 63 different multicast-group addresses. For further information on programming the NI1010A for accepting multicast-group addressed packets, see Chapter 3, Command Descriptions.

2.2.3.1 NI1010A Physical Address

A unique 48-bit physical address has been assigned to each NI1010A. This address value was selected from within a contiguous block of Ethernet physical addresses obtained by Interlan from Xerox Corporation through their Ethernet licensing arrangement. The physical address assigned to the board provides the station with a physical address that is distinct from the physical address of any other station on any Ethernet.

As a convenience, the board has a label displaying its Interlan assigned

physical address. The board's physical address may also be read under program control. See the description of the "Report and Reset Statistics" command in Chapter 3, Command Descriptions, for details.

You may change the physical address of the NI1010A by issuing the command "Load Physical Address" as described in Chapter 3, Command Descriptions.

The following are Interlan Ethernet physical addresses:

Physical Address Byte:	A	B	C	D	E	F	
start of physical address block:	02	07	01	00	00	00	(hex)
end of physical address block:	02	07	01	FF	FF	FF	
24 bits assigned by...	Xerox			Interlan			

2.2.3.3 Multicast Addresses

The Ethernet specification allows for a multiple-destination address, associated with one or more stations on a given Ethernet. There are two kinds of multiple-destination addresses:

- Multicast-group. An address associated by higher level convention with a group of logically related stations.
- Broadcast. An address associated with all stations.

The NI1010A recognizes the broadcast address and all, none, or up to 63 programmable multicast-group addresses. See Chapter 3, Command Descriptions, for a complete description on how to use the NI1010A multicast-group address recognition capability.

2.3 CONFIGURING THE NI1010A FOR THE UNIBUS

The user must properly configure the following UNIBUS parameters:

i) Device Base Address (BA);

ii) Interrupt Vector Address (VA); and

iii) Interrupt Bus Request Priority Level (BR).

2.3.1 NI1010A Factory Configuration

Each unit has the following configuration when shipped from the factory:

* Device Base Address (BA): 764000 (octal)
  - o Command and Status Register (CSR) = 764000
  - o Buffer Address Register (BAR) = 764002
  - o Byte Count Register (BCR) = 764004

* Interrupt Vector Address (VA): 340
  - o Receive DMA Done (RCV DONE) = 340
  - o Command Done (CMD DONE) = 344

* Interrupt Bus Request Priority Level (BR): BR5

Refer to Figure 2-4 for the positions of the dip switches U22 and U23.

2.3.2 Setting the Device Base Address (BA)

The NI1010A device base address is set by dip switches at U22 and U23 as follows:

NI1010A Register	UNIBUS Address
Command and Status Register (CSR)	7MXXN0
Buffer Address Register (BAR)	7MXXN2
Byte Count Register (BCR)	7MXXN4

Where: M may be set to either 6 or 7;
X may be set to either 0, 1, 2, 3, 4, 5, 6, or 7; and
N may be set to either 0, 2, 4, or 6.

The factory configuration of the Device Base Address is 764000. This address is illustrated in Figure 2-4 inside the boxed area identified as "Device Base Address".

Refer to Figure 2-4, for assigning each user-alterable bit of the Device Base Address with dip switches 1 through 8 at U22, and switch 7 at U23.

Note the closed switch position, ON, corresponds to a logic 0; the open switch position, OFF, corresponds to a logic 1.

CAUTION

Dip switch 8 at U23 MUST ALWAYS BE OFF (logic 1) for the NI1010A to properly function. Placing this switch in the ON position causes the NI1010A to enter a manufacturing test mode, making it unresponsive to commands from the UNIBUS host.

****

2.3.3 Setting the Interrupt Vector Address (VA)

The NI1010A has two interrupt vectors: Receive DMA Done (RCV DONE), and Command Done (CMD DONE). These interrupt vectors are switch selectable by means of the dip-switch located at U23. The Command Done vector is fixed at an address of four higher than the Receive DMA Done vector.

The factory configuration of the Interrupt Vector Address is 340. This address is illustrated in Figure 2-4 inside the boxed area identified as "Interrupt Vector Address".

Refer to Figure 2-4 for assigning each user-alterable bit of the Interrupt Vector Address with the dip switches 1 through 6 at U23.

2.3.4 Interrupt Bus Request Priority Selection

When shipped from the factory the NI1010A is strapped to the BR5 level by a jumper block inserted at U6. To reconfigure the NI1010A for BR4 operation the block should be removed, rotated 180 degrees, and inserted such that pins 1 and 16 of U6 are connected together.

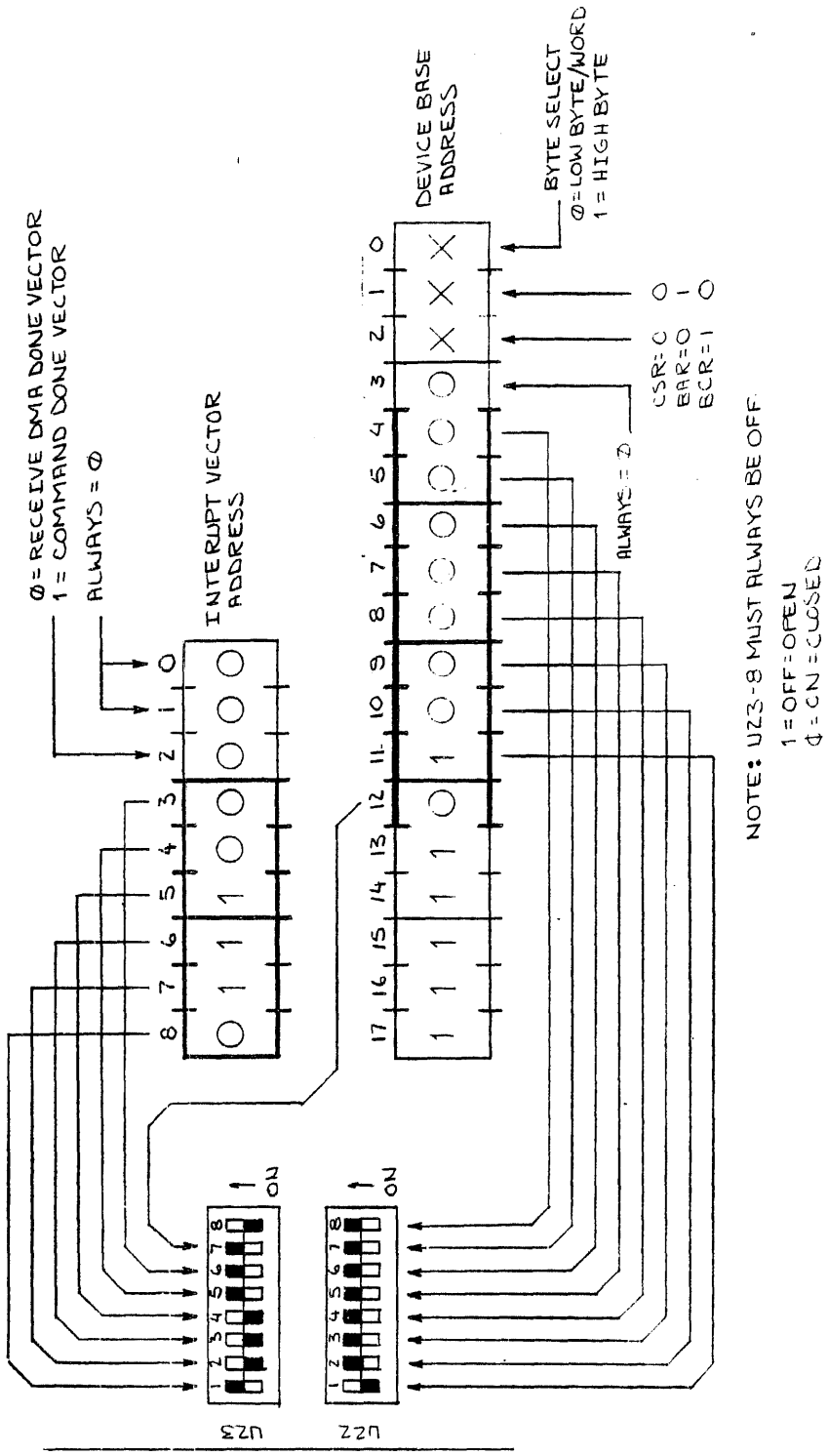


Figure 2-4. Factory configuration of NI1010A Device Base and Interrupt Vector Addresses

2.4 UNIBUS INSTALLATION

The NI1010A is a single width hex height board that is compatible with the Small Peripheral Controller (SPC) slots on a UNIBUS backpanel.

2.4.1 Unibus Backpanel Modification

The NI1010A is a DMA device controller and as such accesses the NPR/NPG request/grant lines on the UNIBUS. These lines, at backpanel pins CA1 and CB1, are frequently found jumpered to provide NPR grant continuity. This jumper, if installed, must be removed for the NI1010A to function properly.

NOTE: Check for, and remove, the wire wrap jumper between pins CA1 and CB1 before installing the NI1010A into a backpanel slot.

2.4.2 Power Requirements

The NI1010A and its Ethernet transceiver unit are powered directly from the UNIBUS backpanel and have the following power requirements:

i) +4.75 to +5.25 Vdc @ 7.5 A typical, 8.2 A maximum; for the NI1010A board (only); and,

ii) +11.4 to +15.75 Vdc @ 0.5 A maximum; for the transceiver unit (only).

The NI1010A provides power to its Ethernet transceiver unit from the +15Vdc power on the UNIBUS backpanel.

CAUTION

Before installing the NI1010A, verify that sufficient power supply capacity exists to provide to satisfy the current requirements of the NI1010A and its Ethernet transceiver unit. After installation, check that the system's +5Vdc and +15Vdc power supply voltages are still within the required voltage levels.

****

2.4.3 Environmental Requirements

The NI1010A has been designed to operate in a standard DEC enclosure with forced air cooling at an inlet air temperature of less than 50 degrees Celsius. Operation in free air (without forced air flow across the surface of the board) should only be done at room ambient temperatures below 25 degrees Celsius with the board away from objects that would impede convective air flow.

CAUTION

Under no circumstances should the board be operated (powered on) in a backpanel without forced air flow. Serious damage to the

board could result due to the concentration of heat that would likely occur.

****

The NI1010A may be operated in an environment with relative humidity up to 90% provided that moisture does not condense on the board.

2.5 NI1010A INSTALLATION CHECK LIST

1. Select Device Base Address (BA) _____
2. Select Interrupt Vector Address (VA) _____
3. Select Interrupt Bus Request Priority (BR) _____
4. Select UNIBUS SPC slot with adequate forced air ventilation _____
5. Check for, and remove, UNIBUS Backpanel Jumpers for NPR/NPG _____
6. Check Power Supply for Adequate Capacity _____
7. With the system power off, install the NI1010A into the selected UNIBUS SPC Slot _____
8. Connect the AC-NM10 Flat Cable to the NI1010A's P2 connector _____
9. Mount the grounding plate on the AC-NM10 Flat Cable to the host's chassis ----------
10. Connect the AC-NM10 Flat Cable to the Ethernet transceiver unit via a transceiver cable (AC-NA1010 or equiv) _____
11. Connect the Ethernet transceiver to the Ethernet coax cable _____

The NI1010A is now ready to be powered on.

When power is applied to the NI1010A, the Pass/Fail LED should light for approximately one-half (0.5) second, then go out. If the Pass/Fail LED should stay lit, the NM10A's power-on self-test has failed. Refer to Chapter 5, Maintenance, Section 5.2.2 Power-up or Reset Operation, for additional information.

Further verification of the NI1010A in a PDP-11 may be performed by executing the NI1010A's Standalone PDP-11 Diagnostics (DS-NI1010). For VAX-11 installations, the Diagnostic Program include with the NS2030 VMS Device Driver (NS2030) may be executed. Additional information for these diagnostic tests is contained in Chapter 5, Maintenance.

CHAPTER THREE

PROGRAMMING INFORMATION

Chapter 3 presents information required for a programmer to use the NI1010A UNIBUS Ethernet Communications Controller board. This chapter

i) presents the commands supported by the NI1010A;

ii) discusses the four basic NI1010A command categories that the commands fall into;

iii) describes in detail the NI1010A registers used as the programming interface for all commands; and

iv) provides detailed descriptions of each NI1010A command, status return, and required data format.

3.1 OVERVIEW

The NI1010A controller is a DMA (direct memory access) device that responds to commands issued by the host UNIBUS system. Command operations are initiated when the host system writes a 6-bit command function code to the NI1010A's Command and Status Register (CSR). The controller performs the specified command function and returns to the CSR a 4-bit command status code indicative of the command's operation. Table 3-1 lists the commands supported by the NI1010A, and Table 3-2 lists the controller's command status returns.

Table 3-1

NI1010A COMMAND FUNCTION CODES

COMMAND CODE (octal)	COMMAND FUNCTION	STATUS CODE RETURNED (octal)
00	Reserved	02
01	Set Module Interface Loopback Mode	00
02	Set Internal Loopback Mode	00
03	Clear Loopback Mode	00
04	Set Promiscuous Receive Mode	00
05	Clear Promiscuous Receive Mode	00
06	Set Receive-On-Error Mode	00
07	Clear Receive-On-Error Mode	00
10	Go Offline	00
11	Go Online	00
12	Run On-board Diagnostics	00,01,02,03,04*
13-14	Reserved- DO NOT USE	00
15	Set Insert Source Address Mode	00
16	Clear Insert Source Address Mode	00
17	Set Physical Address to Default	00
20	Set Receive All Multicast Packets	00
21	Clear Receive All Multicast Packets	00
22	Perform Network Loopback Test	00,01,03,04, 05,14,15,16
23	Perform Collision Detect Test	00,03,04, 05,10
24-27	Reserved	02
30	Report and Reset Statistics	00,17
31	Report Collision Delay Times	00,17
32	Reserved (Maintenance)	00,17
33-37	Reserved	02
40	Supply Receive Buffer	00,17
41-47	Reserved	Undefined
50	Load Transmit Data	00,05,17
51	Load Transmit Data and Send	00,01,03,04, 05,06,10,17
52	Load Group Address(es)	00,05,12,17
53	Delete Group Address(es)	00,05,12,17
54	Load Physical Address	00,12,17
55-57	Reserved	02
60	Flush Receive BAR/BCR Queue	00
61-67	Reserved	Undefined
70-76	Reserved	Undefined
77	Reset	00,01,02,03,04*

* Note: These are diagnostic status codes not command status codes. They have a different meaning than that found in Table 3-2.

Table 3-2

NI1010A COMMAND STATUS CODES

CODE (octal)	COMMAND STATUS
00	Success
01	Success With Retries
02	Illegal Command
03	Inappropriate Command
04	Failure
05	Buffer Size Exceeded
06	Frame Too Small
07	Reserved
10	Excessive Collisions
11	Reserved
12	Buffer Alignment Error
13	No Heartbeat Detected
14	No CRC Error Occurred
15	Inappropriate CRC Error
16	Last Data Byte Not Received Correctly
17	Non-Existent Memory

3.2 NI1010A COMMAND CATEGORIES

The NI1010A commands of Table 3-1 can be classified into four command-type categories. Each of these categories reflects the various exchange sequences that take place between the host system and the NI1010A. The NI1010A Command-Type Categories are:

Type I	NON-DMA COMMANDS Command function codes 00 through 27, 60 through 67, and 77.
Type II	COMMANDS WITH DMA DATA RETURNED Command function codes 30 through 37, and 70 through 76.
Type III	COMMANDS WITH DMA DATA SUPPLIED Command function codes 50 through 57.
Type IV	COMMANDS WITH DMA DATA ASYNCHRONOUSLY RETURNED Command function codes 40 through 47.

3.2.1 NI1010A Type I Commands: Non-DMA Commands

NI1010A commands with function codes 00 through 27, 60 through 67, and 77 perform without use of the NI1010A's onboard DMA controller. Non-DMA commands operate as follows:

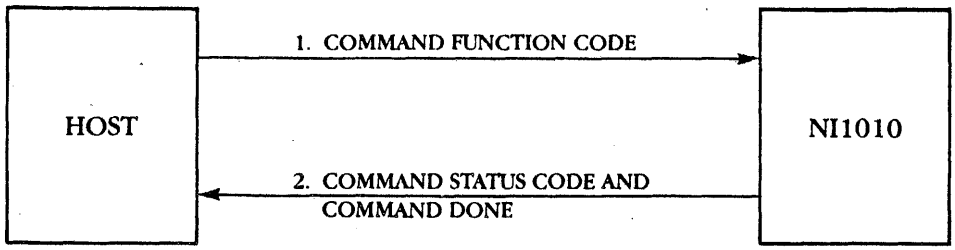


Figure 3-1

1. The host UNIBUS system writes a 6-bit command function code to the CSR register. The NI1010A decodes the command function code, executes the command, and,

2. returns a 4-bit command status code to the host via the CSR register, and sets the CSR's Command Done bit (CMD DONE).

NI1010A Type I commands are:

01	Set Module Interface Loopback Mode
02	Set Internal Loopback Mode
03	Clear Loopback Mode
04	Set Promiscuous Receive Mode
05	Clear Promiscuous Receive Mode
06	Set Receive-On-Error Mode
07	Clear Receive-On-Error Mode
10	Go Offline
11	Go Online
12	Run On-board Diagnostics
15	Set Insert Source Address Mode
16	Clear Insert Source Address Mode
17	Set Physical Address to Default
20	Set Receive All Multicast Packets
21	Clear Receive All Multicast Packets
22	Perform Network Loopback Test
23	Perform Collision Detect Test
60	Flush Receive BAR/BCR Queue
77	Reset

3.2.2 NI1010A Type II Commands: Commands with DMA Data Returned

NI1010A commands with function codes 30 through 37, and 70 through 76, are DMA commands that transfer NI1010A controller data into the host's UNIBUS memory. The data returned to the host is synchronous with the issuance of the command, ie., writing the 6-bit command function code to the CSR initiates the DMA data transfer (this is in contrast to an asynchronous return of data which occurs with Type IV commands). Commands with DMA data synchronously returned operate as follows:

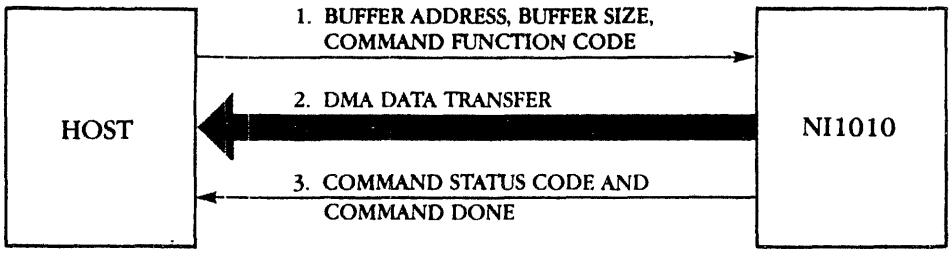


Figure 3-2.

1. The host system writes a buffer address and buffer size to the Buffer Address Register (BAR) and Byte Count Register (BCR) respectively, and then the 6-bit command function code to the CSR.

2. The NI1010A decodes the command and performs a DMA data transfer into UNIBUS memory with the appropriate controller data.

3. at completion of the DMA data transfer, a 4-bit command status code is returned to the CSR, and the Command Done bit is set.

NI1010A Type II commands are:

30	Report and Reset Statistics
31	Report Collision Delay Times
32	Reserved (maintenance)

3.2.3 NI1010A Type III Commands: Commands with DMA Data Supplied

NI1010A commands with function codes 50 though 57 are DMA commands that transfer host data in UNIBUS memory to the controller. As with Type II commands, DMA controller operation is synchronous with the writing of the command function code to the CSR. Commands with DMA data supplied operate as follows:

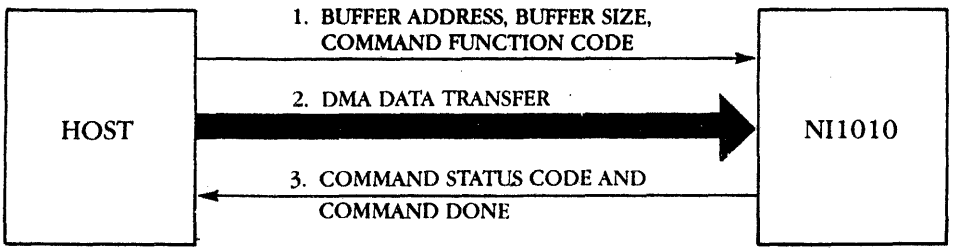


Figure 3-3.

1. The host system writes a buffer address and buffer size to the BAR and BCR respectively, and then the 6-bit command function code to the CSR.

2. The NI1010A decodes the command function code and performs a DMA transfer of the UNIBUS buffer to the controller.

3. At completion of the DMA transfer, a 4-bit command status code is returned in the CSR, and the Command Done bit set.

NI1010A Type III commands are:

50	Load Transmit Data
51	Load Transmit Data and Send
52	Load Group Address(es)
53	Delete Group Address(es)
54	Load Physical Address

3.2.4 NI1010A Type IV Commands: Commands with DMA Data Asynchronously Returned

NI1010A commands with command function codes 40 through 47 perform DMA data transfers in an asynchronous manner. When the host system issues a Type IV command it only "arms" the DMA controller with a UNIBUS memory address and buffer byte count, it does not initiate a DMA transfer. DMA transfers are initiated when an external event (eg., a frame has been received) has occurred.

The NI1010A supports only one Type IV command; Supply Receive Buffer (40). Asynchronous DMA data transfers for this command operate as follows:

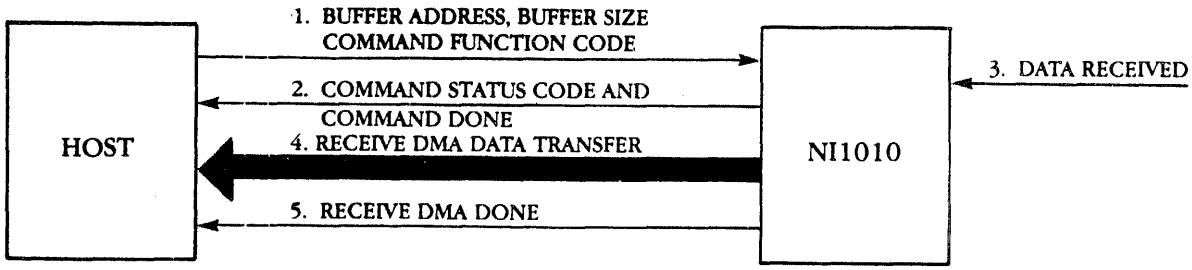


Figure 3-4.

1. The host system writes a receive buffer address and buffer size into BAR and BCR respectively, and then the 6-bit command functi. code (40) to the CSR.

2. The NI1010A checks for non-existent memory at the first word of the buffer pointed to by BAR by doing a one word write, then the NI1010A stores the BAR and BCR values at the end of the controller's Receive BAR/BCR Queue and returns a 4-bit command status code in the CSR, and sets the Command Done bit. The controller is now "armed".

3. If no receive data is currently in the Receive FIFO buffer, the controller waits for receive data to come in from the network. (while waiting for receive data additional controller commands may be issued) Once receive data is available,

4. The DMA controller takes the first buffer available in FIFO order from the Receive BAR/BCR Queue, and DMA's the receive data into UNIBUS memory.

5. At the end of the DMA transfer, the Receive DMA Done bit (RCV DONE) is set in the CSR.

3.3 PROGRAMMING INTERFACE

The NI1010A has three UNIBUS registers in the I/O page that i) accept command function codes, buffer addresses, and buffer size values from the host system, and ii) return status codes and done flags. These registers are

- Command and Status Register (CSR)
- Buffer Address Register (BAR)
- Byte Count Register (BCR)

3.3.1 Command and Status Register (CSR)

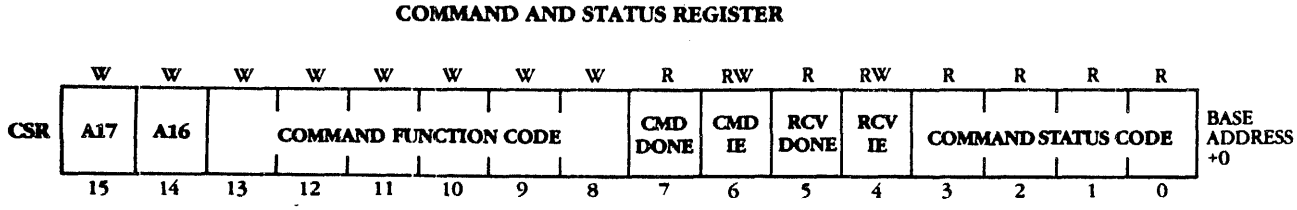


Figure 3-5. Command and Status Register

A17,A16:
A17 and A16 are write-only address extension bits for forming an 18-bit Buffer Address Register (BAR) value. These bits must be written concurrent with the writing of a command function code. Reading these bits returns an undefined value.

Command Function Code:
The command function codes of Table 3-1 are written to this field. When this field is written the command is executed by the controller. Reading this field returns an undefined value.

NOTE: Writing a command function code to this field before a previous command's Command Done (CMD DONE) is returned may result in improper NI1010A operation. An exception to this is the Reset (77) command which may be issued at any time.

CMD DONE (Command Done):
The Command Done bit is set by the NI1010A when a controller command has been completed. It is reset to zero when CSR<7:0> is read by the host. Writing to this bit has no effect.

CMD IE (Command Done Interrupt Enable):
Writing a one to this bit enables an interrupt request when the CMD DONE bit becomes set. Writing zero to this bit disables CMD DONE interrupt service requests, cancelling any pending interrupt request.

RCV DONE (Receive DMA Done):
The Receive DMA Done bit is set to one by the NI1010A controller after the last byte of a received frame has been transfered or the last byte in the buffer has been loaded. This bit is reset to zero when CSR<7:0> is read. Writing to this bit has no effect.

RCV IE (Receive DMA Done Interrupt Enable):
Writing a one to this bit enables

an interrupt request when the RCV DONE bit is set to one. Writing zero to this disables RCV DONE interrupts, cancelling any pending interrupt request.

Command Status Code:
The command status codes of Table 3-2 are returned to this field when CMD DONE is set to one. Writing to this field has no effect.

3.3.2 Buffer Address Register (BAR)

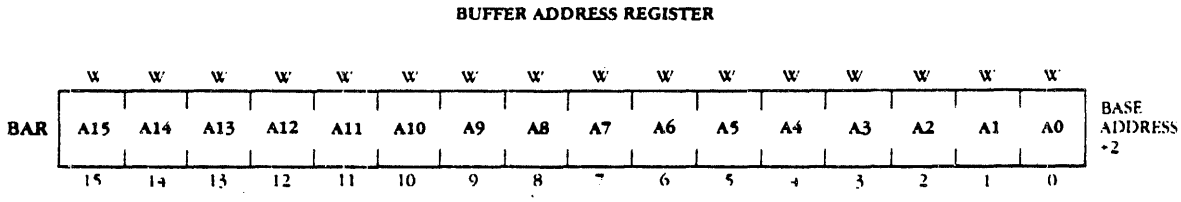


Figure 3-6. Buffer Address Register

The Buffer Address Register is a write-only register into which the low 16-bits of the base address of a UNIBUS memory buffer are written for use by the NI1010A's DMA controller. Reading BAR returns an undefined value. NOTE: BAR (and BCR, below) must be initialized prior to issuing any of the Type II, III, or IV commands.

3.3.3 Byte Count Register (BCR)

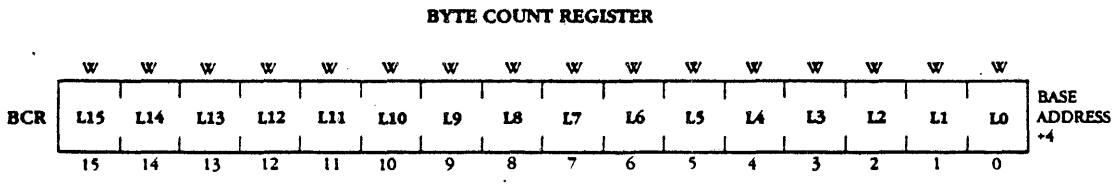


Figure 3-7. Byte Count Register

The Byte Count Register is a write-only register into which a binary value for the byte length of a UNIBUS buffer is written for use by the NI1010A's DMA controller. Issuing a byte count value of 0 returns a Non-Existent Memory (17) status code. Reading BCR returns an undefined value.

3.4 NI1010A COMMAND DESCRIPTIONS

This section describes each NI1010A command in detail. Presented for each command is:

i) a description of the command's function,

ii) the sequence required to invoke the command,

iii) a description of all possible status codes returned in the CSR register,

iv) the required data format, and

v) any special notes on the command.

NOTE: All command function codes, status codes, and buffer offset pointers are in octal notation unless otherwise noted.

Reserved (00)

3.4.1 RESERVED (00) Command Description:

Command function code 00 is functionless and has been reserved by Interlan.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

02 - Illegal Command; command function code 00 is functionless and disallowed for use.

Notes: None.

3.4.2 SET MODULE INTERFACE LOOPBACK MODE (01)

Command Description:

The Set Module Interface Loopback Mode command causes the NI1010A to enter a data loopback mode whereby all frames transmitted by the host are entered into the module's transmit buffer, moved to the receive buffer and presented as received frames awaiting receive DMA transfer back to the host. When placed in this mode the NI1010A goes into an offline operational state where it is unable to transmit and receive network traffic.

This command is useful for testing the NI1010A's UNIBUS hardware, DMA controller, and the NM10A's processor system (see Chapter 4, FUNCTIONAL DESCRIPTION). It does not, however, exercise any of the NM10A's Ethernet link management and physical channel logic.

Command Sequence:

1. Move command function code 01 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A is now in Module Interface Loopback mode and in an offline state.

Data Format:

When in this loopback mode all receive frames are identical to transmit frames.

Notes:

When in any data loopback mode the NM10A's CRC generating logic is not used. Consequently, the received frames will not contain a valid CRC value.

When in Internal Loopback Mode the Pass/Fail LED indicator turns on.

3.4.3 SET INTERNAL LOOPBACK MODE (02)

Command Description:

This command causes the NI1010A to enter a data loopback mode that returns the transmit bit-stream through the receiver, just prior to going out to the transceiver cable. The packet is processed for return back to the host. When in this mode the NI1010A goes into an offline operational state where it is unable to transmit and receive network traffic.

This command is useful for testing the integrity of the NM10A's transmit and receive memory. It does not, however, test any of the NM10A's Ethernet link management and physical channel logic.

Command Sequence:

1. Move command function code 02 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A is now in Internal Loopback Mode and offline.

Data Format:

Transmitted frames will be returned as received frames IF they meet the conditions currently established in the module (e.g., physical or multicast address match unless "Set Promiscuous Mode" has been previously issued).

Notes:

When in any data loopback mode the NM10A's CRC generating logic is not used. Consequently, the received frames will not contain a valid CRC value.

When in Internal Loopback Mode the Pass/Fail LED indicator turns on.

Data in the Receive FIFO prior to issuance of this command is not affected.

3.4.4 CLEAR LOOPBACK MODE (03)

Command Description:

The Clear Loopback Mode command clears either the data loopback modes previously set by the Set Module Interface Loopback Mode command (01) or the Set Internal Loopback Mode command (02).

Command Sequence:

1. Move command function code 03 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A is out of loopback mode

Notes:

The Pass/Fail LED indicator is OFF after execution of this command.

3.4.5 SET PROMISCUOUS RECEIVE MODE (04)

Command Description:

Issuing the Set Promiscuous Receive Mode command permits the NI1010A controller to receive all valid frames (ie, not collision fragments) which appear on the network, regardless of the value in the frame's destination address field.

This command is often found useful for performing network-level diagnostic and maintenance functions. Command Sequence:

1. Move command function code 04 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A is now in Promiscuous Receive mode.

Notes:

When in this mode the controller will receive all of its own transmissions, though it does not perform CRC error checking on them, reporting a status of no CRC error.

When in any data loopback mode the NM10A's CRC generating logic is not used. Consequently, the received frames will not contain a valid CRC value.

3.4.6 CLEAR PROMISCUOUS RECEIVE MODE (05)

Command Description:

The Clear Promiscuous Receive Mode command resets the controller to a state where it only receives those frames on the network which have a destination address value that matches either the NI1010A's physical address, one of the multicast-group addresses assigned by the user to the controller, or the network broadcast address.

Command Sequence:

1. Move command function code 05 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A is now in Normal Receive mode (ie, out of Promiscuous Receive mode).

Notes:

Frames received during the time the controller was in promiscuous receive mode may still reside in the Receive FIFO buffer. This command does not invalidate any of the contents of this buffer.

3.4.7 SET RECEIVE-ON-ERROR MODE (06)

Command Description:

The Set Receive-On-Error Mode command permits the NI1010A to receive frames with CRC errors and/or alignment errors. Frames containing these errors are flagged in the Frame Status byte located at the header of the receive packet.

Command Sequence:

1. Move command function code 06 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A will now accept all received frames with CRC and/or Alignment errors.

Notes:

When not in this mode the NI1010A will filter out all frames received with CRC and/or alignment errors.

3.4.8 CLEAR RECEIVE-ON-ERROR MODE

Command Description:

Issuing the Clear Receive-On-Error Mode command informs the NI1010A controller to filter-out (ie, reject) all frames received with CRC and/or alignment errors.

Command Sequence:

1. Move command function code 07 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A will now reject all received frames with CRC and/or Alignment errors.

Notes:

Frames previously received with either CRC or alignment errors may still be in the Receive FIFO after issuance of this command.

Frames received from the network with CRC or alignment errors will not be delivered to the host but will be counted by the statistics counters.

3.4.9 GO OFFLINE (10)

Command Description:

The Go Offline command logically disconnects the NI1010A's transmitter and receiver from the network.

The host is unable to transmit and receive frames from the network.

Command Sequence:

1. Move command function code 10 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A's transmitter and receiver are now "disconnected" from the network.

Notes:

Received frames pending in the Receive FIFO buffer are not affected by this command.

The Go Offline command does not interrupt the NI1010A receiver. In the event that a frame is currently being received from the network it will be received completely.

An "Inappropriate Command" status code is returned should a Load Transmit Data and Send command (51) be issued when the NI1010A is in an offline state.

If either Module Loopback or Internal Loopback have been previously set, issuing this command will have no effect.

3.4.10 GO ONLINE (11)

Command Description:

The Go Online command is used to logically connect the NI1010A's transmitter and receiver onto the Ethernet.

Command Sequence:

1. Move command function code 11 into CSR<13:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A is now "connected" to the Network.

Notes:

The Go Online command should be issued after the controller has been powered up, or after issuing either the Clear Loopback Mode (03) or Run On-Board Diagnostics (12) commands.

The Pass/Fail LED will be OFF while the controller is online.

3.4.11 RUN ON-BOARD DIAGNOSTICS (12)

Command Description:

Issuing the Run On-board Diagnostics command causes the NI1010A's NM10A to an execute an onboard diagnostic routine, and return a diagnostic status code. The sequence of tests performed by the on-board diagnostics is described in section 5.2.1.

Command Sequence:

1. Move command function code 12 into CSR<13:8>

Possible Diagnostic Status Codes Returned in CSR<3:0>:

00 - Success; the onboard diagnostics executed without detecting any faults on the NM10A module.

01 - ROM Checksum Error;

02 - Memory Error; an error was detected in the NM10A read/write memory.

03 - Address Error; an error was detected while attempting to read the board's Ethernet address.

04 - Loopback Failure; the NM10A was unsuccessful at transmitting a test frame while in internal loopback mode.

05 - Carrier Sense Failure; Internal loopback test could not be completed due to Ethernet cable failure.

NOTE: These codes are Diagnostics Status Codes, NOT Command Status Codes. These diagnostic status codes are unique to this command.

Notes:

All information in the transmit and receive memories is destroyed.

During execution of the diagnostic tests the Pass/Fail LED indicator will be on, and will turn off only if all tests have been successfully executed.

The NM10A is left in the reset state after successfully performing these diagnostic routines. A "RESET" command should be issued before attempting to use the board if a failure status code is returned.

3.4.12 RESERVED (13-14)

Command Description:

Command function codes 13 and 14 are reserved for Interlan use. Do not issue them.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Interlan has reserved these command codes

Notes:

Inadvertent use of these command codes may cause unexpected behavior. Issue a RESET command to resume known state.

3.4.13 SET INSERT SOURCE ADDRESS MODE (15)

Command Description:

This command causes the NI1010A to insert its physical address into the source address field of an Ethernet frame prior to its being transmitted onto the coaxial cable. The frame format required by this mode for transmitting onto the Ethernet is shown in Figure 3-12.

Command Sequence:

Move command function code 15 into CSR<13:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A will insert its physical address as the source address of any frame being transmitted onto the Ethernet.

Notes:

This mode is assumed by the NI1010A as part of its initialization process which is invoked after power-up, or as a consequence of the execution of the Reset (77) or Run On-Board Diagnostics (12) commands.

After the issuance of this command, any data in the transmit memory will be lost.

The NI1010A's source address is either the factory-programmed Ethernet address, or one previously loaded by the user. See "Set Physical Address to Default", section 3.4.15, and "Load Physical Address", section 3.4.29.

3.4.14 CLEAR INSERT SOURCE ADDRESS MODE (16)

Command Description:

This command will disable the insertion of the NI1010A's physical address as the source address prior to transmitting a frame onto the Ethernet. In this mode, the user must include the source address as part of a frame to be transmitted.

Command Sequence:

Move command function code 16 into CSR<13:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A will not insert its physical address as the source address of subsequently transmitted frames; it must be provided by the user.

Notes:

After the issuance of this command, any data in the transmit memory will be lost.

The receiver is not affected by this command. A physical address other than the factory-programmed address may be specified by the Load Physical Address (54), section 3.4.29.

3.4.15 SET PHYSICAL ADDRESS TO DEFAULT (17)

Command Description:

This command causes the NI1010A to use its factory-programmed Ethernet address, cancelling out the effect of any other address previously provided by the user (see section 3.4.29 Load Physical Address (54)). While the NI1010A is in this state, the physical address checking hardware will prevent packets whose destination address do not match that of the pre-programmed Ethernet address from consuming space in the NM10A's receiver FIFO memory. This is the most efficient method of using the receiver.

Command Sequence:

Move the command function code 17 into CSR<13:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A will use its factory-programmed physical address for all subsequent operations, enabling its receiver physical address recognition hardware.

Notes:

This mode is assumed by the NI1010A as part of its initialization process which is invoked after power-up, or as a consequence of the execution of the Reset (77) or Run On-Board Diagnostics (12) commands. .s Issuing the Report and Reset Statistics command (30) at any time after this command has been issued will cause the NI1010A to return its pre-programmed Ethernet address as its source address.

As mentioned above, the receiver's physical address recognition hardware is enabled when the default (factory-programmed) address is used as the NI1010A's Ethernet address. The receiver's FIFO is therefore used in the most efficient manner.

3.4.16 SET RECEIVE ALL MULTICAST PACKETS (20)

Command Description:

This command allows the NI1010A to receive all packets from the Ethernet whose destination addresses are multicast. The group address recognition filtering process is bypassed while the NI1010A is operating in this mode. The user is responsible for filtering all multicast packets while the NI1010A is operating in this mode.

Command Sequence:

Move the command code 20 into CSR<13:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A will accept all multicast packets from the network if the Go Online command is or has been issued, or from the user if the NI1010A has been placed in the Internal Loopback mode.

Notes:

On a large network with numerous "server" devices, placing the NI1010A into this mode may impose considerable burden on the host's receive buffers by forcing the host to deal with all network multicast packets.

3.4.17 CLEAR RECEIVE ALL MULTICAST PACKETS (21)

Command Description:

Issuing this command causes the NI1010A to filter all received multicast packets by first hashing the destination address, then comparing the packet's destination address byte for byte with one or more entries in the group address recognition table. This command countermands the effect of the previous Set Receive All Multicast Packets (20) command.

Command Sequence:

Move command function code 21 into CSR<13:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the NI1010A will accept multicast packets from the network or internally looped back packets only if the packet's destination address matches one of those previously loaded in the group address recognition table.

Notes:

This mode is assumed by the NI1010A as part of its initialization process which is invoked after power-up, or as a consequence of the execution of the Reset (77) or Run On-Board Diagnostics (12) commands.

The group address recognition table may contain up to 63 entries. If more multicast addresses need to be recognized, the host may issue the Set Receive All Multicast Packets (20) command described previously.

3.4.18 PERFORM NETWORK LOOPBACK TEST (22)

Command Description:

Issuing this command causes the NI1010A to transmit two minimum length Ethernet packets with source address and destination address of 2 7 1 0 0 0. The first of these packets is sent with an incorrect CRC byte appended, causing a CRC error at all receiving stations. The second packet is sent with valid CRC bytes appended. The NI1010A receives its own transmission, checks each packet for valid or invalid CRC as appropriate, and checks that the last byte received matches the last byte transmitted.

This command is useful for checking the NM10A's transmit and receive bit logic, the CRC checker and the Ethernet connection.

Command Sequence:

Move command function code 12 into C_REG.

Command Status Codes Returned in S_REG:

00 - Success; the NM10A has transmitted and received the correct data through the Ethernet transceiver.

01 - Success with Retries; the NM10A encountered one or more collisions before completing this transmit/receive test.

03 - Inappropriate Command Error; the NI1010A must be Online when this command is issued.

04 - Transmitter Failure; the module failed to transmit the packet onto the Ethernet because the processor's "jabber control" timer expired. This indicates a serious network system problem that should be investigated. Potential sources of this symptom include: malfunction of the local transceiver by its continuously asserting carrier sense; a remote transmitter "babbling" endlessly; a faulty transceiver cable.

05 - Carrier Sense Failure; Internal loopback test could not be completed due to Ethernet cable failure.

08 - Excessive Collisions Error; the NI1010A was unable to transmit after sixteen attempts were halted by collisions.

0C - CRC Detect Error; a valid CRC was reported after transmitting a packet containing a CRC error.

0D - Received Data Error; the received data did not match the transmitted data.

0E - CRC Detect Error; a CRC error was reported after transmitting a packet containing valid CRC bytes.

Notes:

The NI1010A must be online and connected to an Ethernet transceiver on a valid Ethernet to successfully execute this command.

3.4.19 PERFORM COLLISION DETECT TEST (23)

Command Description:

Issuing this command causes the NI1010A to transmit a minimum length packet trhough the transceiver onto the Ethernet. The transceiver is expected to assert the collision detect signal approximately 1 microsecond after the last transmitted bit.

This command is useful to test that the transceiver's collision detect hardware is working properly.

Command Sequence:

Move command function code 13 into C_REG.

Command Status Codes Returned in S_REG:

00 - Success; the NI1010A detected the heartbeat signal from the Ethernet transceiver after transmission of a packet.

0B - No Heartbeat Error; No heartbeat was detected after transmission of a packet.

Notes:

The NI1010A must be online and connected to an Ethernet transceiver which generates a heartbeat signal to successfully execute this command.

3.4.20 RESERVED (24-27)

Command Description:

Command function codes 24 through 27 are reserved by Interlan for future use. Do not issue them.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

02 - Illegal Command; Interlan has reserved these command codes

Notes: None.

3.4.21 REPORT AND RESET STATISTICS (30)

Command Description:

The Report and Reset Statistics command causes the NI1010A to return network statistical data and controller board identification information. The contents of all statistical counters (except Number of Frames in Receive FIFO) are reset to zero upon completion of this command.

All statistics are represented as 16-bit unsigned integers, except where noted.

Overflow of any statistical quantity is not reported.

Command Sequence:

1. Move low 16-bits of an 18-bit UNIBUS buffer address into BAR<15:0>

2. Move the byte count of 66 (decimal) into BCR<15:0>

3. Concatenate the high 2-bits of the 18-bit UNIBUS buffer address with the command function code 30 and move it into CSR<15:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the statistical data and identification information has been DMA transfered, and the statistical counters reset to zero.

17 - Non-Existent Memory; read-only or non-existent memory at UNIBUS memory pointed to by BAR

Data Format:

See Figure 3-8 on the next page.

15 8	7 0	
0	0	← BAR +0
LENGTH		+2
PHYSICAL ADDR <B>	PHYSICAL ADDR <A>	+4
PHYSICAL ADDR <D>	PHYSICAL ADDR <C>	+6
PHYSICAL ADDR <F>	PHYSICAL ADDR <E>	+8
NUMBER OF FRAMES RECEIVED		+10
NUMBER OF FRAMES IN RECEIVE FIFO		+12
NUMBER OF FRAMES TRANSMITTED		+14
NUMBER OF EXCESS COLLISIONS		+16
NUMBER OF COLLISION FRAGMENTS RECEIVED		+18
NUMBER OF TIMES 1 OR MORE FRAMES LOST		+20
NUMBER OF MULTICAST FRAMES ACCEPTED		+22
NUMBER OF MULTICAST FRAMES REJECTED		+24
NUMBER OF FRAMES RECEIVED WITH CRC ERROR		+26
NUMBER OF FRAMES RECEIVED WITH ALIGNMENT ERROR		+28
NUMBER OF COLLISIONS		+30
NUMBER OF OUT-OF-WINDOW COLLISIONS		+32
(RESERVED FOR FUTURE USE)		+34
(RESERVED FOR FUTURE USE)		+36
(RESERVED FOR FUTURE USE)		+38
(RESERVED FOR FUTURE USE)		+40
(RESERVED FOR FUTURE USE)		+42
(RESERVED FOR FUTURE USE)		+44
(RESERVED FOR FUTURE USE)		+46
(RESERVED FOR FUTURE USE)		+48
MODULE ID <B>	MODULE ID <A>	+50
MODULE ID <D>	MODULE ID <C>	+52
MODULE ID <F>	MODULE ID <E>	+54
<NUL>	MODULE ID <G>	+56
FIRMWARE ID <B>	FIRMWARE ID <A>	+58
FIRMWARE ID <D>	FIRMWARE ID <C>	+60
FIRMWARE ID <F>	FIRMWARE ID <E>	+62
<NUL>	FIRMWARE ID <G>	← BAR +64

Figure 3-8.

LENGTH: a binary value for the number of bytes that follow (62 decimal).

PHYSICAL ADDR: the NI1010A's current physical address (6 bytes).

NUMBER OF FRAMES RECEIVED: a binary value for the number of frames that were received on the network. This also includes multicast-group packets that were filtered by the controller.

NUMBER OF FRAMES IN RECEIVE FIFO: the number of frames currently residing in the Receive FIFO buffer.

NUMBER OF FRAMES TRANSMITTED: the number of frames transmitted with or without retries onto the network.

NUMBER OF EXCESS COLLISIONS: a binary value for the number of times a transmit frame incurred 16 successive collisions when attempting access to the network.

NUMBER OF COLLISION FRAGMENTS RECEIVED: a binary value for the number of collison fragments (ie, "runt packets") that were received and filtered by the controller.

NUMBER OF FRAMES LOST: a binary value for the number of frames that the Receive FIFO did not have sufficient buffer space available to accept.

NUMBER OF MULTICAST FRAMES ACCEPTED: a binary value for the number of frames received on the network containing a multicast-group destination address matching one of those assigned to the controller.

NUMBER OF MULTICAST FRAMES REJECTED: a binary value for the number of frames detected on the network with a multicast-group destination address value not matching one of those assigned to the controller.

NUMBER OF FRAMES RECEIVED WITH CRC ERROR: a binary value for the number of frames received by the controller containing a CRC error. (These frames will be delivered to the host only if the controller is operated in "Receive-On-Error" mode.)

NUMBER OF FRAMES RECEIVED WITH ALIGNMENT ERROR: a binary value for the number of frames received by the controller containing an alignment error (ie, the frame length was not an integral multiple of 8-bits). (These frames will be delivered to the host only if the controller is operated in "Receive-On-Error" mode.)

NUMBER OF COLLISIONS: a binary value for the number of collisions incurred by the controller when transmitting frames onto

the network.

NUMBER OF OUT-OF-WINDOW COLLISIONS: a binary value for the number of "out-of-window" (ie, beyond the 51.2 uSec slot time) collisions incurred by the controller when transmitting frames onto the network.

RESERVED FOR FUTURE USE: these 8 words read zero.

MODULE ID: these 8 ASCII bytes identify the module on the NI1010A motherboard. (eg. "NM10A<SP><SP><SP><NUL>")

FIRMWARE ID: these 8 ASCII bytes identify the firmware version installed in the NI1010A's NM10A module. (e.g."V03.01<SP><NUL>").

Notes:

The statistical counters are reset to zero (except Number of Frames in Receive FIFO) after issuance of either this command or the Reset (77) command, and when the NI1010A is powered-up.

3.4.22 REPORT COLLISION DELAY TIMES (31)

Command Description:

This command returns timing information on collisions incurred by the controller during transmission of the last frame. Up to 16 time delays are measure from the start of a frames's transmission onto the network until the sensing of Collision Presence in the transceiver.

Because an open Ethernet transmission cable causes the transmitting station's frame to "collide" with itself (due to signal reflection), the collision delay time information returned by this command permits the host system to ascertain whether or not a catastrophic (ie, open cable) network fault has occurred. In such case all collision delay times would be identical. This time domain reflectrometry data can be used to calculate roughly how far away a cable fault may be from the station's transceiver.

Command Sequence:

1. Move the low 16-bits of an 18-bit UNIBUS buffer address into BAR<15:0>

2. Move 36 (decimal) into BCR<15:0>

3. Concatenate the high 2-bits of the 18-bit UNIBUS buffer address with the command function code 31 and move it into CSR<15:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the collision delay data has been DMA transfered to the host.

17 - Non-Existent Memory; read-only or non-existent UNIBUS memory resides at UNIBUS memory pointed to by BAR. The command was aborted.

Data Format:

See Figure 3-9 on the next page.

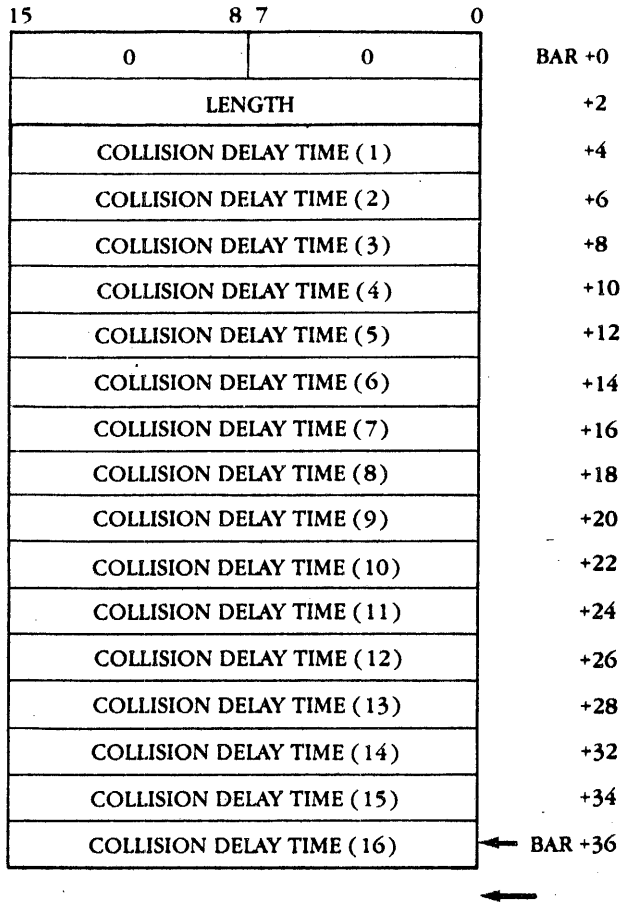


Figure 3-9.

LENGTH: a binary value for the number of bytes that follow. This value, ranging from 0 to 32 (decimal), is two times the numbe of collisions incurred during transmission of the last frame.

COLLISION DELAY TIME (N): a binary value for the number of bits transmitted before the transceiver's collision presence signal was sensed. The NI1010A has a timing resolution of 8 bits. (One bit represents 100 nanoseconds of transmission time.)

Notes:

The propagation time of an Ethernet frame on the transmission cable is approximately 5 nanoseconds per meter, and the NI1010A's TDR resolution is 800 nanoseconds. Consequently, this command provides cable fault resolution to 80 meters. (Note: the Collision Delay Times includes the transceiver's inherent delay in its collision delay circuit. This delay must be taken into consideration whenever attempting to locate a cable fault with this data.)

3.4.23 RESERVED (32)

Command Description:

Command function code 32 is a maintenance command reserved for Interlan manufacturing. You should not attempt to make use of this command. If, however, you should accidentally issue this command, you should be aware that this command will attempt to transfer up to 2048 (decimal) bytes of data to the buffer specified by the contents of the BAR and BCR.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; up to 2048 (decimal) bytes were transferred to host memory.

17 - Non-Existent Memory; read-only or non-existent UNIBUS memory resides at UNIBUS memory pointed to by BAR. The command was aborted.

Notes:

None.

3.4.24 RESERVED (33-37)

Command Description:

Command function codes 33 through 37 are reserved by Interlan for future use. Do not issue them.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

02 - Illegal Command; Interlan has reserved these command codes.

Notes:

None.

3.4.25 SUPPLY RECEIVE BUFFER (40)

Command Description:

The Supply Receive Buffer command loads a buffer address and buffer size value into the NI1010A's Receive BAR/BCR Queue. These values indicate to the DMA controller where to transfer receive data into UNIBUS memory. The Receive BAR/BCR Queue permits up to 16 Supply Receive Buffer commands to be outstanding. When the Supply Receive Buffer command is issued, its BAR/BCR value is entered at the end of the queue. When receive data becomes available, the DMA controller takes the first buffer entry from the beginning of the queue, and then transfers the receive data into the supplied receive buffer area into UNIBUS memory.

Command Sequence:

1. Move low 16-bits of an 18-bit buffer address into BAR<15:0>

2. Move the buffer's byte size into BCR<15:0>

3. Concatenate the high 2-bits of the 18-bit buffer address with the command function code 40 and move it into CSR<15:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the supplied BAR and BCR values have been entered at the end of the Receive BAR/BCR queue.

17 - Non-Existent Memory; read-only or non-existent memory resides at the first word of the UNIBUS buffer pointed to by BAR. The command was aborted by the NI1010A.

Data Format:

When the supplied BCR value is larger than the received frame then the following figure illustrates the format of data received. When BCR is less than the received frame size, and is a multiple of 8 bytes (ie., 4 words), then "buffer chaining" occurs. See Figure 4-6 for an illustration of buffer chaining.

See Figure 3-10 on the next page.

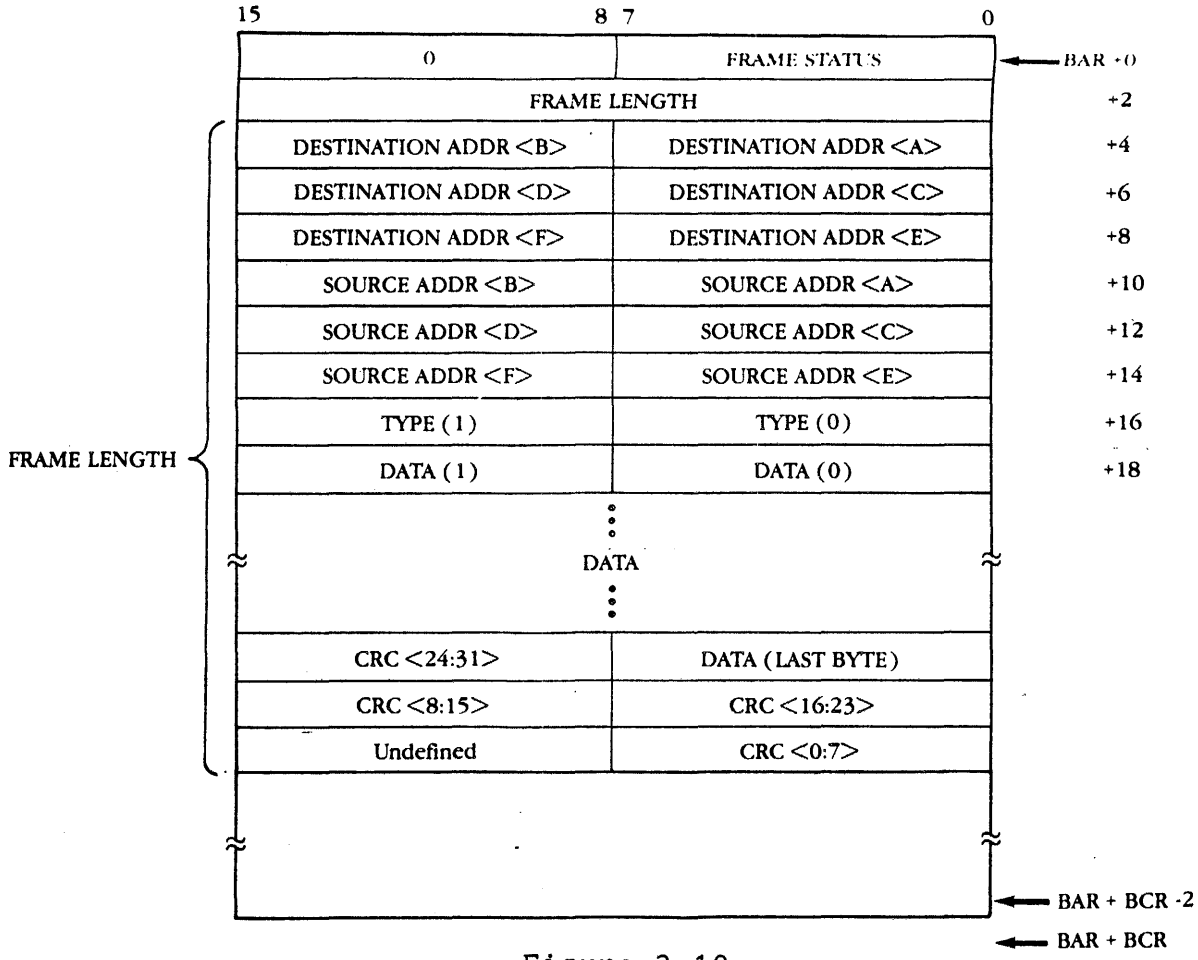


Figure 3-10.

FRAME STATUS:

7	6	5	4	3	2	1	0
0	0	0	0	0	L	A	C

Figure 3-11. Received Frame Status

L - Lost; set to one if one or more frames were lost immediately prior to reception of this frame

A - Alignment Error; set to one if during reception of this frame an alignment error was detected

C - CRC Error; set to one if during reception of this frame a CRC error was detected

FRAME LENGTH: a binary value for number of bytes contained in the received frame. This number includes the Destination Address

field, Source Address field, Type field, Data field, and th. CRC field. The Frame Length number does not include the undefined filler byte found in odd length frames. The Frame Length ranges between 64 and 1518.

DESTINATION ADDR: the frame's 48-bit (6-byte) destination address that was recognized by the controller

SOURCE ADDR: the physical address of the frame's sending station

TYPE: the type field of the frame

DATA: the variable length data field (minimum of 46 bytes, maximum of 1500 bytes)

CRC: the frame's Frame Check Sequence field's CRC value.(Note: the CRC polynomial bits are in reverse order.)

Notes:

Typically the Supply Receive Buffer command executes in less than 10 microseconds permitting polling operations to be used instead of Command Done interrupt service. However, should the controller be busy doing a receive DMA transfer, a significantly longer time may elapse before Command Done is set. For this reason, if Command Done polling is used with this command, it should be backed-up with Command Done interrupt service after 10 microseconds.

A "lost" frame occurs whenever there is insufficient buffer space in the Receive FIFO. The frame may or may not have been intended for the controller. This information is historical in that the "lost" frame appeared sometime in the past. The present frame is not affected in any way by this bit.

Odd length frames return an "Undefined" filler byte. This byte is NOT included in the Frame Length count.

3.4.26 RESERVED (41-47)

Command Description:

Command function codes 41 through 47 are reserved for future use by Interlan.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

Undefined - An undefined command status code is returned.

Notes:

None.

3.4.27 LOAD TRANSMIT DATA (50)

Command Description:

The Load Transmit Data command DMA transfers a UNIBUS buffer of data to the controller's Transmit FIFO buffer. After being transferred, the data is stored until the Load Transmit Data and Send (51) command is issued.

This command permits the host to assemble frames for transmission in the controller, avoiding the possible need to move blocks of data to an intermediate buffer region before being loaded into the controller.

Command Sequence:

1. Move the low 16-bits of the 18-bit buffer address into BAR<15:0>

2. Move the buffer byte size into BCR<15:0>

3. Concatenate the high 2-bits of the 18-bit buffer address with the command function code 50 and move it into CSR<15:8>

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the UNIBUS buffer was DMA'd to the controller and the data now resides in the controller's Transmit FIFO buffer.

05 - Buffer Size Exceeded; the sum of the amount of data currently in the Transmit FIFO buffer and the amount of data in the buffer supplied exeeds 1508 bytes.

17 - Non-Existent Memory; non-existent memory resides in the buffer pointed to by BAR. The command was aborted by the NI1010A.

Data Format:

The required data format for frame transmission is shown in Figure 3-12 under the Load Transmit Data and Send (51) command. Notes:

None.

3.4.28 LOAD TRANSMIT DATA AND SEND (51)

Command Description:

The Load Transmit Data and Send command transfers a buffer of data to the controller's Transmit FIFO buffer and transmits the assembled frame onto the network.

Command Sequence:

1. Move the low 16-bits of the 18-bit buffer address into BAR<15:0>

2. Move the buffer byte size into BCR<15:0>

3. Concatenate the high 2-bits of the 18-bit buffer address with the command function code 50 and move it into CSR<15:8>.

Possible Command Function Codes Returned in CSR<3:0>:

00 - Success; the UNIBUS buffer was DMA'd to the Transmit FIFO and the frame was successfully transmitted onto the network without any collisions.

01 - Success With Retries; the UNIBUS buffer was DMA'd to the Transmit FIFO and the frame was successfully transmitted onto the network with one or more retransmissions required because collisions were experienced.

03 - Inappropriate Command; the controller is in an offline state and not in a data loopback mode. No action was taken by the controller.

04 - Transmitter Failure; the NI1010A failed to transmit the packet onto the Ethernet because the NM10A's processor's "jabber control" timer expired. This indicates a serious network system problem that should be investigated. Potential sources of this symptom include: malfunction of the local transceiver by its continuously asserting carrier sense; a remote transmitter "babbling" endlessly; a faulty transceiver cable.

05 - Buffer Size Exceeded; the algebraic sum of the amount of data currently in the Transmit FIFO buffer and the amount of data in the UNIBUS buffer supplied exceeds 1508 bytes. The controller purged the contents of the Transmit FIFO.

06 - Frame Too Small; the total data supplied for frame transmission was less than 8 bytes (incomplete destination and type fields). The controller purged the contents of the Transmit FIFO.

10 - Excessive Collisions; the UNIBUS buffer was transferred to the Transmit FIFO but the frame could not be delivered to the Ethernet after 16 transmission attempts. The controller purged the contents of the Transmit FIFO.

17 - Non-Existent Memory; non-existent memory resides in the buffer pointed to by the BAR. The command was aborted by the NI1010A.

Data Format:

See Figures 3-12 and 3-12a.

Notes:

On transmission the controller appends the CRC to the end of the data field.

The minimum valid data field is 46 bytes in length. If fewer than 46 (decimal) bytes are supplied, the NI1010A controller zero-fills the remainder of the data field to create a minimum sized frame.

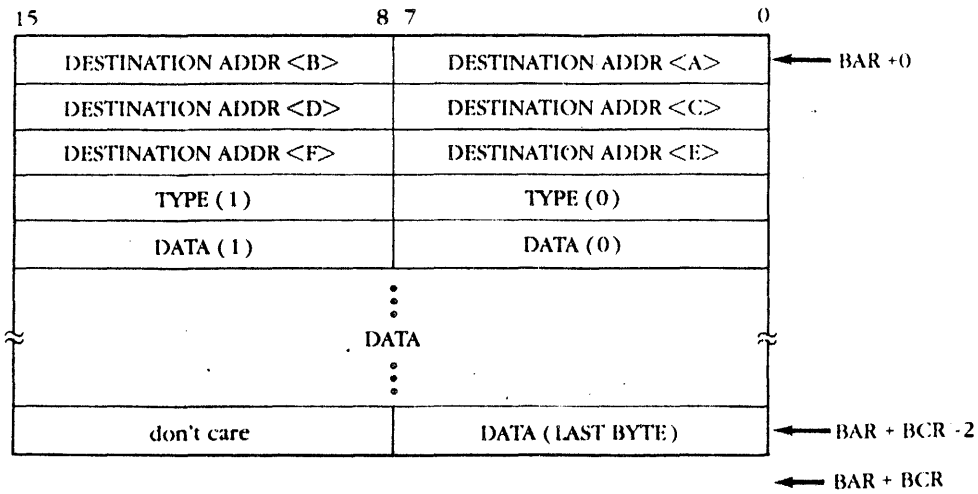


Figure 3-12. Transmit Data Format
(source address insertion)

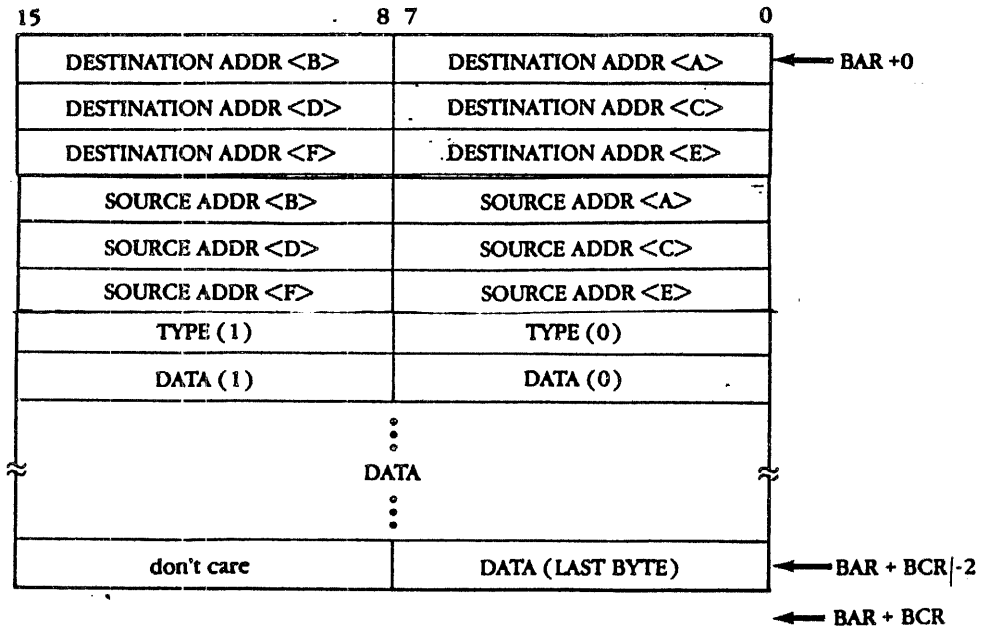


Figure 3-12a. Transmit Data Format
(no source address insertion)

3.4.29 LOAD GROUP ADDRESS(ES) (52)

Command Description:

The Load Group Addresses command loads multicast-group address values into the controller for multicast address recognition. Up to 63 multicast-group addresses may be loaded into the controller.

Prior to moving a received multicast packet to a receive buffer, the NM10A's processor hashes the destination address, using the result as a "first guess" into the group address recognition table, then performing a byte for byte comparison of the packet's destination address with one or more table entries.

Command Sequence:

1. Move the low 16-bits of the 18-bit buffer address into BAR<15:0>

2. Move the buffer byte size into BCR<15:0>

3. Concatenate the high 2-bits of the 18-bit buffer address with the command function code 52 and move it into CSR<15:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the UNIBUS buffer of multicast-group addresses was loaded into the controller's onboard multicast-group address table.

05 - Buffer Size Exceeded; the controller's multicast-group address table is now full. Not all (or any) of the supplied addresses were entered.

12 - Buffer Alignment Error; the UNIBUS buffer supplied is not a multiple of 6 bytes. No entries are made to the table.

17 - Non-Existent Memory; non-existent memory resides in the buffer pointed to by BAR. The command was aborted by the NI1010A.

Data Format:

See Figure 3-13 on the next page.

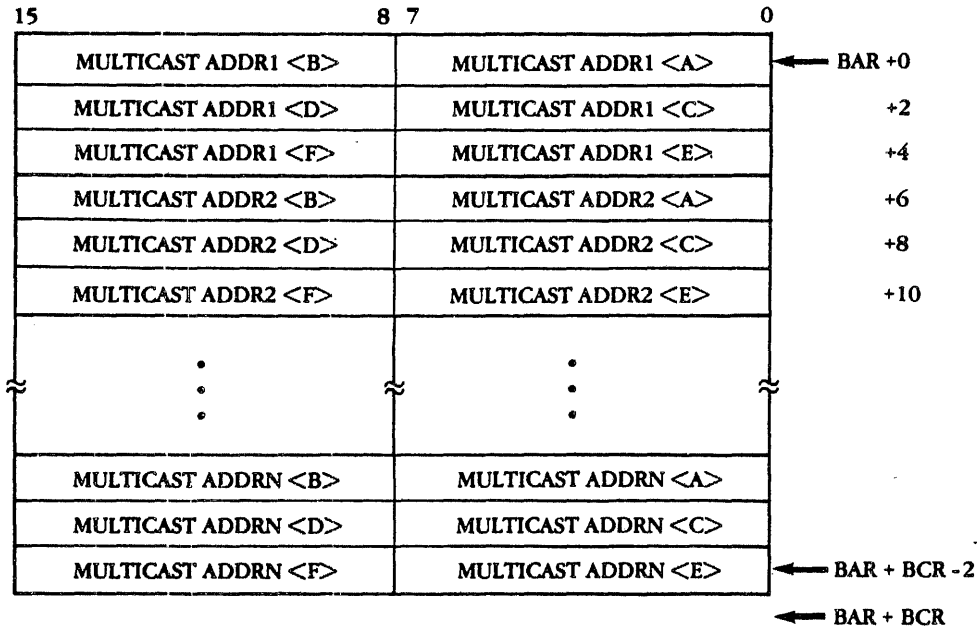


Figure 3-13. Multicast-Group Address Format

Notes:

No address mode checking is done by the controller on the address values supplied. If non-multicast-group addresses are entered they will not be used for multicast address recognition (they do, however, consume space in the table).

The controller performs multicast-group address recognition on the entire 48-bit field, not on any selected subfields.

3.4.30 DELETE GROUP ADDRESS(ES) (53)

Command Description:

The Delete Group Address command removes selected multicast-group addresses from the controller's multicast-group table.

Command Sequence:

1. Move the low 16-bits of the 18-bit buffer address into BAR<15:0>

2. Move the buffer byte size into BCR<15:0>

3. Concatenate the high 2-bits of the 18-bit buffer address with the command function code 53 and move it to CSR<15:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the UNIBUS buffer of multicast-group addresses are not in the controller's multicast-group address table.

05 - Buffer Size Exceeded; more than 63 multicast-group addresses were supplied. No action was taken by the controller.

12 - Buffer Alignment Error; the UNIBUS buffer supplied is not a multiple of 6 bytes. No table entries were deleleted.

17 - Non-Existent Memory; non-existent memory resides in the buffer pointed to by BAR. The command was aborted.

Data Format:

See Figure 3-13 under the Load Group Address(es) command (52).

Notes:

The NI1010A does not support a "Flush Group Addresses" command. It is the users responsibility to keep track of the current contents of the controller's multicast-group address table.

It is possible to delete the Broadcast Address from the table, thus disabling reception of Broadcast frames. The Broadcast Address may be replaced in the table using the Load Group Address(es) Command.

3.4.31 LOAD PHYSICAL ADDRESS (54)

Command Description:

This command loads a user-specified Ethernet physical address into the NI1010A. The factory-programmed Ethernet physical address will not be used in processing susequently received packets.

The physical address recognition hardware is disabled by this command. All Ethernet packets will consume space in the receiver FIFO memory until it has passed through an address filtering process invoked prior to output to a user receive buffer. Each physically addressed packet will have its destination address compared with the address specified.

Command Sequence:

1. Move the low 16 bits of the 18 bit UNIBUS memory address containing the first byte of the new Ethernet address into BAR<15:0>

2. Move a byte count of 6 into BCR<15:0>

3. Concatenate the upper 2 bits of the 18 bit UNIBUS buffer address with the command function code 54; move this result into CSR<15:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the data specified is now the physical address of the NI1010A.

12 - Buffer Alignment Error; the byte count specified was not equal to 6. The physical address of the NI1010A is not changed.

17 - Non-Existent Memory; read-only or non-existent memory in UNIBUS buffer pointed to by BAR.

Notes:

Use of this command disables the physical address recognition hardware forcin the NM10A's processor to filter all physically addressed packets prior to delivering them to a receive buffer.

The NI1010A's physical address recognition hardware may be enabled by issuing the command "Set Physical Address to Default" (see Section 3.4.15).

If source address insertion mode is set, the address specified by this comm will be inserted as the source address of packets subsequently transmitted.

Different physical addresses may be loaded at any time by this command. Load a new physical address while the NI1010A is online may give bizarre results (i.e., previously acceptable packets waiting in the receiver FIFO will be discarded prior to their delivery to a receive buffer).

3.4.32 RESERVED (55-57)

Command Description:

Command function codes 55 through 57 are reserved by Interlan for future use. Do not issue them.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

02 - Illegal Command; Interlan has reserved these command codes.

Notes:

None.

3.4.33 FLUSH RECEIVE BAR/BCR QUEUE

Command Description:

The Flush BAR/BCR Queue command invalidates the contents of the Receive BAR/BCR Queue.

Command Sequence:

1. Move command function code 60 into CSR<13:8>.

Possible Command Status Codes Returned in CSR<3:0>:

00 - Success; the contents of the Receive BAR/BCR Queue have been invalidated.

Notes:

None.

3.4.34 RESERVED (61-67)

Command Description:

Command function codes 61 through 67 are reserved for future use by Interlan.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

Undefined - An undefined command status code will be returned.

Notes:

None.

3.4.35 RESERVED (70-76)

Command Description:

Command function codes 70 through 76 are reserved for future use by Interlan.

Command Sequence:

Not applicable.

Possible Command Status Codes Returned in CSR<3:0>:

Undefined - An undefined command status code will be returned.

Notes:

CAUTION
These command function codes perform a DMA transfer of undefined data into UNIBUS memory.
****

3.4.36 RESET (77)

Command Description:

The Reset command initializes the NM10A. The following states are assumed by the module:

a) Offline; the module's transmitter and receiver are "disconnected" from the network. The board is not in any loopback mode. The Receiver FIFO is empty.

b) Normal Receive Mode; only frames with destination addresses that match the module's factory assigned physical address, or have the network broadcast address will be accepted.

c) Receive-On-Error Mode Turned Off; received frames with CRC and/or alignment errors are not accepted by the module.

d) All Multicast-Group Addresses Deleted; the contents of the multicast-group address table are cleared.

e) Source Address Insert Mode enabled; the module will insert the source address and perform size checking of all transmitted packets.

f) Receive BAR/BCR Queue Flushed; the contents of the Receive BAR/BCR are invalidated.

g) All Statistical Counters Reset to Zero.

Command Sequence:

1. Move command function code 77 into CSR<13:8>.

Diagnostic Status Codes Returned in S_REG:

00 - Success; the on-board diagnostics executed without detecting any faults on the NM10A module.

01 - Checksum Error; a checksum error was detected in the NM10A's processor memory.

02 - Memory Error; an error was detected in processor's scratchpad memory.

03 - Address Error; an error was detected while attempting to read the board's Ethernet address.

04 - Loopback Failure; the NM10A was unsuccessful at

transmitting a test frame while in internal loopback mode.

05 - Carrier Sense Failure; Internal loopback test could not be completed due to Ethernet cable failure.

NOTE: These are diagnostic status codes, NOT command status codes.

Notes:

The Reset command may be given to CSR<13:8> at any time. This command will be recognized and executed even while the controller is executing a previous command.

This command takes approximately one-half second to execute.

CHAPTER FOUR

FUNCTIONAL DESCRIPTION

Chapter 4 provides a functional description of the NI1010A UNIBUS Ethernet Communications Controller. Presented are descriptions of the controller's architecture, the Ethernet frame format, and the processes the controller goes through when transmitting, receiving, and performing its onboard self-test diagnostic.

4.1 NI1010A BOARD ARCHITECTURE

4.1.1 Physical

The NI1010A is a single hex-height board assembly that fits into one UNIBUS SPC slot. It is physically constructed of two separate board assemblies; a UNIBUS interface board (UIB) and the Interlan NM10A Ethernet Protocol Module. The NM10A module, measuring 10.0"x6.75"x0.375", is flush-mounted onto the UIB. The UIB and NM10A interconnect by a field of 2 by 30 posts on the UIB that enter through the bottom of the NM10A's P1 connector. A mylar sheet between the boards electrically isolates the boards from each other. Connection to the Ethernet is made via the NM10A's P2 16-pin flat cable interface.

4.1.2 Functional Architecture

Functionally, the UIB contains the logic necessary for transfering data between the NM10A module and the UNIBUS host system, and the NM10A module contains the NI1010A's Ethernet data communications logic. See Figure 4-1 for a block diagram of the NI1010A's UIB, and Figure 4-2 for a functional diagram of the NI1010A's NM10A module.

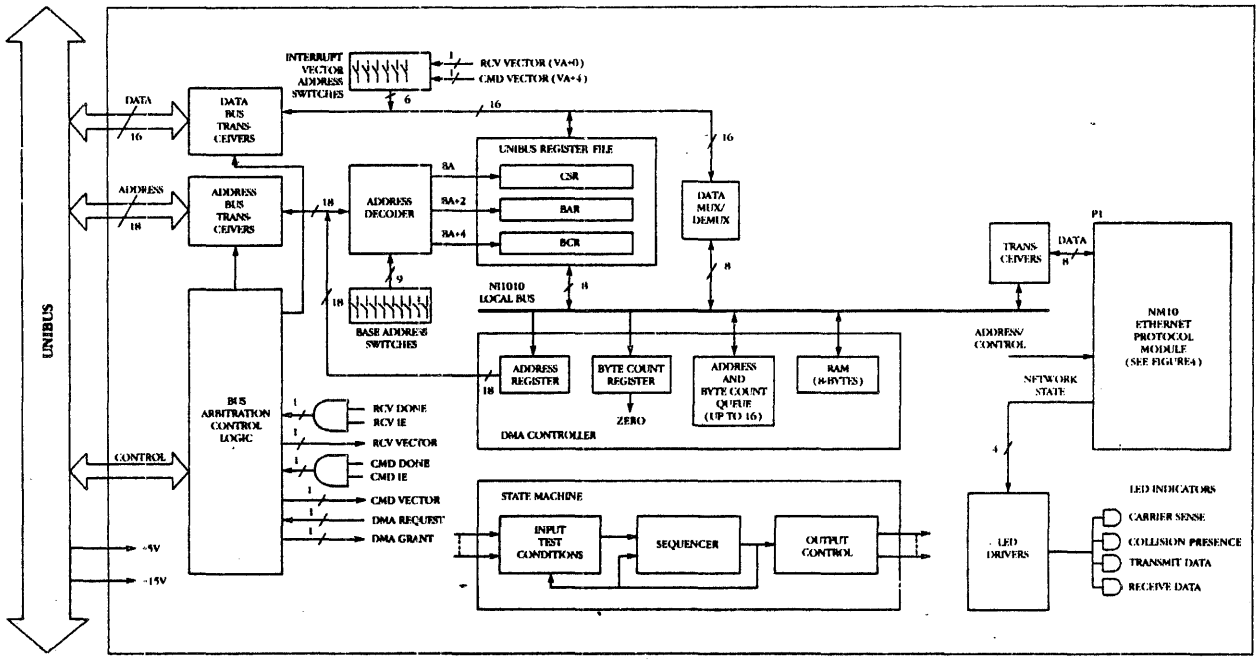


Figure 4-1. NI1010A UIB Block Diagram

4.2 NI1010A UNIBUS INTERFACE BOARD (UIB)

The NI1010A UIB (shown in figure 4-1 above) contains the following:

- o UNIBUS "SLAVE" interface logic
- o UNIBUS Register File
- o DMA controller
- o State Machine
- o Network LED Indicators

4.2.1 UNIBUS "SLAVE" Interface

This logic handles all UNIBUS data transactions which are not NI1010A initiated DMA or interrupt sequences. These include processor initiated data-in transactions (DATI or DATIP), and data-out transactions (DATO or DATOB). Each time a UNIBUS transaction occurs, the "SLAVE" logic performs the following actions:

- * Determines that the NI1010A is the device being selected by comparing the UNIBUS I/O page address with the NI1010A base address as selected by the base address switches. If the NI1010A is not being selected, the UNIBUS transactio is ignored. (For a detailed description of the use of the base address switches, refer to chapter 2.)

- * If selected, the lower UNIBUS address bits determine which register in the UNIBUS register file is accessed.

- * For data-in transactions, the data from the appropriate register is placed on the UNIBUS.

- * For data-out transactions, the data from the UNIBUS is written into the appropriate register.

- * The transaction is completed, and if the transaction was a write into the command register, an action flag is set to indicate to the state machine that a command has been issued.

4.2.2 UNIBUS Register File

The UNIBUS register file contains the Control and Status Register (CSR), Buffer Address Register (BAR), and Byte Count Register (BCR). These are program accessible (as described in chapter 3) through the I/O page.

4.2.3 DMA Controller

The DMA controller logic contains the registers necessary to support direct memory access (DMA) transfers of data to or from the host UNIBUS system memory. These include the 18-bit DMA Address Register (DAR), the 18-bit DMA byte Count Register (DCR), the Receive Buffer Queue, and the 8-byte data buffer.

The DAR is used (under control of the state machine) to generate the 18-bit address on the UNIBUS during DMA transactions. It is loaded by the state machine with an address from either the Receive BAR/BCR queue or the BAR register, depending upon the type of transfer. For DMA transfers that transfer a block of receive data, the starting DMA address is loaded into the DAR from the receive buffer queue. For all other types of DMA transfers (transmit data or statistics), the starting DMA address in the DAR is loaded with 16 bits from the BAR concatenated with the upper 2 address bits from the CSR.

The DCR is used (under control of the state machine) to count the number of bytes transferred during the DMA. It is loaded by the state machine with a byte count from either the receive buffer queue or the BCR, depending upon the type of transfer. For DMA transfers that transfer a block of receive data, the starting DMA byte count is loaded into the DCR from the receive buffer queue. For all other types of DMA transfers (transmit data or statistics), the starting DMA byte count in the DCR is loaded from the BCR. The byte count in the DCR is decremented for each byte transferred during the DMA. If it reaches zero (indicating that the buffer has been filled) then the DMA transfer is stopped. It should be noted that the DMA data transfers are performed in bursts of 4 16-bit words (8 bytes) until the last burst, which can be 1, 2, 3, or 4 words long depending on the number of bytes remaining.

The receive buffer queue can hold up to 16 pairs of addresses and byte counts to be used for the DMA of receive data. These receive buffers are loaded by the program using the "Supply Receive Buffer" command which is described in chapter 3.

The 8-byte data buffer is used to match the UNIBUS DMA transfer rate with the rate at which the NM10A can give or take data. During receive DMA transfers, 8 bytes are loaded from the NM10A into the data buffer. This data is then transferred into the host UNIBUS system memory. During transmit DMA transfers, 8 bytes are loaded from the host UNIBUS system memory into the data buffer. This data is then transferred to the NM10A.

4.2.4 State Machine

The NI1010A DMA and command functions are controlled by a PROM based Finite State Machine (FSM). The FSM generates all UNIBUS "MASTER" signals for DMA and interrupt operation. Additionally, it manages the data transfers between the various registers, buffers, UNIBUS data bus (during DMA), and the NM10A. The FSM also transfers commands, when appropriate, to the NM10A.

4.2.5 Network LED Indicators

The NI1010A has 4 Network LED Indicators which provides information about the state of the Ethernet Network. These LED's are located on the top of the board and are labelled CS (carrier sense), CP (collision presence), XMT (transmitting), and RCV (receiving). If they are on,

then the appropriate condition is indicated. The on times of the LED's do not directly reflect the network condition in real time, but rather they are lengthened to a minimum time that can be detected by the human eye.

4.2.5.1 Carrier Sense (CS) LED Indicator

The Carrier Sense LED is ON whenever any network traffic is detected on the Ethernet. This indicator is useful to affirm that the NI1010A is properly connected to the network (ie, the transceiver is attached to the coaxial cable, and all cabling is attached between the NI1010A and its transceiver unit).

4.2.5.2 Collision Presence (CP) LED Indicator

The Collision Presence LED is ON whenever the transceiver unit detects that a collision has occurred on the Ethernet. The collision may or may not be one involving the NI1010A controller itself.

4.2.5.3 Transmitting (XMT) LED Indicator

The Transmitting (XMT) LED is ON whenever the NI1010A controller is transmitting a bit-stream to the Ethernet.

4.2.5.4 Receiving (RCV) LED Indicator

The Receiving (RCV) LED is ON whenever the NI1010A is receiving a frame on the network that matches the controller's physical address, or a frame with a destination address field that is a multicast address. This LED will also light during reception of loopback packets.

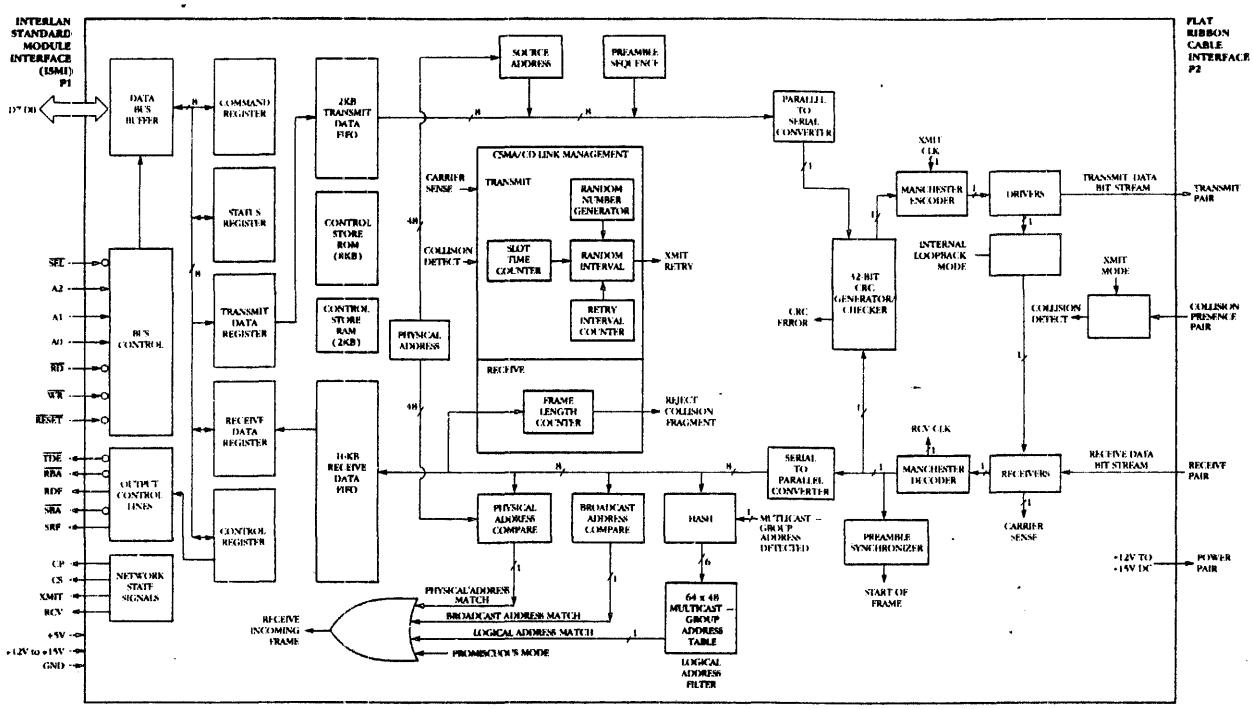


Figure 4-2. Functional Diagram of the NM10A

4.3 NM10A ETHERNET PROTOCOL MODULE

The NM10A Ethernet Protocol Module (shown in Figure 4-2 above) provides the data communication control logic required for interfacing the NI1010A's UIB to the Ethernet local area network. The module complies with the Xerox/Intel/DEC Ethernet V1.0 Specification. The module has a microprocessor-bus compatible interface at the 60 pin P1 connector interface. Data is transfered through an 8-bit bidirectional data bus, to five internal 8-bit registers selected by three address lines. The internal NM10A registers are called:

- Command Register (C_REG)
- Status Register (S_REG)
- Transmit Data Register (T_DATA)
- Receive Data Register (R_DATA)
- Control Register (H_REG).

Two buffers are included to support high station performance:

* a 13.5K Byte Receive FIFO buffer

* a 1.5K Byte Transmit FIFO buffer

The 1.5K Byte Transmit FIFO buffer permits the host UNIBUS system to perform a one-time transfer of a frame to the NI1010A for transmission on the network. Reloading of a transmit data frame by the user is not required when transmission is rescheduled after a network access collision, since the NI1010A performs collision retries automatically.

The 13.5K Byte Receive FIFO buffer stores receives frames to buffer the UNIBUS from the unpredictable arrival times of network traffic. This front-end buffering serves to reduce time-critical service requirements of the host UNIBUS system.

4.4 ETHERNET FRAME FORMAT

The Ethernet frame format shown in Figure 4-3 illustrates the five fields of a frame:

Destination Address,
Source Address,
Type,
Data, and
Frame Check Sequence.

All fields are of fixed size except the Data field, which can range in size from a minimum of 46 bytes to a maiximum of 1500 bytes. Given the minimum and maximum sizes of the data field and the 18 bytes of the other four fields, the smallest valid frame contains 64 bytes and largest frame contains 1518 bytes. For synchronization purposes, the frame is preceeded by a 64-bit preamble sequence and terminated with a minimum interframe spacing period of 9.6 microseconds.

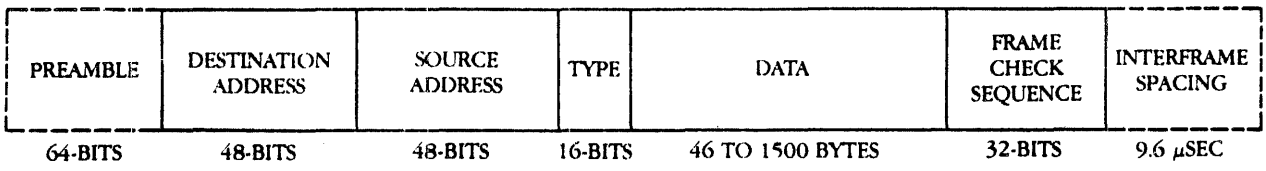


Figure 4-3, Ethernet Frame Format

4.4.1 Destination Address Field

The destination address field specifies the station or stations to which the frame is intended. The destination address provided by the user may be either:

* the physical address of a particular station
* a multicast-group address associated with one or more stations, or
* the broadcast address for all stations at once.

By convention, addressing is defined as consisting of 6 bytes labled A,B,C,D,E, and F. The first bit of the destination address distinguishes a physical from a multicast address:

0 = physical address

1 = multicast address

The least significant bit of each of these bytes is transmitted first. Note that an odd value in byte A indicates a multicast address, an even value indicates a physical address. The broadcast address is composed of all ones, ie.,A=FF, B=FF, C=FF, D=FF, E=FF, F=FF(hex).

The NM10A requires that a destination address be specified for all frames to be transmitted over the Ethernet.

4.4.2 Source Address Field

The source address field specifies the physical address of the transmitting station. To eliminate the possibility of an addressing ambiguity on a network, a unique 48-bit physical address value has been assigned by Interlan to each NI1010A at the time of manufacture. The physical address is resident in a programmable read-only memory device mounted on the NM10A module.

The user may change the physcial address by means of the "Load Physical Address" command.

On transmission, the NM10A will insert its physical address into the source address field of the frame if source address insertion mode is set, padding the frame with null characters if necessary.

If source address insertion is not set, the user must supply the source address in the appropriate location in host memory.

4.4.3 Type Field

The type field is specified by the user to designate the higher level protocol by which the data field is to be interpreted.

The NM10A requires that the type field be specified in all frames to be transmitted over the Ethernet.

4.4.4 Data Field

The data field can contain a variable number of data bytes ranging from 46 minimum to 1500 maximum. The NI1010A accepts less than 46 bytes of data from the user by automatically inserting null characters into the data field to complete the 46-byte minimum frame size.

The NI1010A will refuse to transmit a frame that would exceed 1518 bytes in length, and will return a command status code of 05 to indicate the buffer supplied by the user was too large.

4.4.5 Frame Check Sequence Field

The frame check sequence (FCS) field contains a 32-bit cyclic redundancy check (CRC) value. The CRC value is computed over the destination address, source address, type, and data fields.

The NI1010A contains shared CRC logic circuits. During transmission, the NI1010A calculates the CRC value on the transmit bit-stream, and inserts the value into the FCS field by appending it to the end of the frame's data field. During reception, the NI1010A calculates a CRC value on the incoming bit-stream and compares it against the CRC value found in the FCS field. Appendix C of the Ethernet V1.0 Specification

provides a detailed description of CRC implementation.

4.5 NI1010A TRANSMIT PROCESS

The NI1010A Transmit Process consists of transferring data packets from host UNIBUS memory, forming them into Ethernet frames, and performing the functions required to successfully deliver the frame onto the network. The NI1010A performs the specified Ethernet Data Link and Physical Layer functions required to form and transmit the frame at 10 Mbits per second.

4.5.1 Transmit Data DMA Transfer

Transfering a data transmit packet from the host system to the NI1010A's NM10A module is done as follows:

1. The host system writes a UNIBUS memory buffer address and buffer byte count size to the NI1010A's BAR and BCR registers respectively;

2. The host system writes a command function code of 50 or 51 (with the two BAR address extension bits) into the CSR register;

3. The NI1010A successively moves 8-bytes (4 words) of data under DMA control from host UNIBUS memory to the NM10A module through the 8-byte DMA Data Buffer located on the NI1010A's UIB.

Figure 4-4 shows two working registers (DAR and DCR) and one queue (DMA Data Buffer). Note that the 18-bit DAR contains the concatenation of the two most significant bits <15:14> of the CSR and the 16 bits <15:0> of the BAR. The DMA controller requests control of the UNIBUS when the CSR register is loaded with command code 50 or 51. (NOTE: although only transmit DMA operations are being discussed, the DMA controller operates in an identical manner for all Type II and Type III commands (see Chapter 3)).

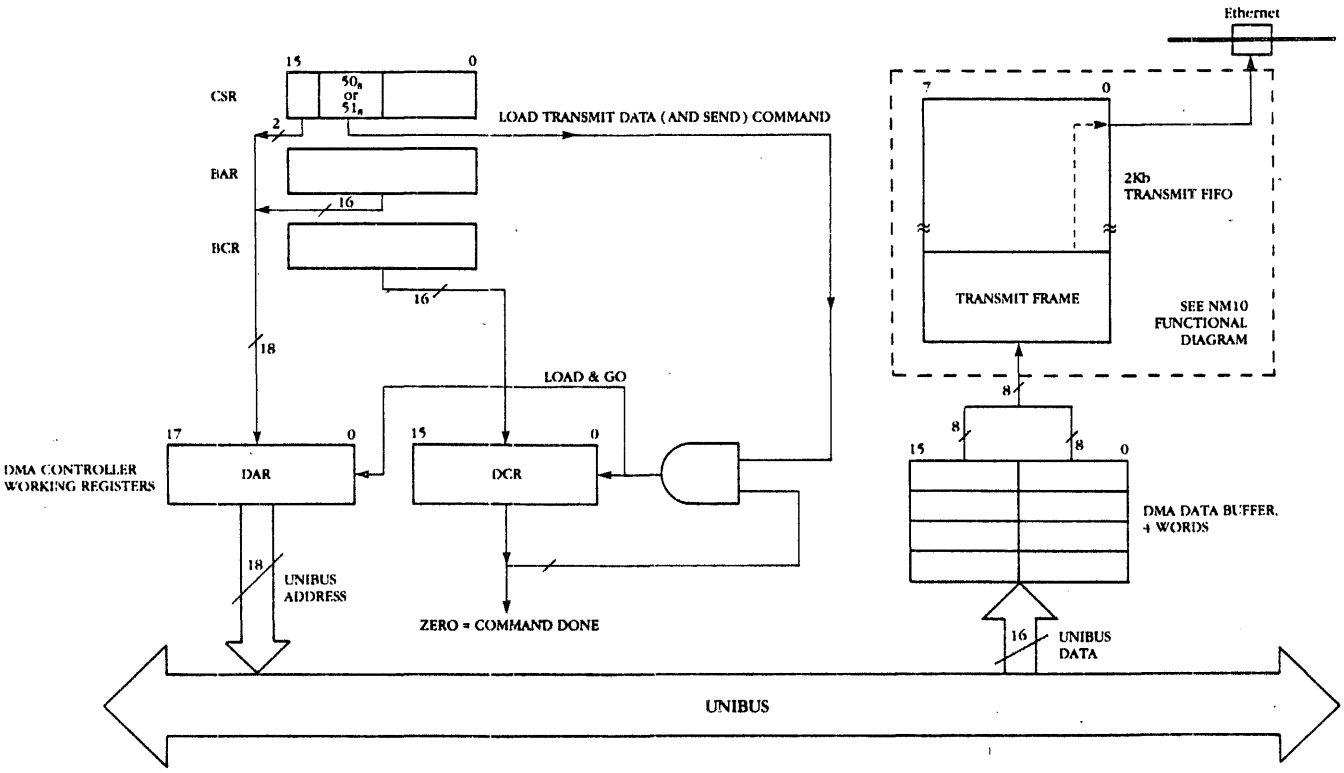


Figure 4-4. Functional Diagram of NI1010A Transmit DMA Operation

Once UNIBUS access is granted, it moves a 4-word block to the DMA Data Buffer. This is accomplished by incrementing the DAR to point to each successive data word in the host buffer. The DCR is decremented when words are transferred into the DMA Data Buffer. The DMA controller re-establishes access to the UNIBUS after each 4-word data burst. Data is asynchronously moved a byte at a time from the DMA Data Buffer into the NM10A's 2KByte Transmit FIFO. This process continues until all data in the UNIBUS memory buffer has been transferred. The Command Done (CMD DONE) status bit is set in the CSR register at the successful completion of the transmit DMA operation and the Command Done Interrupt issued if the Command Done Interrupt Enable (CMD IE) bit in the CSR is set.

4.5.2 Frame Assembly

The NM10A performs the following frame assembly processes:

* The NM10A inserts the module's source address into the frame found in the Transmit FIFO;

* The NM10A generates the frame's FCS field

as the bit-stream is transmitted onto the network.

4.5.3 Channel Access

The NI1010A performs carrier detection and deference, collision handling, and interframe spacing processes.

* Carrier Detection and Deference: the NI1010A monitors the physical channel and defers its transmission should the channel be busy carrying other traffic;

* Collision Detection: once the NI1010A has finished deferring to traffic on the network, it proceeds with its own transmission. In the event that another station simultaneously begins a transmission, a "collision" occurs. The NI1010A detects this occurence and goes into a "jam" sequence in which it transmits between 32 and 48 zeros before terminating its transmission. The jam guarantees that all staions on the network detect that a collision has occurred on the network (ie, runt packet).

* Backoff and Retransmission: when a transmission is terminated due to a collision, the NI1010A attempts its transmission again after delaying a short period of time. The resheduling of the transmission is determined by the Ethernet process called "truncated binary exponential backoff". The NI1010A reports an error should it be unable to deliver its frame onto the network after 16 transmission attempts.

4.5.4 Physical Channel Functions During Transmission

During transmission of the bit-stream, the NI1010A:

* provides proper channel access by noting the absence or prescence of carrier from the transceiver;

* generates the 64-bit preamble sequence;

* maintains channel access by noting the state of the collision presence signal from the transceiver;

* creates a self-synchronizing bit-stream through Manchester encoding of transmitted data.

4.6 NI1010A RECEIVE PROCESS

The NI1010A Receive Process consists of decoding relevant transmissions from the network and transfering them as formatted data packets to host UNIBUS memory buffers. The NI1010A performs the specified Ethernet physical layer and data link functions required to permit a frame to be received from the network at 10 Mbits per second. With receive operations having the highest priority on the NM10A, the controller is always listening for incoming network traffic.

4.6.1 Physical Channel Functions During Reception

Within the Ethernet physical channel layer, the NI1010A performs the following functions during the reception of a frame:

- o synchronizing to, and removal of, the preamble sequence,
- o Manchester decoding the incoming bit-stream into a receive data stream and a clock stream.

After the preamble synchronization occurs, the NI1010A performs address recognition on the destination address field to determine whether or not the frame is intended for the controller. The NI1010A controller will only accept a frame with a destination address that either:

- * matches the physical address of the NI1010A board itself,
- * contains the broadcast address, or
- * matches one of the 63 multicast-group addresses which the user may assign to the board.

Whenever a frame with a multicast-group destination address is received, the NI1010A converts the frame's 48-bit destination address into a 6-bit table entry pointer through the application of a many-to-few mapping called "hashing". It uses the resultant pointer to look into a table of valid multicast-group addresses to see if the received address is a 48-bit match with one that the controller has been programmed to accept. Accepted frames are then placed into the next available UNIBUS buffer under DMA control. Frames which do not pass through the multicast filter are flushed from the Receiver FIFO.

The NI1010A may also be operated in a "promiscuous" receive mode which permits it to receive all undamaged frames passing on the network.

4.6.1 Collision Fragment Filtering

Since collisions are a normal occurrence on the Ethernet, the NI1010A's CSMA/CD receive link management process filters out collision fragments by rejecting all received frames that are less than 64 bytes in length.

4.6.2 Receive FIFO Buffer Management

The NM10A portion of the NI1010A manages its 13.5k byte receive memory as a ring buffer: received frames are stored in first-in/first-out fashion, wrapping the input pointer at the 13.5k byte boundary. The NM10A firmware manages the input pointer, the output pointer and the total byte count of received frames. When a frame is received from the Ethernet, its length and status bits are latched by the receiver hardware. The NM10A's microprocessor reads the latched length and status bits, adds the byte count to the total currently in the FIFO, and stores the combined length and status as an entry in received packet status table. If insufficient space exists in the FIFO for reception of a maximum-sized frame, the receiver hardware is disabled. Each frame is allocated exactly its size in bytes. The maximum number of packets which may reside in the receive memory is (13311 - 2048) div 64 = 175.

Free space is made available to incoming frames by two mechanisms:

> an accepted frame is formatted with four bytes of header information (status, null, length low byte, length high byte) and transferred to the next available UNIBUS memory buffer;
>
> or, a rejected frame is flushed from the receiver memory.

In either case, the output pointer and the total byte count is adjusted by the frame's length, thus freeing the number of bytes occupied by the frame.

4.6.3 Receive Data DMA Transfers

After the frame is received from the network, it is formatted into a receive packet by inserting the Frame Status and Frame Length fields. The packet is then moved under DMA control from the 13.5K Byte Receive FIFO to a UNIBUS memory buffer through the DMA Data Buffer.

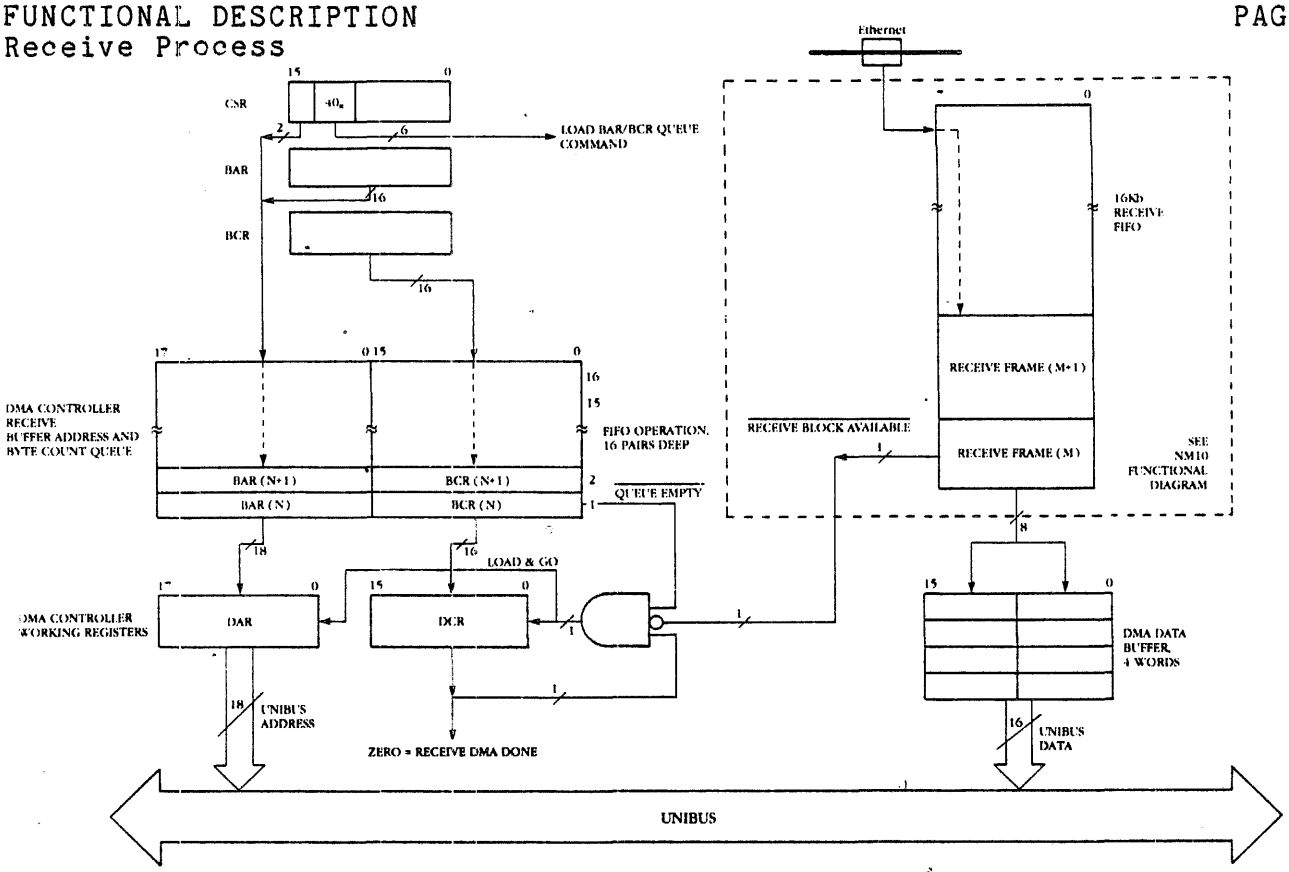


Figure 4-5. Functional Diagram of NI1010A Receive DMA Operation

Figure 4-5 shows the DMA controller's two working registers, DAR and DCR, the DMA controller's Receive BAR/BCR Queue, and DMA Data Buffer.

The Receive BAR/BCR Queue is loaded by the Supply Receive Buffer command (40). Up to 16 buffer descriptor pairs may be pre-loaded into this queue by successive issuance of this command.

The DMA controller begins operation when the Receive Block Available line from the NM10A goes active, indicating a data packet is available in the 13.5K Byte Receive FIFO. If the DMA controller's address queue contains a buffer descriptor, the controller requests access to the UNIBUS. Once access is granted, the DMA controller moves a 4-word block to the host memory location pointed to by the first address in the address queue. The DAR address is incremented and the DCR count is decremented as data words are transferred to the host memory.

The DMA controller re-establishes access to the UNIBUS after every 4 word data transfer. This process continues until the host's memory buffer has been filled, or until the complete data packet in the receive buffer has been transferred. The RCV DONE status bit is set in the CSR register at the successful completion of the receive DMA operation and the receive interrupt is issued if the RCV IE bit in the CSR is set.

4.6.4 Buffer Chaining

When the received frame is larger than the host UNIBUS memory buffer supplied and the supplied buffer is a multiple of 8 bytes, then the

NI1010A's DMA controller completes the DMA transfer by re-initializing from the Receive BAC/BCR Queue and continuing its transfer until the entire frame has been placed into UNIBUS memory. If the received frame is larger than the host UNIBUS memory buffer, and the supplied buffer is not a multiple of 8 bytes, then the receive frame is truncated to the allocated buffer size. After each UNIBUS memory buffer has been loaded, the CSR's RCV DONE bit is set.

It is the user's responsibility to keep track of the number of BAR/BCR buffer descriptors consumed by the DMA controller. The Frame Length word, at the second word of the received packet, should be used to facilitate this process.

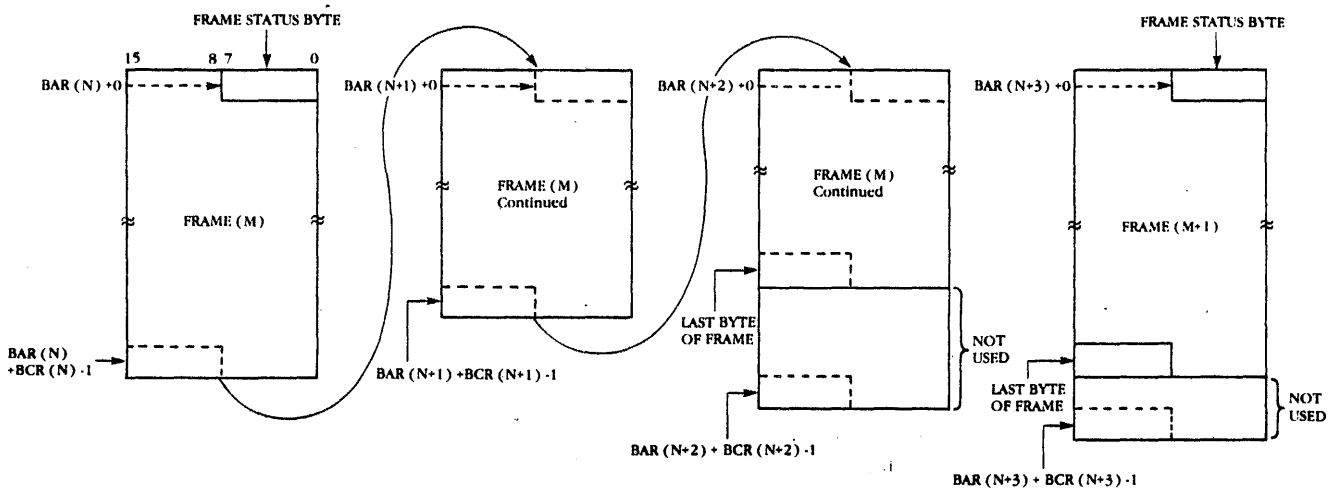


Figure 4-6. Receive DMA Buffer Chaining

4.7 DIAGNOSTIC PROCESS

For diagnostics, the controller has i) three levels of data loopback operation, and ii) power-up self tests with a Pass/Fail LED indicator.

4.7.1 Module Interface Data Loopback Operation

Isolation of a problem relative to the NI1010A controller, the transceiver cable, the transceiver, or the Ethernet coaxial cable requires the ability to test at different levels. To verify the correct operation of the UNIBUS DMA hardware, and the basic functionality of the NM10A, you may use the "Set Module Interface Loopback" command in conjunction with subsequent "Load Transmit Data and Send" and "Supply Receive Buffer" commands. Module interface loopback causes all transmitted frames to return at the module's interface as received frames without being passed through the CRC generator, manchester encoding, or transceiver level-shifting hardware. Each transmitted frame will be placed into a UNIBUS memory buffer exactly as if it were received from the Ethernet.

4.7.2 Internal Data Loopback Operation

The next level of loopback available for fault isolation is known as internal loopback. Once the NI1010A controller has been placed into this state transmitted frames will be passed through the NM10A's transmit buffer and receiver memory hardware. If the frame passes the address recognition logic (or "promiscuous mode" is set), the frame will be transferred into a UNIBUS receive buffer. A program may then compare the contents of the receive buffer with the contents of the "transmitted" frame. These frames will be returned exactly as if received from the Ethernet except that the appended CRC bytes are not valid. While in this state, the NI1010A controller processes all commands normally, one exception being that transmitted frames will not be sent to the Ethernet.

4.7.3 External Data Loopback Operation

The third loopback operation involves the transmission of an Ethernet frame over the network to the transmitting station. That is, the NI1010A is capable of sending to itself, making use of the controller, the transceiver cable and connectors, the transceiver itself, and the Ethernet coaxial cable. To do this, the controller must be online, and a test frame transmitted onto the network by use of the "Perform Network Loopback Test" command. All of the Ethernet rules are observed including backoff on collision detect and aborting a transmission after sixteen attempts. Issuing this command causes the NI1010A to transmit two minimum length Ethernet packets with source address and destination address of 2 7 1 0 0 0. The first of these packets is sent with an incorrect CRC byte appended, causing a CRC error at all receiving stations. The second packet is sent with valid CRC bytes appended. The

NI1010A receives its own transmission, checks each packet for valid or invalid CRC as appropriate, and checks that the last byte received matches the last byte transmitted. (See "Perform Network Loopback Test".)

Using the combination of loopback modes will assist you in isolation of a network problem to the level of a field replaceable component.

4.7.4 NM10A Power-up Diagnostic Self-Test

The NM10A contains on-board (ROM resident) diagnostics and a pass/fail LED indicator. These diagnostics are invoked by issuing the Run On-board Diagnostics (12) command.

See Section 5.2 for details on these power-up diagnostic tests.

CHAPTER FIVE

MAINTENANCE

5.0 GENERAL

The NI1010A does not require special maintenance procedures or any calibration. The controller is designed to be installed in standard UNIBUS backplanes and connected to standard Ethernet transceivers and cables, and go on line. If a problem does develop, Interlan supplies adequate Test and Verification routines to determine proper operation. These diagnostics include:

1) NM10A On-board diagnostics; and

2) DS-NI1010 Standalone PDP-11 Diagnostics for the NI1010A controller

5.1 VERIFYING PROPER OPERATION

Either upon installation or when in doubt about NI1010A board operation the following sequence should be followed:

1) Verify proper NM10A operation by the state of the Pass/Fail LED indicator (see section 5.2); and

2) Verify proper NI1010A operation by the successful completion of DS-NI1010 routines (see section 5.3).

When both 1) and 2) above are successful the user can assume that the NI1010A controller board is operating properly and that if station communication problems exist, a fault resides either on transmission cable or in another network station.

5.2 NM10A ON-BOARD DIAGNOSTICS

The NM10A contains on-board (ROM resident) diagnostics and a Pass/Fail LED indicator. These diagnostics by issuing the Run On-board Diagnostics (12) command.

5.2.1 Run On-Board Diagnostics (12) Command Operation

On-board diagnostics perform a state-of-health check of the NM10A. This command returns status if the hardware is sufficiently healthy to allow it. If the module fails self-test, the LED will be left in the on state. If the module passes self-test, then the LED will be left off. An implicit Go Offline command is performed when this command is issued.

The sequence of events that result by issuing this command are:

1. Pass/Fail LED turned ON

2. 1. Perform a checksum of the program memory device (ROM), matching the calculated value with the value stored in the last two bytes of the device. If failure detected, set status value to 01, go to step 7; otherwise, continue to step 3.

3. Perform test of the module's transmit and recieve memory by writing then reading four predetermined data patterns. If failure detected, set status value to 02, go to step 7; otherwise, continue to step 4. (Caution: All data stored in the transmit and receive memory will be lost.)

4. Perform test of the module's Ethernet address hardware by reading the receiver's assigned address and checking that values are valid. If failure detected, set status value to 03, go to step 7; otherwise, continue to step 5.

5. Perform test of the module's ability to send and receive a predetermined Ethernet packet using the internal loopback data path to execute a Perform Network Loopback Test command. If failure detected, set status value to 04, go to step 7; otherwise, set status value to 00, continue to step 6.

6. The diagnostics were successfully completed. Reset the board to initialize all variables and data stuctures.

7. If status value is success (00), turn OFF Pass/Fail LED.

8. Report status value to user thru S_REG.

   NOTE:

   Failure of the on-board diagnostics may leave the board in an unknown state. Always reset the board before attempting further use after failure.

5.2.2 Power-up or Reset Operation

Upon application of power, or assertion of the reset signal, or after a Reset command (77), the module will:

1. Perform a checksum of the program memory device (ROM), matching the calculated value with the value stored in the last two bytes of the device. If failure detected, set status value to 01, go to step 6; otherwise, continue to step 2.

2. Perform a test of the microprocessor's variable store (RAM), by writing then reading four predetermined data patterns (all 0's, and all 1's). If failure detected, set status value to 02, go to step 6; otherwise, continue to step 3.

3. Perform test of the module's Ethernet address hardware by reading the receiver's assigned address and checking that values are valid. If failure detected, set status value to 03, go to step 6; otherwise, continue to step 4.

4. Initialize all of the microprocessor's variables and data structures.

5. Perform test of the module's ability to send and receive a predetermined Ethernet packet using the internal loopback data path to execute a Perform Network Loopback Test command. If failure detected, set status value to 04, go to step 6; otherwise, set status value to 00, continue to step 6.

6. If status value is success (00), turn OFF Pass/Fail LED.

7. Report status value to user thru S_REG.

NOTE: Detection of any failure condition by a diagnostic test will cause the Pass/Fail LED to remain ON, giving a visual indication to the user as to the module's "state of health".

5.3 DS-NI1010 PDP-11 STANDALONE DIAGNOSTICS

The DS-NI1010 contains tests for both verification and error reporting of the NI1010A. The software runs on either an LSI-11 or PDP-11 in a standalone (i.e., under no operating system) environment.

The VAX/VMS device driver (NS2030-RX01 or NS2030-TU58) contains user mode diagnostics which may be linked and executed under the VMS operating system. This version of the diagnostics requires the NS2030 VMS device driver, but otherwise performs the same tests as the PDP-11 standalone diagnostics. See the documentation of the VMS device driver (UM-NS2030) for further details.

5.3.1 Operating the DS-NI1010 Standalone PDP-11 Diagnostic

Configuration requirements:

The DS-NI1010 standalone diagnostic runs on either a PDP-11 or LSI-11 system and is available on single (DS-NI1010-RX01) or double (DS-NI1010-RX02) density floppy diskettes (DEC RX01 and DEC RX02 compatible), or on cartridge tape (DS-NI1010-TU58).

Because each diagnostic diskette contains a device-dependent bootstrap, DS-NI1010-RX01 can not be booted on an RX02 and DS-NI1010-RX02 can not be booted on an RX01.

MAKE SURE YOU ARE USING THE CORRECT DISKETTE BEFORE CONTINUING.

DS-NI1010 requires an ASCII console terminal and console interface (DL11 type) configured to the standard console CSR and Vector addresses (777560 and 60).

The floppy diskette interface (RX01 or RX02 or equivalent) must be configured to the standard floppy CSR address (777170).

The cartridge tape requires a serial line interface at the CSR for the second DL11 (776500).

The PDP-11/LSI-11 system must contain a minimum of 32K words of main memory.

Operating Instructions:

1. There are several methods available to bootstrap a device on your system. The method you use depends on the type of processor and hardware bootstrap on your system. Refer to the appropriate Processor Handbook for the bootstrapping procedure for your system.

   Place the diskette into either drive and bootstrap the drive. The console terminal should print:

"Interlan NI1010/NI2010 Ethernet Controller
Diagnostic: Rev "

If your system is running but the above line was not displayed, there may be something wrong with the console terminal or interface. Check it out and reboot. If your system has halted, use the halt address with Table 1 of the diagnostics' Release Notes to determine the cause of the halt and the action to take.

2. The console will print:

"Please enter the CSR address if not 164000 (764000):"

If the CSR base address configured on your NI1010 board differs from 164000, type the CSR base address as configured. Press the RETURN.

3. The console should print:

"Using NI1010 (Unibus) diagnostics." (if PDP-11 system)
or
"Using NI2010 (Q-bus) diagnostics." (if LSI-11 system)

The DS-NI1010 and DS-NI2010 programs are identical; the bus type is determined when the diagnostic is run.

If instead of the above lines,

"Address xxxxx does not respond."

was printed, the NI1010A (or any other device) is not responding to the CSR address. Make sure the NI1010A is inserted in the backplane securely and that the CSR base address (BA) switches on the module are correct, then go back to step 2.

If the entered address is not for valid I/O controllers, the console will print:

"Please enter an octal address between 160000 and 177760."

Check the CSR address and enter it again.

4. The console will print:

"Please enter the interrupt vector address if not 340:"

If the interrupt vector address as configured on your

NI1010A differs from 340, type the interrupt vector address as configured. Press RETURN.

If the entered address is not a valid vector address, the console will print:

"Please enter an octal address less than or equal to 770."

Check the vector address (VA) before entering it again.

5. The console will print:

"Normal board priority is BR5" (if PDP-11 system)
"Enter 'Y' to change to BR4, else <CR>:"
-or-
"Normal board priority is BR4" (if LSI-11 system)
"Enter 'Y' to change to BR5, else <CR>:"

If you have changed the controller's interrupt priority, type YES, and press RETURN; otherwise, press RETURN.

Changing the interrupt priority level is discussed in Chapter 2, Installation, Section 2.3.4 Interrupt Bus Request Priority Selection.

6. The console will print:

"Enter test name, ? to list or RETURN for all tests:"

You have three choices:

a. You can start an automatic test series by pressing RETURN. The console will print:

"Repeat forever?"

If you would like the test series to repeat indefinitely, type YES. Press RETURN. The test series will run until a test fails or you stop it by pressing the CNTRL key then the letter C together.

b. You can obtain a list of individual diagnostic test names by pressing the question mark (?) key and pressing RETURN. The function of these tests is explained in section 5.3.2.

c. You can invoke an individual diagnostic test by typing the name of the diagnostic test and pressing the RETURN key. The function of each diagnostic test is explained in section 5.3.2. The console will print:

"Repeat forever?"

If you would like the test to repeat indefinitely, type YES. Press RETURN. The test will run until it fails or you stop it by pressing the CNTRL key then the letter C together.

If any diagnostic test fails, the console will print a line resembling (but not the same as):

"*** Test CHKCSR failed: 0 200 0 4 ***"

Use the test name (CHKCSR in the above example) and the test description in section 5.3.2 and the Diagnostic Error Messages list to help isolate the cause of the failure.

DS-NI1010 Diagnostic Tests

DS-NI1010 Test	Mnemonic	Description
1	RESET	Reset Test
2	REGTST	Power-Up Condition
3	CHKCSR	CSR Bit Test
4	PIOTST	Bus Addressing
5	DMATST	Single DMA Transfer
6	NXMTST	Non-Existant Memory Test
7	NMDIAG	Runs NM10A On Board Diagnostics
8	STATST	Checks NM10A Statistics
9	CMDTST	Responds to Simple Commands
10	MLBTST	DMA Transfers (loopback)
11	ILBTST	DMA Transfer Functions (loopback)
12	*NLBTST	Verify On Line Transmission/Reception
13	LTDTST	NM10 Verify of Load Transmit Data
14	QUETST	Verify DMA Buffer Queue
15	CHNTST	Tests Non-Contiguous Buffering
16	BOUNDARY	Checks All Boundary Values
17	NMMEM	Checks NM10A 13.5K Receive FIFO
18	NMEMADD	Check NM10A 13.5K FIFO addressing
20	CRCTST	Check CRC generator
21	NETADD	Check E'net address decoding
22	ADDTST	Check promiscuous mode operation
23	INTTST	Check NI interrupt logic
24	PRITST	Check NI priority logic
25	*STECHO	Send/receive with another station
26	*ENDECHO	Receive/send with another station
27	*HEARTBT	Verify transceiver heartbeat functions

All diagnostic routine error messages are displayed as follows:

"*** Test NAME failed: A B C D ***"

where NAME designates the failing diagnostic routine, and A, B, C, and D display meaningful parameters for the specific test. These numbers are always displayed in octal. In most cases,

A is the actual data,
B is the expected data,
C is an index or mask,
D is the subtest number.

Masks and indices are defined specifically wherever they are used. Many tests require that STATST be run before they will execute. This is indicated by a 0 0 0 0 error return. A description of each test and specific parameters for subtest errors are listed in this section.

TESTNAME: reset

PURPOSE: Verify that NI1010A/2010A recognizes and responds to RESET command.
Verify that NM10A successfully completes its on-board diagnostic.

ASSUMES: Nothing.

RESULTS: NI1010A/2010A -responds to command in CSR,
-causes NM10A reset,
-receives status from ISMI port,
-sets command done.

NM10A -responds to NI1010A/2010A reset,
-executes on board diagnostic,
-passes status through ISMI port.

ERRORS:

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
CSR	200	0	1	Reset Command- DONE TIMEOUT
CSR	200	1	1	Reset Command- STATUS ERROR

TESTNAME: regtst

PURPOSE: Verify condition of NI1010A/NI2010A bus registers after power-up.

ASSUMES: Nothing.
This is a basic check that the board can be read.

RESULTS: Successful completion assures that:

-reading any register (CSR, BAR, BCR, BER) returns all ones (0177777).
This is the undriven bus response.

ERRORS:

	ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
	CSR	0	0	1	Reading CSR did not return 0.
	BAR	0	0	2	Reading BAR did not return 0
	BCR	0	0	3	Reading BCR did not return 0
	BER	0	0	4	Reading BER did not return 0

TESTNAME: chkcsr

PURPOSE: Verify that all read/write bits can be set and cleared.

ASSUMES: Nothing

RESULTS: Only the command and receive interrupt enable bits in the CSR can be read and written.

ERRORS:

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
CSR	120	0	1	Command and/or receive intrpt enable didnt set
CSR	0	0	2	Command and/or receive intrpt enable didnt clr

TESTNAME: piotst

PURPOSE: Check NI1010A/2010A bus addressing, verify that CSR responds only to its address.

ASSUMES: CSR can be read.

RESULTS: CSR is not affected by writing BAR, BCR, or BER. Basic register addressing functions properly.

ERRORS:

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
CSR	0	0	1	Writing BAR caused CSR change
CSR	0	0	2	Writing BER caused CSR change
CSR	0	0	3	Writing BCR caused CSR change

TESTNAME: dmatst

PURPOSE: Verify that NI1010A/2010A can correctly address and write to memory locations, properly executing a single DMA transfer.

ASSUMES: NI1010A/2010A can execute commands and return command done.

RESULTS: NI1010A/2010A -properly executes Load Buffer/Queue command,
-writes all zeros to first location of data buffer,
-returns command done, successfully.

ERRORS:

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
CSR	200	0	1	Supply Rcv Buffer Cmd-DONE TIMEOUT
CSR	200	1	1	Supply Rcv Buffer Cmd-STATUS ERROR
(first word)	0	0	2	NI did not write zeros to first word in DMA buff (buffer location 0)

TESTNAME: nxmtst

PURPOSE: Verify NI1010A/2010A checks that the first location of each DMA buffer is a valid memory location.

ASSUMES: CSR can be read/written.

RESULTS: NI1010A/2010A
-executes Load Buffer/Queue command
-returns command done with nonexistant memory status to signify it could not access that location.
-execute Load Transmit Data command
-returns command done with nonexistant memory status to signify it could not access that location.
-executes Report and Reset Statistics command
-returns command done with nonexistant memory status to signify it could not access that location.

ERRORS:

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
CSR	217	0	1	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	217	1	1	Supply Rcv Buffer Cmd - STATUS ERROR Should be NXM.
CSR	217	0	2	Load Transmit Data Cmd - DONE TIMEOUT
CSR	217	1	2	Load Transmit Data Cmd - STATUS ERROR Should be NXM.
CSR	217	0	3	Report/Reset Stats Cmd - DONE TIMEOUT
CSR	217	1	3	Report/Reset Stats Cmd - STATUS ERROR Should be NXM.

TESTNAME: nmdiag

PURPOSE: Verify NI1010A/2010A correctly passes command to NM10A.
Verify NM10A successfully completes on board diagnostic tests.

ASSUMES: CSR can be read and written.

RESULTS: NI1010A/2010A -passes command and receives status through ISMI port.

NM10A -receives command and passes status through ISMI port.
-completes on-board diagnostic in response to command.

ERRORS:

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
CSR	200	0	1	Run On-Board Diagnostics Cmd- DONE TIMEOUT
CSR	200	1	1	Run On-Board Diagnostics Cmd- STATUS ERROR

TESTNAME: statst

PURPOSE: Verify that statistics DMA transfers (30, 31, 32 commands) are done correctly.
Verify that all other 3x (33-37) commands return Illegal command status.
Allow operator to verify correct Ethernet address, network module type and firmware version number.

ASSUMES: Commands can be executed.

RESULTS: NI1010A/2010A -transfer 3x commands to NM10A
-transfers statistics to memory

NM10A -executes 3x commands properly

Check status and size of 30, 31, and 32 returned blocks. (Returned data starts at buffer offset 2000.)
Print Ethernet address, module type and firmware version number from 30 returned block.
Verify that all other data returned by 30 command and all data returned by 31 command are 0.

Data returned by 32 command is variable.

ERRORS:

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	1	1	Reset command- STATUS ERROR
CSR	200	0	2	Report/Reset stats command - DONE TIMEOUT (030)
CSR	200	1	2	R/R Stats cmd- STATUS ERROR
status byte	0	0(LSB) 1(MSB)	3	R/R Stats returned data DMA block status error
size (LSB)	76	2	3	R/R Stats returned data DMA block size error
size (MSB)	0	3	3	R/R Stats returned data DMA block size error
data	0	index to data	3	R/R Stats returned data DMA data error index= 12-61
CSR	200	0	4	Report Collision Delay Times

				command- DONE TIMEOUT (031)
CSR	200	1	4	RCDT command- STATUS ERROR
Status byte	0	0(LSB) 1(MSB)	5	RCDT returned data DMA block status error
size byte	0	2(LSB) 3(MSB)	5	RCDT returned data DMA block size error
data	377	4	5	Should not write anything here
CSR	200	0	6	Dump NM memory command - DONE TIMEOUT (032)
CSR	200	1	6	Dump NM memory command - STATUS ERROR
Status byte	0	0(LSB) 1(MSB)	7	Dump NM memory command returned DMA block -STATUS ERROR
size (LSB)	0	2	7	Dump NM memory command returned DMA block -SIZE ERROR
size (MSB)	10	3	7	Dump NM memory command returned DMA block -SIZE ERROR
CSR	202	index to cmd	10	Issued command= (33+index) command- DONE TIMEOUT Index= 33-37
CSR	202	index to cmd	11	Issued command= (33+index) -STATUS ERROR did not return ILLEGAL status index= 33-37

TESTNAME: cmdtst

PURPOSE: Verify the NI1010A/2010A correctly responds to all simple commands. These are commands that require only a single transfer to execute and a single transfer to return status.

ASSUMES: CSR can be read.

RESULTS: NI1010A/2010A -returns command done with success for all valid simple commands.
-returns command done with illegal status for all invalid simple commands.

ERRORS:

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
CSR	200	CMD (octal)	1	Indexed command- DONE TIMEOUT Index= 0-23
CSR	200	CMD (octal)	2	Indexed command- STATUS ERROR Should be SUCCESS Index= 0-23
CSR	202	CMD (octal)	3	Indexed command- DONE TIMEOUT Index= 24-27
CSR	202	CMD (octal)	4	Indexed command- STATUS ERROR Should be ILLEGAL Index= 24-27

TESTNAME: mlbtst (Test is run with ISMI loopback set)

PURPOSE : Verify NI1010A/2010A can transfer a DMA buffer from memory to the NM10A, and from the NM10A to memory.
Verify NM10A can accept packet data from ISMI port, correctly count bytes, insert source address and pad zeros to create a minimum length buffer.

ASSUMES : NI1010A/2010A correctly responds to commands.
ISMI port functions.
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.

RESULTS : NI1010A/2010A -Transfer a 46 byte packet from memory
6 bytes- dest address = board address
2 bytes- Interlan diagnostic type
38 bytes- incrementing data

-Transfer a 68 byte packet to memory
2 bytes- packet status
2 bytes- packet size
6 bytes- dest address = board address
6 bytes- source addr = board address
2 bytes- Interlan diagnostic type
38 bytes- incrementing data
10 bytes- NM10A stuffed 0's
4 bytes- CRC (not valid)

-Check that received packet contains correct status, size, addresses, type and data. (CRC is not checked.)

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	1	1	Reset command- STATUS ERROR
CSR	200	0	2	Set Module Lpbk Mode Cmd - DONE TIMEOUT
CSR	200	1	2	Set Module Lpbk Mode Cmd - STATUS ERROR
CSR	200	0	3	Load Transmit Data/Send cmd - TIMEOUT ERROR

CSR	200	1	3	Load Transmit Data/Send cmd - STATUS ERROR
CSR	240	0	4	Supply Rcv Buffer Cmd done and Rcv - DONE TIMEOUT
CSR	240	0	5	Supply Rcv Buffer Cmd or rcv done - STATUS ERROR
CSR	40	0	6	Receive- DONE TIMEOUT
CSR	40	0	7	Receive done- STATUS ERROR
CSR	200	0	10	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	0	11	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index	0	12	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	13	Loopback data error Index=0-47 (0=rbuf[16])
Rec data	0	index to data	14	NM padding error Index= 67-77

TESTNAME: ilbtst (Test is run with internal loopback set)

PURPOSE : Verify NI1010A/2010A can transfer a DMA buffer from memory to the NM10A, and from the NM10A to memory.
Verify NM10A can accept packet data from ISMI port, correctly count bytes, insert source address, pad zeros to create a minimum length buffer, and transfer that packet to the receive RAM, informing the NI1010A/2010A of a received packet.

ASSUMES : NI1010A/2010A correctly responds to commands.
ISMI port functions.
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.

RESULTS :

-Transfer a 46 byte packet from memory
6 bytes- dest address = board address
2 bytes- Interlan diagnostic type
38 bytes- incrementing data

-Transfer a 68 byte packet to memory
2 bytes- packet status
2 bytes- packet size
6 bytes- dest address = board address
6 bytes- source addr = board address
2 bytes- Interlan diagnostic type
38 bytes- incrementing data
10 bytes- NM10A stuffed 0's
4 bytes- CRC (not valid)

-Check that received packet contains correct status, size, addresses, type and data. (CRC is not checked.)

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	1	1	Reset command- STATUS ERROR
CSR	200	0	2	Set Internal Lpbk Mode Cmd - DONE TIMOUT
CSR	200	1	2	Set Internal Lpbk Mode Cmd - STATUS ERROR
CSR	200	0	3	Load Transmit Data/Send cmd - TIMEOUT ERROR

CSR	200	1	3	Load Transmit Data/Send cmd - STATUS ERROR
CSR	240	0	4	Supply Rcv Buffer Cmd done and Rcv - DONE TIMEOUT
CSR	240	0	5	Supply Rcv Buffer Cmd or rcv done - STATUS ERROR
CSR	40	0	6	Receive- DONE TIMEOUT
CSR	40	0	7	Receive done- STATUS ERROR
CSR	200	0	10	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	0	11	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index	0	12	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	13	Loopback data error Index= 0-47
Rec data	0	index to data	14	NM padding error Index= 67-77

TESTNAME: *nlbtst (Test is run with board set online)

PURPOSE : Verify NM10A can transmit through the Ethernet transceiver and receive its own packet. This test causes two packets to be sent to address 2 7 1 0 0 0. The first of these packets contains a CRC error, while the second packet contains valid CRC bytes. See Perform Network Loopback Test command.

WARNING: THIS TEST IS DESIGNED TO FAIL IF COLLISIONS OR BACKOFFS OCCUR.

TEST SHOULD BE RUN ON A QUIET NETWORK.

ASSUMES : NI1010A/2010A correctly responds to commands.
ISMI port functions.

RESULTS :

- Issue Perform Network Loopback Test command.
  NM10A transmits a packet with source and destination addresses of 2 7 1 0 0 0, and bad CRC byte.
  NM10A verifies that packet is received and CRC error is reported.

  NM10A transmits a packet with source and destination addresses of 2 7 1 0 0 0, and good CRC bytes.
  NM10A verifies that packet is received and no CRC error is reported.

  NI1010A/2010A returns status to host.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	1	1	Reset command- STATUS ERROR
CSR	200	0	2	Go ONLINE cmd- DONE TIMOUT
CSR	200	1	2	ONLINE cmd- STATUS ERROR
CSR	200	0	3	Perform Network Loopback Test cmd- TIMEOUT ERROR
CSR	200	1	3	Perform Network Loopback Test cmd- STATUS ERROR

TESTNAME: ltdtst (Test is run with internal loopback set)

PURPOSE : Verify NM10A distinguishes Load Transmit Data command from Load Transmit Data and Send command.
Verify Load Transmit Data/Send causes all previous data to be transmitted.

ASSUMES : Internal loopback functions properly.
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.

RESULTS :

-Transfer a 46 byte packet from memory with Load Transmit Data command
6 bytes- dest address = board address
2 bytes- Interlan diagnostic type
38 bytes- incrementing data

-Check packet is not received

-Transfer a 0 byte packet from memory with Load _Send command

-Transfer a 68 byte packet to memory
2 bytes- packet status
2 bytes- packet size
6 bytes- dest address = board address
6 bytes- source addr = board address
2 bytes- Interlan diagnostic type
38 bytes- incrementing data
10 bytes- NM10A stuffed 0's
4 bytes- CRC (not valid)

-Check that received packet contains correct status, size, addresses, type and data. (CRC is not checked.)

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	0	2	Set Internal Lpbk Mode Cmd - DONE TIMEOUT
CSR	200	1	2	Set Internal Lpbk Mode Cmd - STATUS ERROR
CSR	200	0	3	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	1	3	Supply Rcv Buffer Cmd - STATUS ERROR
CSR	200	0	4	Load Transmit Data Cmd - DONE TIMEOUT
CSR	200	1	4	Load Transmit Data Cmd - STATUS ERROR
CSR	0	0	5	No Receive done expected (No Load Transmit Data/Send command)
CSR	240	0	6	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	7	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	10	Receive- DONE TIMEOUT
CSR	40	0	11	Receive done- STATUS ERROR
CSR	200	0	12	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	13	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	14	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err

Rec Data	Xmit Data	index to data	15	Loopback data error Index= 0-47
Rec data	0	index to data	16	NM padding error Index= 67-77

TESTNAME: quetst (Test is run with ISMI loopback set)

PURPOSE : Verify buffer/queue counters function properly after overloading or when full.
Verify Flush Buffer/Queue instruction functions correctly.
Verify that buffer/queue loads correctly by issuing Dump Buffer/Queue command.

ASSUMES : ISMI loopback functions
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.
(unless otherwise noted)

RESULTS :
-Buffer/queue accepts only one received packet after getting 17 buffer locations, putting data in last location (starts at buffer offset = TBD).
-Buffer/queue accepts waiting packet after getting one buffer location.
-Buffer/queue accepts received packet after getting 16 buffer locations, putting data in first location.
-Buffer/queue accepts no packets after receiving Flush Buffer command.
-Dump buffer command returns 16 buffers with correct order and data.

-All received packets are checked for correct status, size, addresses, type and data.
-All packets are identical to those used in loopback tests.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	0	2	Set Module Lpbk Mode Cmd - DONE TIMEOUT
CSR	200	1	2	Set Module Lpbk Mode Cmd - STATUS ERROR
CSR	200	index	3	Supply Rcv Buffer Cmd - DONE TIMEOUT index=position in queue (index= 0-16)
CSR	200	index	4	Supply Rcv Buffer Cmd

				- STATUS ERROR index=position in queue (index= 0-16)
CSR	240	0	5	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	6	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	7	Receive- DONE TIMEOUT
CSR	40	0	10	Receive done- STATUS ERROR
CSR	200	0	11	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	12	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	13	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	14	Loopback data error Index= 0-47
Rec data	0	index to data	15	NM padding error Index= 67-77
CSR	200	0	16	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	1	16	Load Transmit Data/Send cmd - STATUS ERROR
CSR	0	0	17	No receive done expected (No buffers in queue)
CSR	240	0	20	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	0	21	Supply Rcv Buffer Cmd or rcv done- STATUS ERROR
CSR	40	0	22	Receive- DONE TIMEOUT
CSR	40	0	23	Receive done- STATUS ERROR
CSR	200	0	24	Supply Rcv Buffer Cmd - DONE TIMEOUT

				- STATUS ERROR index=position in queue (index= 0-16)
CSR	240	0	5	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	6	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	7	Receive- DONE TIMEOUT
CSR	40	0	10	Receive done- STATUS ERROR
CSR	200	0	11	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	12	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	13	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	14	Loopback data error Index= 0-47
Rec data	0	index to data	15	NM padding error Index= 67-77
CSR	200	0	16	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	1	16	Load Transmit Data/Send cmd - STATUS ERROR
CSR	0	0	17	No receive done expected (No buffers in queue)
CSR	240	0	20	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	0	21	Supply Rcv Buffer Cmd or rcv done- STATUS ERROR
CSR	40	0	22	Receive- DONE TIMEOUT
CSR	40	0	23	Receive done- STATUS ERROR
CSR	200	0	24	Supply Rcv Buffer Cmd - DONE TIMEOUT

CSR	200	0	25	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index	0	26	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	27	Loopback data error Index= 0-47
Rec data	0	index to data	30	NM padding error Index= 67-77
CSR	200	index	31	Supply Rcv Buffer Cmd - DONE TIMEOUT index= queue position (index= 0-15)
CSR	200	index	32	Supply Rcv Buffer Cmd - STATUS ERROR index= queue position (index= 0-15)
CSR	240	0	33	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	34	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	35	Receive- DONE TIMEOUT
CSR	40	0	36	Receive done- STATUS ERROR
CSR	200	0	37	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	40	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	41	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	42	Loopback data error Index= 0-47
Rec data	0	index to data	43	NM padding error Index= 67-77

CSR	200	0	44	Flush buf/Q cmd- DONE TIMEOUT
CSR	200	1	44	Flush Buf/Q cmd- STATUS ERROR
CSR	200	0	45	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	0	0	46	No receive done expected (Queue empty)
CSR	240	0	47	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	0	50	Supply Rcv Buffer Cmd or rcv done- STATUS ERROR
CSR	40	0	51	Receive- DONE TIMEOUT
CSR	40	0	52	Receive done- STATUS ERROR
CSR	200	0	53	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	0	54	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	Xmit data	index	55	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	56	Loopback data error Index=0-50
Rec data	0	index to data	57	NM padding error Index= 67-77
CSR	200	index	60	Supply Rcv Buffer Cmd - DONE TIMEOUT Index= position in queue (index= 0-15)
CSR	200	index	61	Supply Rcv Buffer Command - STATUS ERROR Index= position in queue (index= 0-15)
CSR	200	0	62	Dump Buf/Q cmd- DONE TIMEOUT
CSR	200	1	62	Dump Buf/Q cmd- STATUS ERROR

Dump Data	Exp Data	index	63	LSByte of Queue address wrong index= position in queue (index= 0-15)
Dump Data	Exp Data	index	64	middle byte of Queue address wrong index= position in queue (index= 0-15)
Dump Data	Exp Data	index	65	MSByte of Queue address wrong index= position in queue (index= 0-15)
Dump Data	Exp Data	index	66	LSByte of Queue count wrong index= position in queue (index= 0-15)
Dump Data	Exp Data	index	67	MSByte of Queue count wrong index= position in queue (index= 0-15)

TESTNAME: chntst (Test is run with internal loopback set)

PURPOSE : Verify that octal length buffers in the queue will allow received packets to be transferred to non-contiguous buffers when received packet size is greater than buffer size.

ASSUMES : Internal loopback functions.
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.
(unless otherwise noted)

RESULTS :

-Buffer/queue is loaded with three (3) 128-byte buffers located at:
buffer offset = 2000
buffer offset = 2130
buffer offset = 2260

-Transmit data is 256 bytes:
6 bytes- dest addr = board address
2 bytes- Interlan diagnostic type
248 bytes- incrementing data

-Received data is 266 bytes:
buffer 1:
2 bytes- status
2 bytes- size
6 bytes- dest addr = board address
6 bytes- source addr= board address
112 bytes- incrementing data

buffer 2:
128 bytes- incrementing data

buffer 3:
10 bytes- incrementing data
4 bytes- CRC (not valid)

-Received packet is checked for correct status, size, addresses, type and data.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	1	1	Reset command- STATUS ERROR
CSR	200	0	2	Set Internal Lpbk Mode Cmd

				- DONE TIMEOUT
CSR	200	1	2	Set Internal Lpbk Mode Cmd - STATUS ERROR
CSR	200	index	3	Supply Rcv Buffer Cmd - DONE TIMEOUT index= queue position index= 0-2
CSR	200	index	4	Supply Rcv Buffer Cmd - STATUS ERROR index= queue position index= 0-2
CSR	240	0	5	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	6	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	7	Receive- DONE TIMEOUT
CSR	40	0	10	Receive done- STATUS ERROR
CSR	200	0	11	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	12	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	13	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	14	Loopback data error (first buffer) Index= 0-157 (0= rbuf[16])
Rec Data	Xmit Data	index to data	15	Loopback data error (second buffer) Index= 0-177 (0= rbuf[130])
Rec Data	Xmit Data	index to data	16	Loopback data error (last buffer) Index= 0-11 (0= rbuf[260])

TESTNAME: boundary (Test is run with internal loopback set)

PURPOSE : Verify NI1010A/2010A functions properly in boundary cases.

ASSUMES : Internal loopback functions.
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.

RESULTS :

-All commands should execute successfully and all data should be returned correctly for the following conditions:

COND0
-Receive buffer = 68 (min packet size)
Transmit buffer= 56 (fills min packet)

COND1
-Receive buffer = 142
Transmit buffer= 100(even byte packet)

COND2
-Receive buffer = 142
Transmit buffer= 99(odd byte packet)

COND3
-Receive buffer = 142
Transmit buffer= 128(octal lenght packet)

-These transmit buffers should be rejected as too small:

COND5
-Transmit buffer= 0
COND6
-Transmit buffer= 7

-This non-octal length receive buffer should cause excess receive data to be discarded:

-Receive buffer = 99
Transmit buffer= 99

-This transmit buffer should be rejected as too big:

-Transmit buffer= 1600

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.

CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	1	1	Reset command- STATUS ERROR
CSR	200	0	2	Set Internal Lpbk Mode Cmd - DONE TIMEOUT
CSR	200	1	2	Set Internal Lpbk Mode Cmd - STATUS ERROR
CSR	200	index to boundary cond index= 0-3	3	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	index to boundary cond index= 0-3	4	Load Transmit Data/Send cmd - STATUS ERROR
CSR	240	index to boundary cond index= 0-3	5	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	index to boundary cond index= 0-3	6	Supply Rcv Buffer Cmd or rcv done- STATUS ERROR
CSR	40	index to boundary cond index= 0-3	7	Receive- DONE TIMEOUT
CSR	40	index to boundary cond index= 0-3	10	Receive done- STATUS ERROR
CSR	200	index to boundary cond index= 0-3	11	Supply Rcv Buffer Cmd DONE TIMEOUT
CSR	200	index to boundary cond index= 0-3	12	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index1	index2	13	Loopback error: index1= 0,1 status error

				2,3 size error 4-11 dest addr err 12-17 srce addr err index2= boundary condition
Rec Data	Xmit Data	index to boundary condition index= 0-3	14	Loopback data error
CSR	206	index to boundary cond (index= 5-6)	15	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	206	index to boundary cond (index= 5-6)	16	Load Transmit Data/Send cmd - STATUS ERROR
CSR	200	0	17	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	1	17	Load Transmit Data/Send cmd - STATUS ERROR
CSR	240	0	20	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	0	21	Supply Rcv Buffer Cmd or rcv done- STATUS ERROR
CSR	40	0	22	Receive- DONE TIMEOUT
CSR	40	0	23	Receive done- STATUS ERROR
CSR	200	0	24	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	0	25	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index	0	26	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	27	Loopback data error index= 0-132
CSR	200	0	30	Supply Rcv Buffer Cmd - DONE TIMEOUT

CSR	200	1	30	Supply Rcv Buffer Cmd - STATUS ERROR
CSR	0	0	31	No receive done expected
CSR	205	0	32	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	205	1	32	Load Transmit Data/Send cmd - STATUS ERROR Should be EXCEEDED

TESTNAME: nmmem (Test is run with internal loopback set)

PURPOSE : Verify that NM10A 13.5K byte receive data memory is good.
(Not guaranteed to find addressing problems.)

ASSUMES : ISMI loopback functions
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.
(unless otherwise noted)

RESULTS :

-Transmit packet is 1508 bytes:
6 bytes- dest addr = board address
2 bytes- Interlan diagnostic type
1500 bytes- 0

-Received packet is 1518 bytes:
2 bytes- status
2 bytes- size
6 bytes- dest addr = board address
6 bytes- source addr=board address
1500 bytes- 0
4 bytes- CRC (not valid)

-Check correct received status, size, addresses, and data.

-Repeat 21 times to check every location.

-Transmit packet is 1508 bytes:
6 bytes- dest addr = board address
2 bytes- Interlan diagnostic type
1500 bytes- 0377

-Received packet is 1518 bytes:
2 bytes- status
2 bytes- size
6 bytes- dest addr = board address
6 bytes- source addr=board address
1500 bytes- 0377
4 bytes- CRC (not valid)

-Check correct received status, size, addresses, and data.

-Repeat 21 times to check every location.

-Transmit packet is 1508 bytes:
6 bytes- dest addr = board address
2 bytes- Interlan diagnostic type
1500 bytes- 0125

-Received packet is 1518 bytes:
2 bytes- status

```
                    2 bytes- size
                    6 bytes- dest addr = board address
                    6 bytes- source addr=board address
                 1500 bytes- 0125
                    4 bytes- CRC (not valid)
```

-Check correct received status, size, addresses, and data.

-Repeat 21 times to check every location.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset command- DONE TIMEOUT
CSR	200	1	1	Reset command- STATUS ERROR
CSR	200	0	2	Set Internal Lpbk Mode Cmd - DONE TIMEOUT
CSR	200	1	2	Set Internal Lpbk Mode Cmd - STATUS ERROR
CSR	200	index to mem slice index= 0-20	3	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	index to mem slice index= 0-20	4	Load Transmit Data/Send cmd - STATUS ERROR
CSR	240	index to mem index= 0-20	5	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	index to mem index= 0-20	6	Supply Rcv Buffer Cmd or Rcv done- STATUS ERROR
CSR	40	index to mem slice index= 0-20	7	Receive- DONE TIMEOUT
CSR	40	index	10	Receive done- STATUS ERROR

		to mem slice index= 0-20		
CSR	200	index to mem slice index= 0-20	11	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	index to mem slice index= 0-20	12	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index1	index2	13	Loopback error: index1 = 0,1 status error 2,3 size error, 4-11 dest addr error 12-17 source add error index2 to mem slice= 0-20
Rec Data	Xmit Data	index to mem slice = 0-20	14	Loopback data error
CSR	200	index to mem slice index= 0-20	15	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	index to mem slice index= 0-20	16	Load Transmit Data/Send cmd - STATUS ERROR
CSR	240	index to mem index= 0-20	17	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	index to mem index= 0-20	20	Supply Rcv Buffer Cmd or Rcv done- STATUS ERROR
CSR	40	index to mem slice index= 0-20	21	Receive- DONE TIMEOUT
CSR	40	index to mem slice index= 0-20	22	Receive done- STATUS ERROR

CSR	200	index to mem slice index= 0-20	23	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	index to mem slice index= 0-20	24	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index1	index2	25	Loopback error: index1 = 0,1 status error 2,3 size error, 4-11 dest addr error 12-17 source add error index2 to mem slice= 0-20
Rec Data	Xmit Data	index to mem slice = 0-20	26	Loopback data error
CSR	200	index to mem slice index= 0-20	27	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	index to mem slice index= 0-20	30	Load Transmit Data/Send cmd - STATUS ERROR
CSR	240	index to mem index= 0-20	31	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	index to mem index= 0-20	32	Supply Rcv Buffer Cmd or Rcv done- STATUS ERROR
CSR	40	index to mem slice index= 0-20	33	Receive- DONE TIMEOUT
CSR	40	index to mem slice index= 0-20	34	Receive done- STATUS ERROR
CSR	200	index to mem slice index= 0-20	35	Supply Rcv Buffer Cmd - DONE TIMEOUT

CSR	200	index to mem slice index= 0-20	36	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index1	index2	37	Loopback error: index1 = 0,1 status error 2,3 size error, 4-11 dest addr error 12-17 source add error index2 to mem slice= 0-20
Rec Data	Xmit Data	index to mem slice = 0-20	40	Loopback data error

TESTNAME: nmemadd (Test is run with internal loopback set)

PURPOSE : Check addressing and data is correct on NM10A receive FIFO

ASSUMES : Internal loopback functions
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.
(unless otherwise noted)

RESULTS : -Transmit 177 64-byte packets with data incrementing between packets.
(I.E., packet 0 has data=0, packet 176 data= 260(octal)).

-Receive first transmitted packet (packet 0).
Check that status, size and data are correct.
Transmit next sequential packet (177).
Repeat this until 208 total packets have been transmitted.

-Receive next sequential packet (32).
Check that status, size and data are correct.
Repeat until all 208 packets have been received.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset Command- DONE TIMEOUT
CSR	200	1	1	Reset Cmd- STATUS ERROR
CSR	200	0	2	Set Internal Lpbk Mode Cmd - DONE TIMEOUT
CSR	200	1	2	Set Internal Lpbk Mode Cmd - STATUS ERROR
CSR	200	index 0-260	3	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	index 0-260	4	Load Transmit Data/Send Cmd - STATUS ERROR
CSR	240	index	5	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT

				Index= 0-41
CSR	240	Index	6	Supply Rcv Buffer Cmd or Rcv done- STATUS ERROR Index= 0-41
CSR	40	Index	7	Receive- DONE TIMEOUT Index= 0-41
CSR	40	Index	10	Receive done- STATUS ERROR Index= 0-41
CSR	200	Index	11	Supply Rcv Buffer Cmd - DONE TIMEOUT Index= 0-41
CSR	200	Index	12	Supply Rcv Buffer Cmd - STATUS ERROR Index= 0-41
Rec data	index	0	13	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	14	Loopback data error
Rec Data	Xmit Data	index to data	15	Loopback data error index= 22-77
CSR	200	index	16	Load Transmit Data/Send Cmd - DONE TIMEOUT Index= 261-320
CSR	200	index	17	Load Transmit Data/Send Cmd - STATUS ERROR INdex= 261-320
CSR	240	Index	20	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT Index= 41-320
CSR	240	Index	21	Supply Rcv Buffer Cmd or Rcv done- STATUS ERROR Index= 41-320
CSR	40	Index	22	Receive- DONE TIMEOUT Index= 41-320
CSR	40	Index	23	Receive done- STATUS ERROR Index= 41-320

CSR	200	Index	24	Supply Rcv Buffer Cmd - DONE TIMEOUT Index= 41-320
CSR	200	Index	25	Supply Rcv Buffer Cmd - STATUS ERROR Index= 41-320
Rec data	index	0	26	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	27	Loopback data error index= 0-2
Rec Data	Xmit Data	index to data	30	Loopback data error index= 22-77

TESTNAME: crctst (Test is run with internal loopback set)

PURPOSE : This test has no effect on the NM10A.

TESTNAME: netadd (Test is run with internal loopback set)

PURPOSE : Verify that the board responds correctly to valid
broadcast, multicast and board addresses.
Verify board does not respond to invalid addresses.

ASSUMES : Internal loopback functions
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.
(unless otherwise noted)

RESULTS : -Transmit a 64-byte packet to braodcast address (= FF FF FF).
Packet should be received correctly.

-Transmit a 64-byte packet to board address + 1.
Nothing should be received.

-Load group address = board address + 1 + multicast bit
Transmit packet to that address.
Packet should be received correctly.

-Transmit to different group address
(= bd address + multicast bit).
Nothing should be received.

-Delete previously loaded group address.
Transmit to that address.
Nothing should be received.

For Version 2 or greater NM10As:

-Load physical address =board address + 1
Transmit packet to that address.
Packet should be received correctly.

-Delete physical address
Transmit packet to board address+1.
Nothing should be received.

-Inhibit source address insertion
Transmit a packet from a buffer containing:
Board address board address+1 type data
Packet should be received with a crc error and
board address+1 as the source address.

-Set source address insertion.
Transmit a packet to board address.
Packet should be received with board
address as source address

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset Command- DONE TIMEOUT
CSR	200	1	1	Reset Command- STATUS ERROR
CSR	200	0	2	Set Internal Lpbk Mode Cmd - DONE TIMEOUT
CSR	200	1	2	Set Internal Lpbk Mode Cmd - STATUS ERROR
CSR	200	0	3	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	1	3	Load Transmit Data/Send Cmd - STATUS ERROR (Broadcast address)
CSR	240	0	4	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	0	5	Supply Rcv Buffer Cmd or Rcv done- STATUS ERROR
CSR	40	0	6	Receive- DONE TIMEOUT
CSR	40	0	7	Receive done- STATUS ERROR
CSR	200	0	10	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	0	11	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index	0	12	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	13	Loopback data error Index= 0-61 (0=rbuf[16])
CSR	200	0	14	Load Transmit Data/Send Cmd - DONE TIMEOUT

CSR	200	1	14	Load Transmit Data/Send Cmd - STATUS ERROR
CSR	200	0	15	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	1	15	Supply Rcv Buffer Cmd -STATUS ERROR
CSR	0	0	16	No data expected.Wrong address
CSR	200	0	17	Load Group Addr Cmd - DONE TIMEOUT
CSR	200	1	17	Load Group Addr Cmd - STATUS ERROR
CSR	240	0	20	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	21	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	22	Receive- DONE TIMEOUT
CSR	40	0	23	Receive done- STATUS ERROR
CSR	200	0	24	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	25	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	26	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	27	Loopback data error Index= 0-61
CSR	200	0	30	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	1	30	Load Transmit Data/Send Cmd - STATUS ERROR
CSR	200	0	31	Supply Rcv Buffer command - DONE TIMEOUT
CSR	200	1	31	Supply Rcv Buffer command

				- STATUS ERROR
CSR	0	0	32	No receive data expected. Wrong group address.
CSR	200	0	33	Delete Group Addr Cmd - DONE TIMOUT
CSR	200	1	33	Delete Group Addr Cmd - STATUS ERROR
CSR	200	0	34	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	1	34	Load Transmit Data/Send Cmd - STATUS ERROR
CSR	200	0	35	Supply Rcv Buffer command - DONE TIMEOUT
CSR	200	1	35	Supply Rcv Buffer command - STATUS ERROR
CSR	0	0	36	No receive data expected. No group addresses loaded.
CSR	200	0	37	Load Physical Addr Cmd - DONE TIMEOUT
CSR	200	1	37	Load Physical Addr Cmd - STATUS ERROR
CSR	240	0	40	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	41	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	42	Receive- DONE TIMEOUT
CSR	40	0	43	Receive done- STATUS ERROR
CSR	200	0	44	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	45	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	46	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err

Rec Data	Xmit Data	index to data	47	Loopback data error Index= 0-61
CSR	200	0	50	Delete Physical Addr Cmd - DONE TIMEOUT
CSR	200	1	50	Delete Physical Addr Cmd - STATUS ERROR
CSR	200	0	51	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	1	51	Load Transmit Data/Send Cmd - STATUS ERROR
CSR	200	0	52	Supply Rcv Buffer command - DONE TIMEOUT
CSR	200	1	52	Supply Rcv Buffer command - STATUS ERROR
CSR	0	0	53	No receive data expected. Wrong address.
CSR	200	0	54	Inhibit Source Address Insertion Cmd - DONE TIMEOUT
CSR	200	1	54	Inhibit Source Address Insertion Cmd - STATUS ERROR
CSR	240	0	55	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	56	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	57	Receive- DONE TIMEOUT
CSR	40	0	60	Receive done- STATUS ERROR
CSR	200	0	61	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	62	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	63	Loopback error: index= 0,1 status error 2,3 size error
Rec Data	Xmit Data	index to data	64	Loopback data error Index= 0-67

CSR	200	0	65	Set Source Address Insertion Cmd - DONE TIMEOUT
CSR	200	1	65	Set Source Address Insertion Cmd - STATUS ERROR
CSR	240	0	67	Load Transmit Data/Send cmd done and Rcv- DONE TIMEOUT
CSR	240	0	70	Load Transmit Data/Send cmd or rcv done- STATUS ERROR
CSR	40	0	71	Receive- DONE TIMEOUT
CSR	40	0	72	Receive done- STATUS ERROR
CSR	200	0	73	Load Transmit Data/Send cmd - DONE TIMEOUT
CSR	200	0	74	Load Transmit Data/Send cmd - STATUS ERROR
Rec data	index	0	75	Loopback error: index= 0,1 status error 2,3 size error
Rec Data	Xmit Data	index to data	76	Loopback data error destination address
Rec Data	Xmit Data	index to data	77	Loopback data error source address
Rec Data	Xmit Data	index to data	100	Loopback data error data

TESTNAME: addtst (Test is run with internal loopback set)

PURPOSE : Verify that promiscuous mode works properly.

ASSUMES : Internal loopback functions
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.
(unless otherwise noted)

RESULTS : -Set promiscuous mode.
Transmit 64-byte packet to board address + 1.
Packet should be received correctly.

-Clear promiscuous mode.
Transmit 64-byte packet to board address + 1.
Nothing should be received.

For Version 2 or greater NM10A's

-Set receive all multicast
Transmit 64-byte packet to board address + multicast bit.
Packet should be received correctly.

-Clear receive all multicast
Transmit 64-byte packet to board address + multicast bit.
Nothing should be received.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset Command- DONE TIMEOUT
CSR	200	1	1	Reset Command- STATUS ERROR
CSR	200	0	2	Set Internal Lpbk Mode Cmd - DONE TIMEOUT
CSR	200	1	2	Set Internal Lpbk Mode Cmd - STATUS ERROR
CSR	200	0	3	Set Promiscuous Mode Cmd - DONE TIMEOUT
CSR	200	1	3	Set Promiscuous Mode Cmd - STATUS ERROR

CSR	200	0	4	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	1	4	Load Transmit Data/Send Cmd - STATUS ERROR
CSR	240	0	5	Supply Rcv Buffer Cmd done and Rcv- DONE TIMEOUT
CSR	240	0	6	Supply Rcv Buffer Cmd or Rcv done- STATUS ERROR
CSR	40	0	7	Receive- DONE TIMEOUT
CSR	40	0	10	Receive done- STATUS ERROR
CSR	200	0	11	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	0	12	Supply Rcv Buffer Cmd - STATUS ERROR
Rec data	index	0	13	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	14	Loopback data error Index= 0-57
CSR	200	0	15	Clear Promiscuous Mode Cmd - DONE TIMOUT
CSR	200	1	15	Clear Promiscuous Mode Cmd - STATUS ERROR
CSR	200	0	16	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	1	16	Load Transmit Data/Send Cmd - STATUS ERROR
CSR	200	0	17	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	1	17	Supply Rcv Buffer Cmd - STATUS ERROR
CSR	0	0	20	No receive expected. (Wrong address)

CSR	200	0	21	Set Receive All Multicast - DONE TIMEOUT
CSR	200	1	21	Set Receive All Multicast - STATUS ERROR
CSR	240	0	22	Load Transmit Data and Send Cmd done and Rcv- DONE TIMEOUT
CSR	240	0	23	Load Transmit Data and Send Cmd or Rcv done- STATUS ERROR
CSR	40	0	24	Receive- DONE TIMEOUT
CSR	40	0	25	Receive done- STATUS ERROR
CSR	200	0	26	Load Transmit Data and Send Cmd - DONE TIMEOUT
CSR	200	0	27	Load Transmit Data and Send Cmd - STATUS ERROR
Rec data	index	0	30	Loopback error: index= 0,1 status error 2,3 size error 4-11 dest addr err 12-17 srce addr err
Rec Data	Xmit Data	index to data	31	Loopback data error Index= 0-61 (0=rbuf[16])
CSR	200	0	32	Clear Receive All Multicast - DONE TIMEOUT
CSR	200	1	32	Clear Receive All Multicast - STATUS ERROR
CSR	200	0	33	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	1	33	Load Transmit Data/Send Cmd - STATUS ERROR
CSR	200	0	34	Supply Rcv Buffer Cmd - DONE TIMEOUT
CSR	200	1	34	Supply Rcv Buffer Cmd -STATUS ERROR
CSR	0	0	35	No data expected.Wrong address

TESTNAME: inttst (Test is run with internal loopback set)

PURPOSE : Verify the NI board interrupts at the correct vector addresses and only when appropriate interrupts are enabled.

ASSUMES : Internal loopback functions
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.
(unless otherwise noted)

RESULTS : -Enable command interrupts.
Issue Supply Rcv Buffer Command.
Wait for interrupt at command vector.
Check command done with correct interrupt.

Issue Load Transmit Data/Send command.
Wait for interrupt at command vector.
Check command done with correct interrupt.
Check receive done with no interrupt.

-Enable receive interrupts.
Issue Supply Rcv Buffer Command.
Wait for command done- check no interrupt.

Issue Load Transmit Data/Send command.
Wait for receive interrupt.
Check receive done with correct interrupt.
Check command done with no interrupt.

-Enable both interrupts.
Issue Supply Rcv Buffer Command.
Wait for command interrupt.
Check command done with correct interrupt.

Issue Load Transmit Data/Send command.
Wait for receive and command interrupts.
Check receive and command dones.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset Command- DONE TIMEOUT
CSR	200	1	1	Set Internal Lpbk Mode Cmd - DONE TIMEOUT

INT FLAG	1	0	2	Supply Rcv Buffer Cmd INTERRUPT TIMEOUT
CSR	300	0	3	Supply Rcv Buffer command - DONE ERROR
INT FLAG	1	0	4	Load Transmit Data/Send Cmd INTERRUPT TIMEOUT
CSR	300	MASK= 0337	5	Load Transmit Data/Send Cmd - DONE ERROR
CSR	40	0	6	Receive- DONE TIMEOUT
CSR	40	MASK= 0277	7	Receive- STATUS ERROR
CSR	200	0	10	Supply Rcv Buffer command - DONE TIMEOUT
CSR	200	MASK= 0357	11	Supply Rcv Buffer cmd - STATUS ERROR
INT FLAG	0	1	11	No interrupt expected.
INT FLAG	2	0	12	Receive INTERUPT TIMEOUT
CSR	60	MASK= 0177	13	Receive- STATUS ERROR
CSR	200	0	14	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	200	MASK= 0357	15	Load Transmit Data/Send Cmd - STATUS ERROR
INT FLAG	0	1	15	No interrupt expected.
INT FLAG	1	0	16	Supply Rcv Buffer Cmd INTERRUPT TIMEOUT
CSR	300	MASK= 0357	17	Supply Rcv Buffer cmd - STATUS ERROR
INT FLAG	3	0	20	Neither interrupt occurred Timeout
CSR	160	40	21	Rcv int without Rcv done

CSR	320	200	22	Cmd int without Cmd done
INT FLAG	3	0	23	Int flag contains bad data
INT FLAG	3	0	24	Neither interrupt occurred timeout
CSR	160	40	25	Rcv int without Rcv done
CSR	320	200	26	Cmd int without Cmd done
INT FLAG	3	0	27	Int flag contains bad data

TESTNAME: pritst (Test is run with internal loopback set)

PURPOSE : Verify NI board obeys bus priority rules.
THIS TEST MAY FAIL IF PRIORITY CHANGES HAVE BEEN MADE ON BOARD BUT NOT PROPERLY SELECTED AT BOOT TIME.

ASSUMES : Internal loopback functions
Transmitted information starts at buffer offset= 0.
Received information starts at buffer offset= 2000.
(unless otherwise noted)

RESULTS : -Enable command interrupt.
Set processor status word (PSW) priority to board priority.
Issue Supply Rcv Buffer Command.
Wait for command done- check no interrupt.
Set priority to one less than board priority.
Wait for command interrupt.

-Enable receive interrupt.
Set processor status word (PSW) priority to board priority.
Issue Load Transmit Data/Send command.
Wait for receive done- check no interrupt.
Set priority to one less than board priority.
Wait for receive interrupt.
Wait for command done.

-Repeat steps above with high priority = 7.
low priority = 0.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	Reset Comand- DONE TIMEOUT
CSR	200	0	2	Set Internal Lpbk Mode Cmd -DONE TIMEOUT
CSR	300	0	3	Supply Rcv Buffer cmd - DONE TIMEOUT
CSR	300	0	4	Supply Rcv Buffer cmd - STATUS ERROR
INT FLAG	0	0	5	No interrupt expected. Priority= board
INT FLAG	1	0	6	Command Int- TIMEOUT ERROR Priority < board

CSR	60	0	7	Receive- DONE TIMEOUT
INT FLAG	0	1	10	No interrupt expected Priority= board
INT FLAG	2	0	11	Receive Int- TIMEOUT ERROR Priority < board
CSR	220	0	12	Load Transmit Data/Send Cmd - DONE TIMEOUT
CSR	300	0	13	Supply Rcv Buffer cmd - DONE TIMEOUT
CSR	300	0	14	Supply Rcv Buffer cmd - STATUS ERROR
INT FLAG	0	1	15	No interrupt expected. Priority= 7
INT FLAG	1	0	16	Command Int- TIMEOUT ERROR Priority= 0
CSR	60	0	17	Receive- DONE TIMEOUT
INT FLAG	0	1	20	No interrupt expected Priority= 7
INT FLAG	2	0	21	Receive Int- TIMEOUT ERROR Priority= 0
CSR	220	0	22	Load Transmit Data/Send Cmd - DONE TIMEOUT

TESTNAME: *stecho (This test is run online and must be paired with a station running *endecho.)

PURPOSE: Verify that node can correctly send and receive packets from another specified node.

ASSUMES: All other diagnostic tests pass, including *nlbtst.
Only the node running *endecho is addressing frames to this node.

RESULTS: Ask user for destination address. This should be entered most significant byte first, i.e. 2-7-1-x-x-x.
Ask data type and size of frame, user must answer.
*endecho must be running before hitting the next <CR>.
A packet of user-defined length is sent to the address specified by the user.

Wait for packet to be returned with destination and source addresses exchanged. Check each data byte.
Report and halt on error.
Else repeat, starting with frame transmission.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	RESET cmd- done timeout error
CSR	200	1	1	RESET cmd- status error
CSR	200	0	2	GO ONLINE cmd- done timeout error
CSR	200	1	2	GO ONLINE cmd- status error
CSR	200	0	3	LOAD TRANSMIT DATA/SEND cmd- done timeout error
CSR	200	1	3	LOAD TRANSMIT DATA/SEND cmd- status error
RCV'D DATA	0	index	4	Rec'd frame status error Index= 0,1 to status bytes
RCV'D DATA	SIZE	0	5	Rec'd frame size error
RCV'D DATA	BD ADDR BYTE	index	6	Rec'd frame dest addr error index= 4-9, rcv'd dest address

RCV'D DATA	DEST ADDR	index	7	Rec'd frame source addr error index= 10-15, rcv'd source addr
RCV'D DATA	EXP DATA	index	10	Rec'd frame data error index= 16-size, rcv'd data
0	0	0	11	LOAD TRANSMIT DATA/SEND cmd-done timeout
CSR	200	0	12	LOAD TRANSMIT DATA/SEND cmd-status error

TESTNAME: *endecho (This test is run online and must be paired with a station running *stecho.)

PURPOSE: Verify that node can correctly send and receive packets from another specified node.

ASSUMES: All other diagnostic tests pass, including *nlbtst.
Only the node running *stecho is addressing frames to this node.

RESULTS: This test must be running before *stecho actually sends a frame on the wire.
Wait for a frame to be received that is addressed to this node.
Exchange source address and destination address fields, truncate the 4 CRC bytes and re-transmit the packet.
Wait for another frame to be received that is addressed to this node.
Compare the new frame to the last frame received.
Report and halt on error.
Else, repeat starting with re-transmission.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	RESET cmd- done timeout error
CSR	200	1	1	RESET cmd- status error
CSR	200	0	2	GO ONLINE cmd- done timeout error
CSR	200	1	2	GO ONLINE cmd- status error
CSR	200	0	3	SET RCV ON ERR cmd- done timeout
CSR	200	1	3	SET RCV ON ERR cmd- status error
CSR	200	0	4	LOAD TRANSMIT DATA/SEND cmd- done timeout
CSR	200	1	4	LOAD TRANSMIT DATA/SEND cmd- status error
RCV'D DATA	EXP DATA	index	5	Rec'd frame does not match previous frame. Index to error in frame

TESTNAME: *heartbt (This test is run online and requires connection to a transceiver with heartbeat response.)

PURPOSE: Verify that transceiver and board collision detect circuits are properly functionsing.

ASSUMES: All other diagnostic tests pass, including *nlbtst.

RESULTS: Issue Perform Collision Detect Test command.
NM10A will transmit a minimum length packet to itself.
The transceiver should respond by asserting the collision detect signal 1-2 milliseconds after the end of the packet.

ERRORS :

ACTUAL DATA (A)	EXP DATA (B)	INDEX (C)	SUBTEST (D)	
0	0	0	0	STATST must be run once before this test will pass.
CSR	200	0	1	RESET cmd- done timeout error
CSR	200	1	1	RESET cmd- status error
CSR	200	0	2	GO ONLINE cmd- done timeout error
CSR	200	1	2	GO ONLINE cmd- status error
CSR	200	0	3	SET RCV ON ERR cmd- done timeout
CSR	200	1	3	SET RCV ON ERR cmd- status error
CSR	200	0	4	Perform Collision Detect Test cmd- done timeout
CSR	200	1	4	Perform Collision Detect Test cmd- status error

5.4 PRODUCT WARRANTY

Interlan warrants that the products covered hereby shall be free from defects in material and workmanship for a period of one(1) year from the date of initial shipment by Interlan. The foregoing warranty does not apply to any products which have been subject to misuse, neglect, accident, or modification. If found defective by Interlan within the terms of this warranty, Interlan's sole obligation shall be to repair or replace at Interlan's option the defective product and carry out the unexpired term of the warranty which was applicable to the defective product. All replaced products become the property of Interlan. As a condition of this warranty, customers must (1) obtain an Interlan Return Authorization Number (RAN), and shipping instructions, (2) return all products (or approved subassemblies) transportation prepaid and insured to Interlan's Chelmsford, Masachusetts facility or other specified location, and (3) include a written description of the claimed defect. If Interlan determines that the product is not defective within the terms of this warranty, Customer shall pay all costs of handling and return postage; otherwise normal transportation charges for the return to Customer shall be paid by Interlan within the United States only. This warranty outside of the United Stated excludes all costs of shipping, Customs clearance, and other relate charges. Except for the express warranties stated above, Interlan disclaims all warranties on products including all implied warranties of merchantability and fitness; and the stated express warranties are in lieu of all obligations or liabilities on the part of Interlan.

5.5 SERVICE POLICY

Should a product fail while under the terms of the warranty agreement, it will be repaired or replaced free of charge. For out-of-warranty service, repairs are charged on a time and material basis. To return a product for out-of-warranty repair:

1. Contact the factory for an Interlan Return Authorization Number (R.A.N.), shipping instructions, and a non-binding repair cost estimate.

2. Return the product (or approved subassembly) transportation prepaid and insured to Interlan's Chelmsford, MA facility (or other specified location) with the R.A.N. number marked on the outside of the packege.

3. Include a written description of the product's symptomatic problem, and the name and telephone number of a technical contact.

4. Include a purchase order for an amount equal to the estimated repair cost, and the name and telephone number of the purchasing contact.

If Interlan determines the product not to be repairable for less than the quoted estimate repair cost, Interlan will notify the purchasing contact for repair authorization before proceeding. In all cases repairs are performed and charged on a time and materials basis, and the product is returned with transportation charges prepaid and billed. Repair is performed at the factory only, typically within a 72 hour turnaround time. To avoid delay in processing the return it is absolutely necessary to return products in the manner stated here. All repairs are warranted for a period of 30 days after return to the customer.

APPENDIX A

ETHERNET NETWORK

PLANNING, INSTALLATION, AND TEST GUIDELINES

SECTION I. PLANNING

An Ethernet system provides simplicity of installation and flexibility of layout. The Ethernet system can be readily enlarged with expanding word and information processing needs. A small Ethernet system concentrated on one floor, supporting four or five work/information processors can be progressively enlarged to a final system servicing an entire multi-floor building complex with up to 1024 stations of various types and processing power. Such a system can also interact with a local main frame supported data processing system and/or, via external transmission lines, can access remote systems and terminals. Planning and installation activity will vary greatly depending on present and future system size and complexity.

Perform the following steps to provide a total, fully integrated and highly efficient Ethernet system:

* Analyze current and expected future information processing needs.
* Select equipment expected to satisfy current and future information processing needs.
* Conduct a feasibility study in cases where an Ethernet system is required to be integrated into an existing data network.
* Plan the cable route and the distribution of the equipment.
* Install the cabling and equipment.
* Purchase the Ethernet network hardware and controlling software.
* Conduct system acceptance tests.

An Ethernet solution to a specific environment can be specified by deciding how the user's current needs will be directly satisfied. System enhancements will also be specified in anticipation of business growth and possible diversification. Specific needs will vary according to the size and nature of the business and the way that business is conducted.

Implementation of an Ethernet system requires installation plans to be specified showing how and where the Ethernet cable shall be routed and where each station shall be sited. The installation of transmission medium hardware (such as cables, transceivers, and repeaters) can be performed by wiring contractors. After cable installation is completed and before system integration of stations, transceivers, and repeaters is initiated, the cable should be subjected to continuity and stress testing. The final stage of implementation is the system acceptance test phase.

SECTION II. INSTALLATION

Before proceeding with the installation of the cable network and its associated components the route must be carefully planned along with the siting of the transceivers, repeaters, and remote repeaters. Prior to drawing up the plans a detailed site inspection must take place in order that the best route is chosen. "Best route" does not necessarily imply the most convenient or the "quickest to install" route. The cable route chosen must take into consideration many factors which will be beyond the control and influence of the system planner and installer. The route must be planned so that it will comply with all Ethernet specifications listed in Table A-1.

TABLE A-1

ETHERNET CHANNEL REQUIREMENTS

* The maximum station separation on the network is 1500 meters (4920 feet).

* The minimum station separation on the network is 2.5 meters (8 feet 2 inches).

* The maximum length of the transceiver cable between any station and its associated transceiver is 50 meters (165 feet).

* The network is comprised of one or more cable segments.

* Each cable segment is made up of the combined length of one or more cable sections, and is terminated at both ends by a 50 Ohm coaxial cable terminator.

* The maximum combined length allowed for a cable segment is 500 meters (1640 feet).

* Cable segments are interconnected by repeaters and/or remote repeaters.

* Any number of repeaters of either kind can be used, but no more than two repeaters may be inline between any two stations on the network.

* Repeaters are used to extend the length of the channel and to extend the topology from 1 to 3 dimensions.

* Remote repeaters allow point-to-point connection of cable segments many hundreds of feet apart, such as between buildings.

* No more than 1000 meters (3280 feet) of total point-to-point link are allowed.

* Repeaters may be attached at any point on a cable segment as long as the 2.5 meter minimum separation distance requirement is not violated.

* Repeater pairs occupy transceiver positions on both cable segments and count towards the maximum number of stations on each segment.

* A cable segment can accomodate up to 100 station/transceiver pairs and repeater pairs.

* The maximum number of station/transceiver pairs and repeater

pairs on a network is 1024.

When installing the Ethernet transmission medium hardware you should also comply with the following requirements:

* Avoid areas where electrical noise is present.
* Avoid areas where mechanical damage is likely.
* Use the most accessible route.
* Use the route least likely to be disturbed.
* Use the shortest route.

Obviously some of these requirements will conflict; the successful installation incorporates the above mentioned guidelines , making tradeoffs when necessary.

Pre-assembled coaxial cables with metal screw-type coaxial connectors are available from Interlan in three fixed lengths:

77 feet (23.4m),
230 feet (70.2m), and
384 feet (117m).

The sizes listed above for standard coaxial cable lengths were chosen to eliminate excessive signal reflections.

The ideal coaxial cable has no joints (i.e., it is made from one cable length). This is feasible if the required cable segment can be made from a standard cable length (e.g., 23.4, 70.2, or 117 meters). If cable segments longer than 117 meters are necessary, they must be built up from a combination of cable lengths from the same manufacturer and model type. Use a combination of cable lengths in preference to a number of identical lengths when making up a cable segment.

When constructing a cable layout, be certain that the total length of each segment does not exceed 500 meters (1640 feet) and that the total length of the network does not exceed 1500 meters (4920 feet).

The minimum bend radius allowed in a coaxial Ethernet cable section is 7 inches. If the installed cable is to be exposed, it is suggested that it be secured with cable ties to prevent possible kinking by later disturbances. In addition, the metal connectors used to interconnect cable sections and the metal cable terminators must be sleeved to prevent electrical contact with ground potential structures and electrical conductors such as conduit and cable troughs.

Devices are normally attached via one of three additional set lengths of drop cable. The multipair transceiver drop cables are sized in the following lengths:

10 feet,
50 feet, and
150 feet.

If necessary, a combination of drop cable lengths may be joined,

provided that the maximum length of 165 feet is not exceeded.

The following examples illustrate three basic installation configurations. It should be possible to design a solution to any particular installation using these examples and their accompanying figures.

A) The Typical Minimal Configuration

Minimal configurations lend themselves to cluster-type installations where the stations and devices are situated within relatively close approximation of each other. This type of installation has one cable segment ranging in length from 77 to 1640 feet. It does not require the use of repeaters. The minimal configuration supports up to 100 stations.

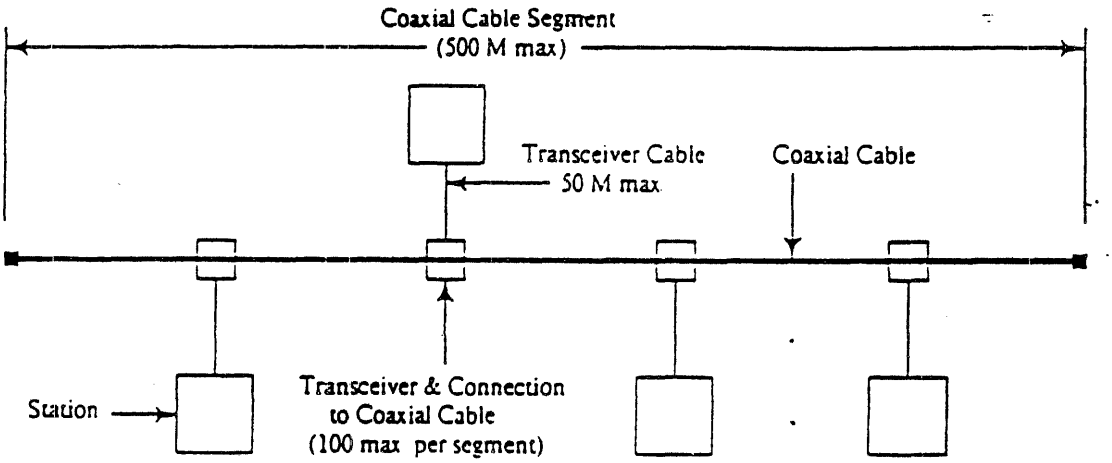


Figure A-1a: The Minimal Configuration

B) The Typical Medium-Scale Configuration

This configuration is typically used for installations that require medium distance interconnection of stations and devices within a one or two story building. A medium-scale configuration employs two cable segments, each from 77 to 1640 feet in length. The two coaxial cable segments are interconnected via a repeater. This configuration can support up to 198 stations.

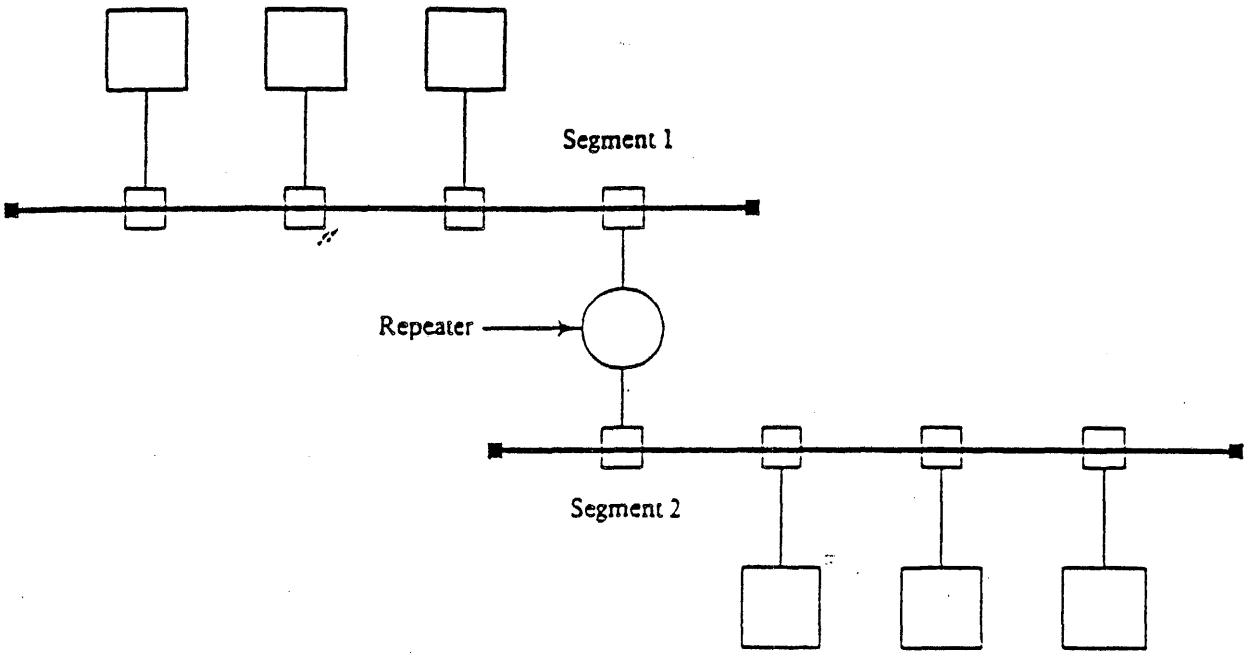


Figure A-1b. A Typical Medium-Scale Configuration

C) The Typical Large-Scale Configuration

This configuration is typically used for installations in multi-floor building complexes with interconnections via remote repeaters to adjacent building(s). This configuration can be a full scale Ethernet implementation consisting of many cable segments and, if required, several point-to-point links. Note in the figure that segment 3 acts as a central bus, insuring that the maximum of 2 repeaters between any two stations is maintained. If remote repeater(s) are used, remember that the maximum total point-to-point link distance can not exceed 1000 meters (3280 feet).

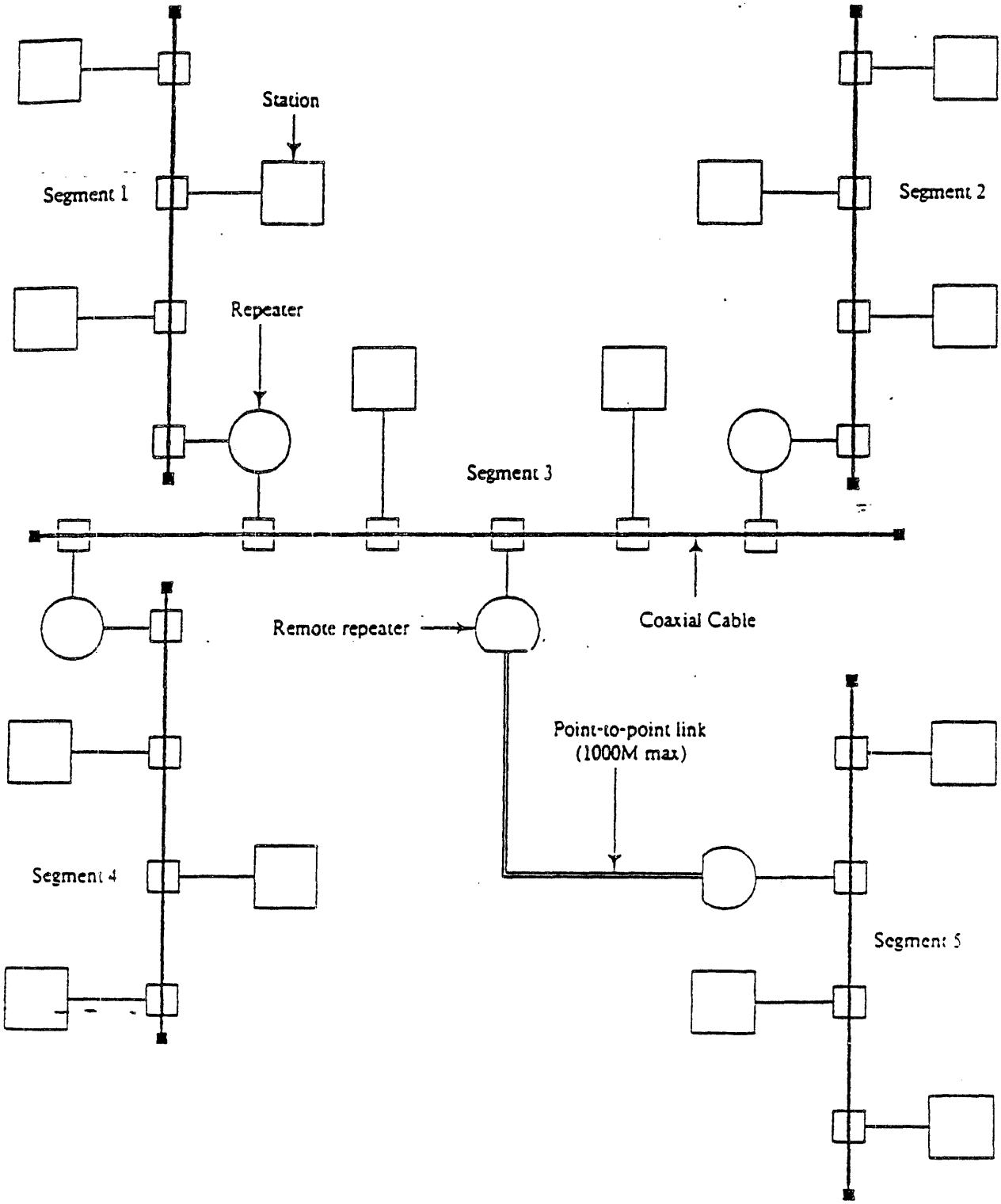


Figure A-1c: A Typical Large-Scale Configuration

SECTION III. TESTING

Testing is done in two phases. The first phase of testing is performed for each coaxial cable segment after interconnection of the individual cable lengths and attachment of the cable segment's coaxial terminators. Each cable segment should be tested individually before it is connected to other tested cable segments and before transceiver connections are made.

The second testing phase is performed to ascertain the proper operation of the network and its interconnected stations, transceivers, repeaters, and devices.

Proper testing of the coaxial cable segment is of paramount importance for successful network operation. Once proper operation of the individual cable segments is verified, testing can proceed to the total system acceptance test.

Cable testing consists of sending a half sine wave voltage pulse down the cable. Each and any cable fault will indicate itself as a point of discontinuity of one type or another and will cause energy to be reflected back down the cable to the enery source, where it is detected. This type of test is known as Time Domain Reflectometry (TDR). Open and shorted areas of cables are displayed on a chart recorder as a dramatic change in amplitude. Lesser variations can indicate frayed and crimped cables. The location of a suspected fault can be ascertained by the time delay between the incident and reflected pulses. Testing and data comparison from both ends of the cable segment provide higher resolution for fault location measurements.