

**NI3010A MULTIBUS ETHERNET  
COMMUNICATIONS CONTROLLER  
USER MANUAL**

**UM-NI3010A (950-0004-AA) (Rev. AA)**

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

---

---

## **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)**

**MICOM-INTERLAN  
DOCUMENTATION CHANGES**

---

---

**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.

REV.

REVISION HISTORY

DATE

00

Original.

11/82

## PREFACE

This manual describes how to install, program, and maintain the NI3010A MULTIBUS Ethernet Communications Controller. It contains five chapters and one appendix. They are

Chapter 1, Introduction. This chapter is a general description of the NI3010A's features, specifications, and accessories.

Chapter 2, Installation. This chapter tells you how to unpack, configure, install, and cable the NI3010A.

Chapter 3, Programming Information. This chapter tells you how to program the NI3010A. It contains complete descriptions of NI3010A data formats, all the NI3010A registers, and all the NI3010A commands. It concludes with some programming guidelines.

Chapter 4, Functional Description. This chapter describes the NI3010A's architecture. It explains how the NI3010A transmits data, receives data, and executes its onboard diagnostic programs.

Chapter 5, Maintenance. This chapter provides more information about the NI3010A's onboard diagnostic programs.

Appendix A, Ethernet Network Planning, Installation, and Test Guidelines. This chapter will help you plan, install, and commission an Ethernet network.

You can find more information about Ethernet in

The Ethernet, a Local Area Network Data Link and Physical Layer Specification, Version 1.0; September 30, 1980, Xerox/Intel/Digital.

Interlan would like to hear any comments, corrections, or suggestions that you might have about this manual. Send correspondence to:

Interlan, Inc.  
160 Turnpike Road  
Chelmsford, Massachusetts 01824  
  
(617) 256-5888      Telex 95-1909

## CONVENTIONS

A slash following a signal name indicates that the signal is asserted low (asserted at 0 volts). An example is RBA/. This is a bit in the interrupt status register. This bit is 0 (RBA/ is true) when there is a receive block available.

Numbers followed by an H are in hexadecimal. An example is 1AH. Otherwise, numbers are in decimal.

## TABLE OF CONTENTS

### LIST OF FIGURES

### LIST OF TABLES

#### Chapter 1 -- Introduction

- 1.1 NI3010A Features
- 1.2 NI3010A Specifications
  - 1.2.1 Network Specifications
  - 1.2.2 MULTIBUS Specifications
  - 1.2.3 Transceiver Specifications
  - 1.2.4 Environmental Specifications
- 1.3 NI3010A Products and Accessories

#### Chapter 2 -- Installation

- 2.1 Unpacking and Inspection
- 2.2 NI3010A ETHERNET Considerations
  - 2.2.1 Transceiver Placement
  - 2.2.2 NI3010A Transceiver Interface and Cable Requirements
  - 2.2.3 NI3010A Ethernet Addresses
    - 2.2.3.1 NI3010A Physical Address
    - 2.2.3.2 Multicast Addresses
- 2.3 Configuring the NI3010A for the MULTIBUS
  - 2.3.1 NI3010A Factory Configuration
  - 2.3.2 Setting the I/O Port Base Address
  - 2.3.3 Selecting Serial or Parallel Priority
  - 2.3.4 Selecting the MULTIBUS Address Width
  - 2.3.5 Selecting the Interrupt Request Line
- 2.4 MULTIBUS Installation
  - 2.4.1 Power Requirements
  - 2.4.2 Environmental Requirements
- 2.5 NI3010A Installation Check List

#### Chapter 3 -- Programming Information

- 3.1 Overview

## TABLE OF CONTENTS

- 3.2 Data Formats
  - 3.2.1 Transmit Data Format in MULTIBUS Memory
  - 3.2.2 Receive Data Format in MULTIBUS Memory
- 3.3 The NI3010A Internal Registers
  - 3.3.1 The Command Register
  - 3.3.2 The Status Register
  - 3.3.3 The Transmit Data Register
  - 3.3.4 The Receive Data Register
  - 3.3.5 The Interrupt Status Register
  - 3.3.6 The Interrupt Enable Register
  - 3.3.7 The Extended Bus Address Register
  - 3.3.8 The High Bus Address Register
  - 3.3.9 The Low Bus Address Register
  - 3.3.10 The High Byte Count Register
  - 3.3.11 The Low Byte Count Register
- 3.4 NI3010A Command Descriptions
  - 3.4.1 Reserved (00H)
  - 3.4.2 Set Module Interface Loopback (01H)
  - 3.4.3 Set Internal Loopback (02H)
  - 3.4.4 Clear Loopback (03H)
  - 3.4.5 Set Promiscuous Mode (04H)
  - 3.4.6 Clear Promiscuous Mode (05H)
  - 3.4.7 Set Receive-on-error Mode (06H)
  - 3.4.8 Clear Receive-on-error Mode (07H)
  - 3.4.9 Go Offline (08H)
  - 3.4.10 Go Online (09H)
  - 3.4.11 Run Onboard Diagnostics (0AH)
  - 3.4.12 Perform Network Loopback Test (0BH)
  - 3.4.13 Perform Collision Detect Test (0CH)
  - 3.4.14 Reserved (0DH - 17H)
  - 3.4.15 Report and Reset Statistics (18H)
  - 3.4.16 Report Collision Delay Times (19H)
  - 3.4.17 Reserved (1AH - 27H)
  - 3.4.18 Load Transmit Data (28H)
  - 3.4.19 Load Transmit Data and Send (29H)
  - 3.4.20 Load Group Address(es) (2AH)
  - 3.4.21 Delete Group Address(es) (2BH)
  - 3.4.22 Reserved (2CH - 3EH)
  - 3.4.23 Reset (3FH)
- 3.5 Programming Guidelines
  - 3.5.1 How to Manage NI3010A Interrupts
  - 3.5.2 When not to Write the Bus Address and Byte Count Registers
  - 3.5.3 When You Must Read the Status Register
  - 3.5.4 Handling NI3010A Interrupts
  - 3.5.5 Issuing an NI3010A Command

## TABLE OF CONTENTS

- 3.5.6 Transmitting Data to the Ethernet
- 3.5.7 Receiving a Status Block from the NI3010A

### Chapter 4 -- Functional Description

- 4.1 NI3010A Architecture
  - 4.1.1 Physical Description
  - 4.1.2 Functional Overview
- 4.2 The MULTIBUS Interface Board
  - 4.2.1 The MULTIBUS 'SLAVE' Interface
  - 4.2.2 The DMA Controller
    - 4.2.2.1 The Interrupt Enable Register
    - 4.2.2.2 The Bus Address Registers
    - 4.2.2.3 The Byte Count Registers
  - 4.2.4 The State Machine
- 4.3 The NM10A Ethernet Protocol Module
  - 4.3.1 The NM10A Transmit and Receive Buffers
  - 4.3.2 The NM10A Registers
    - 4.3.2.1 The Command Register
    - 4.3.2.2 The Status Register
    - 4.3.2.3 The Transmit Data Register
    - 4.3.2.4 The Receive Data Register
    - 4.3.2.5 The Interrupt Status Register
- 4.4 The Ethernet Frame Format
  - 4.4.1 The Destination Address Field
  - 4.4.2 The Source Address Field
  - 4.4.3 The Type Field
  - 4.4.4 The Data Field
  - 4.4.5 The Frame Check Sequence Field
- 4.5 The NI3010A Transmit Process
  - 4.5.1 The Transmit Data DMA Transfer
  - 4.5.2 Ethernet Frame Assembly
  - 4.5.3 Channel Access
- 4.6 The NI3010A Receive Process
  - 4.6.1 Checking the Destination Address
  - 4.6.2 Collision Fragment Filtering
  - 4.6.3 Cyclic Redundancy Check
  - 4.6.4 Receive Buffer Management
  - 4.6.5 The Receive Data DMA Transfer

## TABLE OF CONTENTS

4.6.6	Buffer Chaining
4.7	Diagnostic Tools
4.7.1	Module Interface Loopback
4.7.2	Internal Loopback
4.7.3	External Loopback
4.7.4	The NM10A Diagnostic Self-Test
Chapter 5 --	Maintenance
5.1	Verifying Proper Operation
5.2	NM10A On-Board Diagnostics
5.2.1	Run On-Board Diagnostics (OAH) Command Operation
5.2.2	Power-up or Reset Operation
5.3	Suggestions for Verifying Further NI3010A Operations
5.4	Product Warranty
5.5	Service Policy
APPENDIX A --	Ethernet Network Planning, Installation, and Test Guidelines

TABLE OF CONTENTS  
LIST OF FIGURES

LIST OF FIGURES

Chapter 1		
	Figure 1-1	Ethernet Architecture and NI3010A Implementation
Chapter 2		
	Figure 2-1	NI3010A Transceiver Cable Connector (NI3010A end)
	Figure 2-2	AC-NM10-10 Transceiver Cable Connector (transceiver end)
	Figure 2-3	The Location of the NI3010A's DIP Switch and Wire Jumpers
Chapter 3		
	Figure 3-1	Transmit Data Block in MULTIBUS Memory
	Figure 3-2	Receive Data Block in MULTIBUS Memory
	Figure 3-3	The Frame Status Byte
	Figure 3-4	The Statistics Status Block
	Figure 3-5	The Collision Delay Times Status Block
	Figure 3-6	Multicast Address Format
Chapter 4		
	Figure 4-1	Functional Block Diagram of the MIB
	Figure 4-2	Functional Block Diagram of the NM10A
	Figure 4-3	Ethernet Frame Format
	Figure 4-4	Functional Diagram of the NI3010A Transmit DMA Operation
	Figure 4-5	Functional Diagram of the NI3010A Receive DMA Operation

TABLE OF CONTENTS  
LIST OF TABLES

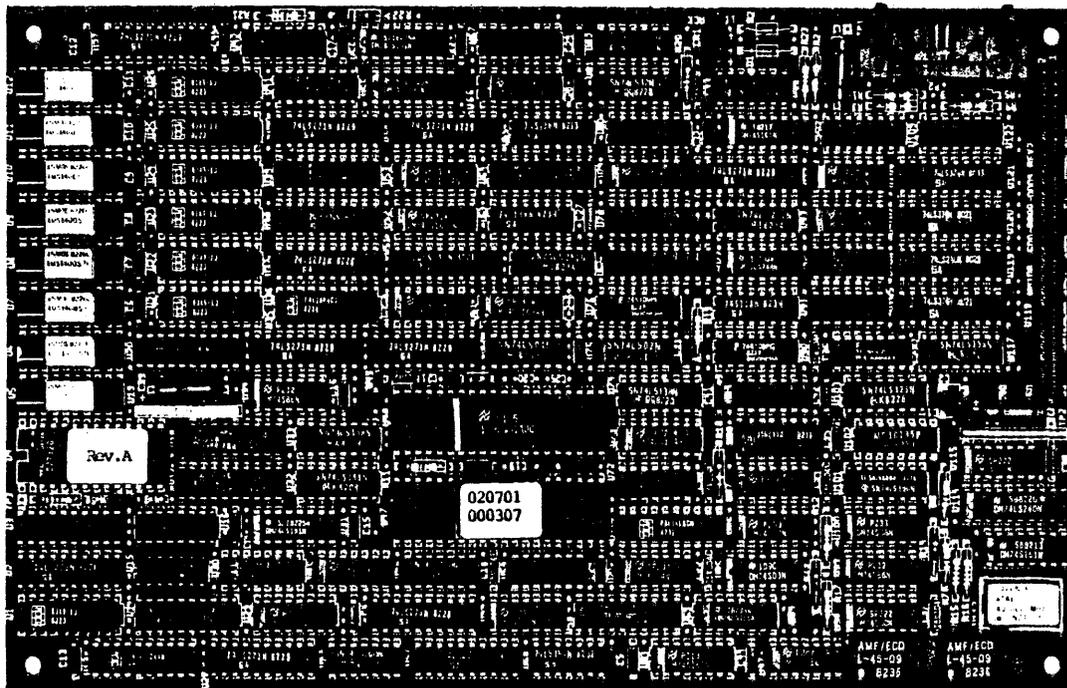
LIST OF TABLES

Chapter 3

Table 3-1	The NI3010A Commands
Table 3-2	The Status Codes
Table 3-3	The Diagnostic Status Codes
Table 3-4	Explanation of the Frame Status Byte

Chapter 4

Table 4-1	The DMA Registers
Table 4-2	The NM10A Registers



REV. A

020701

000307

AMF/ECO

1-45-09

8239

Rev. A

020701  
000307

AMF/ECO  
1-45-09  
8239

AMF/ECO  
1-45-09  
8239

INTRODUCTION  
 General Description

CHAPTER ONE

INTRODUCTION

Interlan's NI3010A is a MULTIBUS Ethernet communications controller. It's a single board that along with a transceiver provides a host MULTIBUS system with a complete connection to an Ethernet network.

The NI3010A incorporates Interlan's NM10A Ethernet protocol module and complies in full with the Xerox/Intel/Digital Ethernet Specification, Version 1.0. It performs the specified data link and physical channel functions.

An Ethernet network is a local area baseband network. Data travel over distances of up to 2500 meters at 10 Megabits/second.

Figure 1-1 illustrates the Ethernet architecture and its NI3010A implementation.

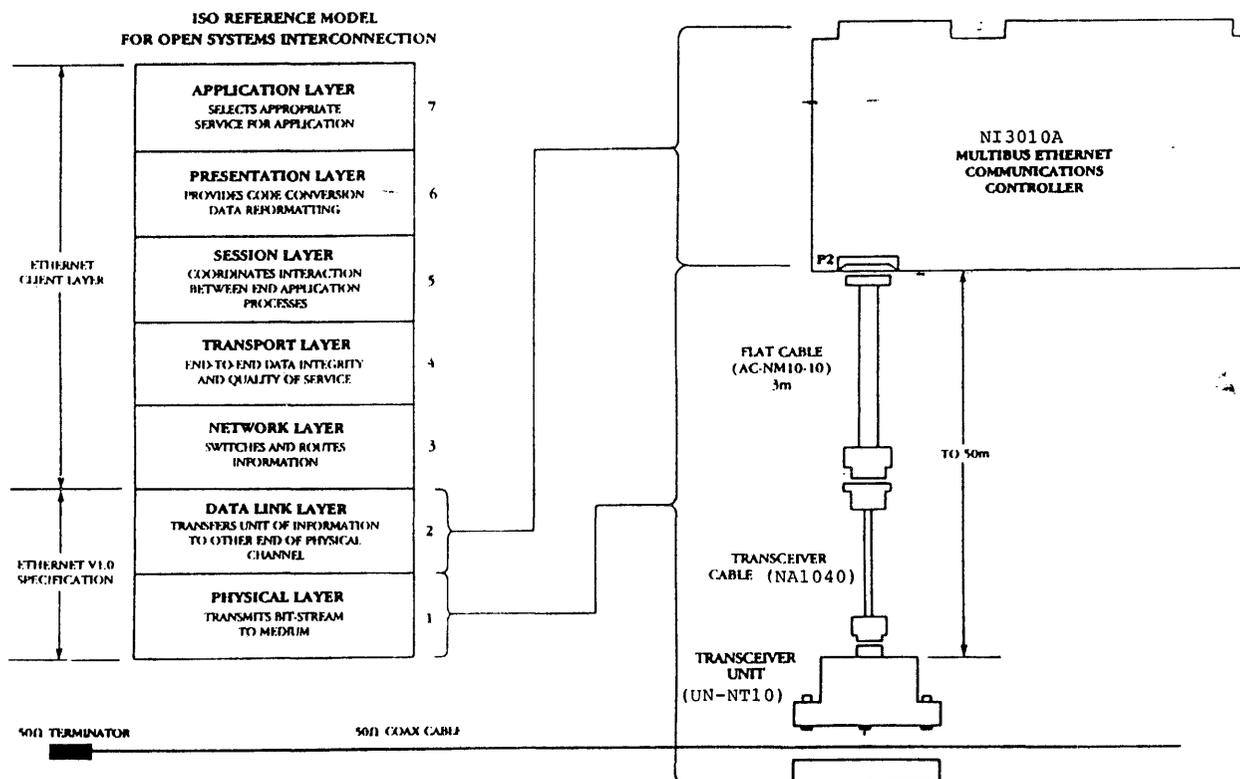


Figure 1-1. Ethernet Architecture and NI3010A Implementation

## 1.1 NI3010A FEATURES

The NI3010A has a number of important features. The NI3010A

- \* Complies fully with the Xerox/Intel/Digital Ethernet Specification, Version 1.0
- \* Performs Ethernet data link functions

The NI3010A formats your data into Ethernet frames and performs CSMA/CD (carrier sense, multiple access with collision detect).

When not transmitting a frame, the NI3010A listens to the network. The NI3010A accepts a frame if the frame's destination address matches the NI3010A's physical address, one of the NI3010A's multicast addresses, or the broadcast address.

The NI3010A performs CRC generation and CRC checking and tests receive frames for alignment errors. You can choose to have the NI3010A accept frames with CRC or alignment errors.

- \* Performs Ethernet physical channel functions.

The NI3010A transmits and receives bit streams at 10 Megabits/second. Its electrical and timing specifications are compatible with an Ethernet transceiver.

The NI3010A performs the required frame synchronization and Manchester encoding/decoding.

- \* Maintains high station performance while minimizing the service load placed on the host MULTIBUS system.

The NI3010A contains a 13.5 Kbyte receive FIFO buffer. Ethernet frames characteristically arrive with unpredictable arrival times. The NI3010A stores receive frames in its receive buffer. The host can read this buffer at its own convenience.

The NI3010A performs DMA transfers to and from host memory.

- \* Provides extensive diagnostic capability.

After a hardware reset, the NI3010A performs a set of power-up tests. A pass/fail LED indicates the success or failure of these tests. If a confidence test fails, a diagnostic status code provides further information.

The NI3010A has three different types of loopback: module interface loopback, internal loopback, and external

loopback.

- \* Collects and reports network statistics.

The NI3010A collects information on network traffic and errors. You can read this information under program control.

## 1.2 NI3010A SPECIFICATIONS

This section contains network, MULTIBUS, transceiver, and environmental specifications.

### 1.2.1 Network Specifications

Data transmission rate	10 Megabits/second
Maximum coaxial cable segment length	500 meters (1640 feet)
Maximum coaxial cable length between two stations	1500 meters (4920 feet)
Maximum length of point-to-point links	1000 meters (3280 feet)
Maximum station separation	2500 meters (1.55 miles)
Maximum distance between the NI3010A and its transceiver	50 meters (165 feet)
Maximum number of transceivers on a coaxial cable segment	100
Maximum number of repeaters between any two stations	2
Maximum number of stations on a network	1024

### 1.2.2 MULTIBUS Specifications

Power	The NI3010A requires +5Vdc +- 5% @ 7.0 A typical, 8.0 A maximum.  The NI3010A transceiver requires +12 Vdc +- 5% @ 0.5 A maximum.
Registers	The NI3010A has 11 registers. Each register has its own I/O port address. This I/O port address is the NI3010A's I/O port base address plus an offset.  A DIP switch determines the NI3010A's I/O port base address.
MULTIBUS interrupt line	Selectable from INTO/ to INT7/. The NI3010A is shipped from the factory with INT1/ selected.
MULTIBUS data transfers	Host processor-requested direct memory access.

### 1.2.3 Transceiver Specifications

Compatibility	All transceiver signals are compatible with the Ethernet Specification, Version 1.0.  The transceiver connects to the NM10A's P2 connector.
Mating Connectors	16-pin Berg 65846-01, 3M 3452, or equivalent.

### 1.2.4 Environmental Specifications

Operating temperature	0 to 50 degrees Celsius (32 to 122 Fahrenheit)
Relative humidity	Maximum of 90%, non-condensing.

### 1.3 NI3010A PRODUCTS AND ACCESSORIES

Model number	Description
BD-NI3010A	NI3010A MULTIBUS Ethernet communications controller.
UM-NI3010A	NI3010A user manual
AC-NM10-10	Flat cable with connectors. Connects the NI3010A to its transceiver or to the transceiver cable. Its length is 10 feet (3 meters).
AC-NA1010-xxx	Transceiver cable. Extends between the flat cable and the Ethernet transceiver. Available in lengths of 10, 50, and 150 feet.
NA1020-xx	Ethernet 50 ohm coaxial cable. Available in lengths of 77, 230, and 385 feet.

## CHAPTER TWO

### INSTALLATION

The NI3010A is a single board assembly that is mechanically, electrically, and architecturally compatible with Intel's standards for MULTIBUS compatibility. You can install the NI3010A in any system that uses the MULTIBUS architecture.

This chapter tells you how to configure the NI3010A, install it in a MULTIBUS system, and cable it to the Ethernet.

#### 2.1 UNPACKING AND INSPECTION

To protect against damage during shipment, INTERLAN packages each NI3010A in a special carton.

Open the carton at either end and remove the contents.

Carefully inspect the NI3010A board and other enclosed material for any visible sign of damage. If you detect any damage, immediately notify Interlan Customer Service and the carrier responsible for shipment.

Save all shipping cartons and packing material in case you need to reship the product.

#### 2.2 NI3010A ETHERNET CONSIDERATIONS

The NI3010A is completely compatible with the Xerox/Intel/Digital Ethernet Specification, Version 1.0. It may be connected to any Ethernet running at 10 Megabits/second. Appendix A will help you install and test 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 ensure that reflections from transceivers do not cause transmission errors, you must control the placement of transceivers along the cable.

Approved Ethernet coaxial cable is marked with annular rings at 2.5 meter intervals. By only placing a transceiver at one of these rings, you minimize 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 NI3010A Transceiver Interface and Cable Requirements

The NI3010A connects to the Ethernet via the board's P2 flat cable connector. Figure 2-1 shows the connector's pinout.

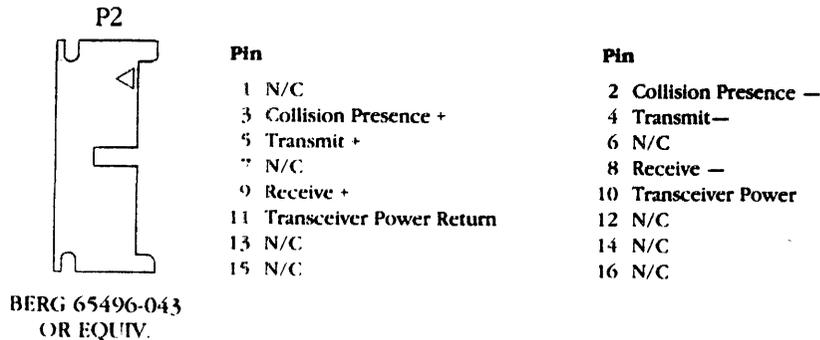


Figure 2-1. NI3010A Transceiver Cable Connector (NI3010A End)

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

The NI3010A connects to the transceiver through either one or two cables. You can connect the NI3010A directly to the transceiver with the AC-NM10-10 cable; or you can mate the AC-NM10-10 cable with the AC-NA1010 cable, which then connects to the transceiver. The total length of cable between the NI3010A and its transceiver must not exceed 50 meters (165 feet).

The AC-NM10-10 cable is a flat cable intended for flexible interconnection in an internal cabinet environment. Its length must not exceed 3 meters (10 feet).

The AC-NA1010 is a twisted-pair round cable. It has less loss than an equivalent length of flat cable.

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). Figure 2-2 shows the pin assignments for this connector. These pin assignments obey the Ethernet Specification.

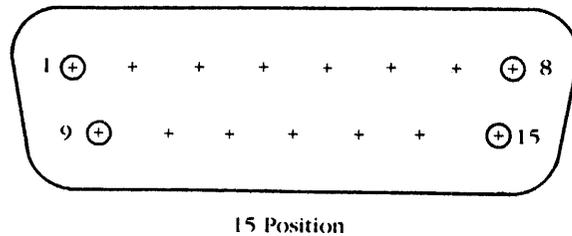


Figure 2-2. AC-NM10-10 Transceiver Cable Connector (Transceiver End)

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 NI3010A's logic ground, only to the frame of the cabinet.

### 2.2.3 NI3010A Ethernet Addresses

Interlan has assigned each NI3010A a unique physical address. You may assign the NI3010A up to 63 different multicast addresses.

#### 2.2.3.1 NI3010A Physical Address

Each NI3010A has a unique 48-bit physical address. Interlan selected this address from within a contiguous block of Ethernet physical addresses obtained from Xerox Corporation through their Ethernet

licensing arrangement. The NI3010A's physical address is distinct from the physical address of any other station on any Ethernet.

The NI3010A's physical address resides in ROM. You cannot alter it. As a convenience, the board wears a sticker displaying its physical address. Should the sticker be missing, you can read the NI3010A's physical address under program control. To find out how to do this, read the description of the Report and Reset Statistics command (18H) in Chapter 3.

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
end of physical address block:	02	07	01	FF	FF	FF
(hexadecimal)						
				-----	-----	
			24 bits assigned by...	Xerox	Interlan	

### 2.2.3.2 Multicast Addresses

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

- \* Multicast-group addresses. These are addresses associated by higher level convention with a group of logically related stations.
- \* The broadcast address. A predefined multicast address (a destination address of all ones) that specifies the set of all stations on a given Ethernet.

The NI3010A recognizes the broadcast address and up to 63 user-assigned multicast addresses. See Chapter 3 for a complete description of how to use the NI3010A's multicast address capability.

## 2.3 CONFIGURING THE NI3010A FOR THE MULTIBUS

Configuring the NI3010A for the MULTIBUS means

- \* Setting the I/O port base address
- \* Choosing serial or parallel priority
- \* Choosing whether to use the 4 most significant bits of the MULTIBUS address
- \* Choosing the MULTIBUS interrupt request line used by the NI3010A

### 2.3.1 NI3010A Factory Configuration

Each NI3010A has the following configuration when shipped from the factory.

The I/O port base address is zero.

Serial priority is selected.

The four most significant bits of the MULTIBUS address are enabled.

The NI3010A interrupts on INT1/.

### 2.3.2 Setting the I/O Port Base Address

You set the I/O port base address with a DIP switch located at U1. The I/O port address is 8 bits. The four positions of this DIP switch determine the 4 most significant bits of the I/O port base address. Position 1 is the most significant bit. The bit is 1 if the switch is on. It's 0 if the switch is off.

Figure 2-3 shows the location of this DIP switch.

### 2.3.3 Choosing Serial or Parallel Priority

A wire jumper (R9) determines whether the NI3010A obeys serial or parallel priority. When the wire jumper is in, the NI3010A obeys serial priority. When the wire jumper is out, the NI3010A obeys parallel priority. If you choose serial priority, be sure that you have priority going to the NI3010A's MULTIBUS slot. Figure 2-3 shows the location of R9.

### 2.3.4 Determining the MULTIBUS Address Width

Wire jumpers (R10 - R13) determine if the upper 4 MULTIBUS address lines are enabled. One wire jumper controls each address line. When the wire jumper is in, the address line is enabled. When the wire jumper is out, the address line is disabled.

R13 controls the most significant address bit, A17H. R10 controls the least significant of the upper 4 bits, A14H. Figure 2-3 shows the locations of these wire jumpers.

### 2.3.5 Choosing the MULTIBUS Interrupt Request Line

A wire jumper determines what MULTIBUS interrupt request line the NI3010A uses. There are eight positions on the NI3010A labeled INTO through INT7. The NI3010A uses the interrupt request line whose position has the jumper. The NI3010A comes with the jumper in the INT1 position.

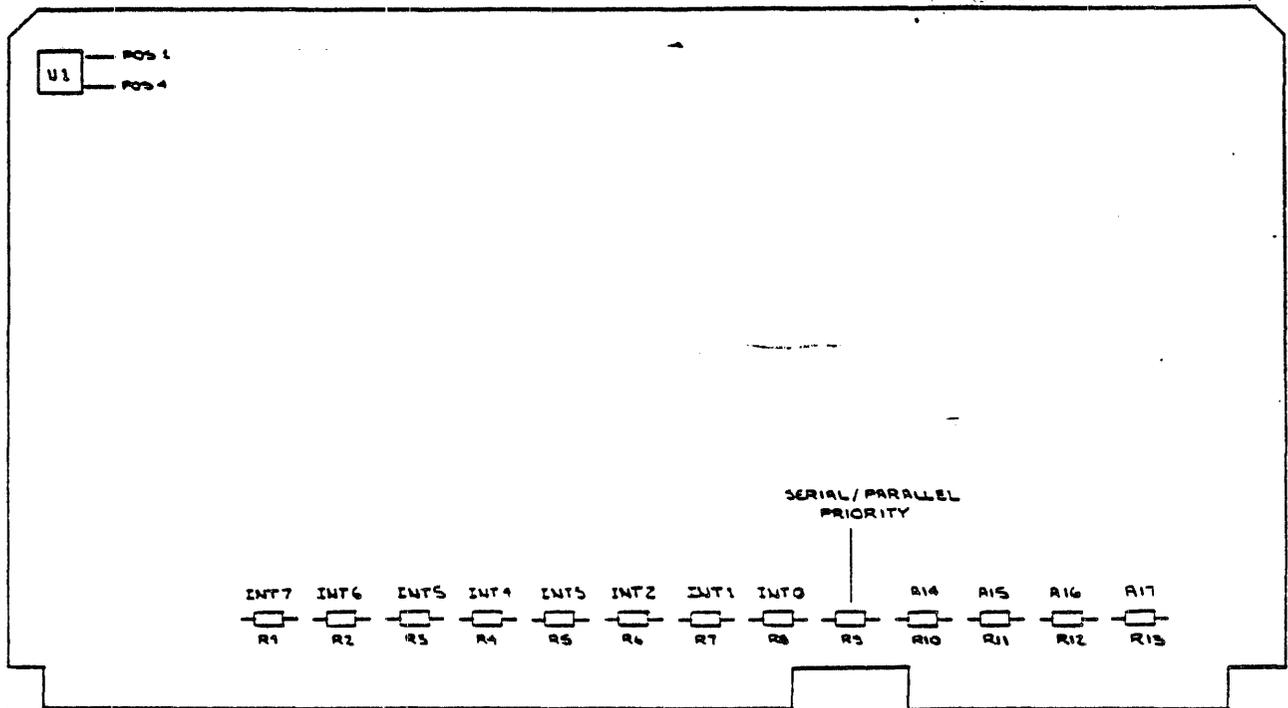


Figure 2-3.  
The Location the NI3010A's DIP Switch and Wire Jumpers

## 2.4 MULTIBUS INSTALLATION

### 2.4.1 Power Requirements

The NI3010A and its Ethernet transceiver are powered directly from the MULTIBUS backplane and have the following power requirements.

+4.75 to +5.25 Vdc @ 7.0 A typical, 8.0 A maximum  
(NI3010A board only)

+11.4 to +12.6 Vdc @ 0.5 A maximum (transceiver unit only)

The NI3010A provides power to its Ethernet transceiver unit from the +12Vdc power on the MULTIBUS backplane.

CAUTION. Before installing the NI3010A, verify that your power supply can satisfy the current requirements of the NI3010A and its transceiver. After installation, check that your power supply's +5Vdc and +12Vdc voltages are still within the required levels.

### 2.4.2 Environmental Requirements

Interlan has designed the NI3010A to operate in a standard MULTIBUS slot with forced air cooling and an inlet air temperature of less than 50 degrees Celsius. If you operate the NI3010A in free air (that is, without forced air flow across its surface) the room ambient temperature must be below 25 degrees Celsius and the NI3010A must be placed away from objects that impede convective air flow.

CAUTION. Under no circumstances should you operate the NI3010A in a backplane without forced air flow. Serious damage to the board can result.

You may operate the NI3010A in an environment with relative humidity up to 90%, provided that moisture does not condense on the board.

2.5 NI3010A INSTALLATION CHECK LIST

1. Select I/O port base address \_\_\_\_\_
2. Select serial or parallel priority \_\_\_\_\_
3. Select MULTIBUS address width \_\_\_\_\_
4. Select MULTIBUS interrupt request line \_\_\_\_\_
5. Select a MULTIBUS slot with adequate forced air ventilation \_\_\_\_\_
6. Check your power supply for adequate capacity \_\_\_\_\_
7. With the system power off, place the NI3010A in the selected MULTIBUS slot \_\_\_\_\_
8. Connect the AC-NM10 flat cable to the NI3010A's P2 connector \_\_\_\_\_
9. Connect the flat cable (AC-NM10-10) to the Ethernet transceiver, either directly or via the transceiver cable (AC-NA1010 or equivalent) \_\_\_\_\_
10. Connect the Ethernet transceiver to the Ethernet coax cable \_\_\_\_\_

You are now ready to power up the NI3010A.

## CHAPTER THREE

### PROGRAMMING INFORMATION

#### 3.1 OVERVIEW

This chapter tells you how to program the NI3010A MULTIBUS Ethernet Communications Controller. It tells you

- \* What the format is for transmit blocks, receive blocks, and statistical blocks
- \* What the NI3010A registers are and how to access them
- \* What all the NI3010A commands do and how to issue them
- \* What functions your host programs might perform

The NI3010A is a DMA (direct memory access) device that responds to commands issued by the host MULTIBUS system.

The host issues a command to the NI3010A by writing the lower 6 bits of the NI3010A's I/O command register. The command register is located at the NI3010A's port base address, which is determined by the setting of an onboard DIP switch.

The NI3010A carries out the command and then loads its status register. The host reads the NI3010A's status register to find out if the NI3010A carried out the command successfully. The status register is at the NI3010A's I/O port base address + 1.

Table 3-1 lists the NI3010A commands, and Tables 3-2 and 3-3 list the NI3010A status codes.

Table 3-1

NI3010A COMMAND FUNCTION CODES

Code	Command	Possible Status Codes
00H	Reserved	02H
01H	Set Module Interface Loopback	00H
02H	Set Internal Loopback	00H
03H	Clear Loopback	00H
04H	Set Promiscuous Mode	00H
05H	Clear Promiscuous Mode	00H
06H	Set Receive-on-Error Mode	00H
07H	Clear Receive-on-Error Mode	00H
08H	Go Offline	00H
09H	Go Online	00H
0AH	Run Onboard Diagnostics	Diagnostic Status Code
0BH-0CH	Reserved- DO NOT USE	00H
0DH	Set Insert Source Address Mode	00H
0EH	Clear Insert Source Address Mode	00H
0FH	Set Physical Address to Default	00H
10H	Set Receive All Multicast Packets	00H
11H	Clear Receive All Multicast Packets	00H
12H	Perform Network Loopback Test	00H,01H,03H,04H,05H,0CH,0DH,0EH
13H	Perform Collision Detect Test	00H,03H,04H,05H,08H
14H- 17H	Reserved	02H
18H	Report and Reset Statistics	00H
19H	Report Collision Delay Times	00H
1AH- 27H	Reserved	00H
28H	Load Transmit Data	00H, 05H
29H	Load Transmit Data and Send	00H,01H,03H,04H,05H,06H,08H
2AH	Load Group Address(es)	00H, 05H, 0AH,
2BH	Delete Group Address(es)	00H, 05H, 0AH
2CH	Load Physical Address	00H,0AH
2DH- 3EH	Reserved	02H
3FH	Reset	Diagnostic Status Code

Table 3-2  
 NI3010A COMMAND STATUS CODES

Code	Status
00H	Success
01H	Success with Retries
02H	Illegal Command
03H	Inappropriate Command
04H	Failure
05H	Buffer Size Exceeded
06H	Frame too Small
07H	Reserved
08H	Excessive Collisions
09H	Reserved
0AH	Buffer Alignment Error
0BH	No Heartbeat Detected
0CH	No CRC Error Occurred
0DH	Inappropriate CRC Error
0EH	Last Data Byte Not Received Correctly
0FH- FFH	Reserved

Table 3-3  
 NI3010 DIAGNOSTIC STATUS CODES

Code	Status
00H	Success
01H	NM10A ROM Error
02H	NM10A RAM error
03H	Address Error
04H	Loopback Error

### 3.2 DATA FORMATS

This section describes the format of transmit and receive blocks in host memory.

#### 3.2.1 Transmit Data Block in MULTIBUS memory

The host transfers Ethernet data to the NI3010A by setting up a transmit block in its own memory, writing the NI3010A's bus address registers (BAR) with the block's starting address, writing the NI3010A's byte count registers (BCR) with the block's byte count, and then initiating a transmit DMA. The host must set up that transmit block in a particular format. Figure 3-1 shows that format.

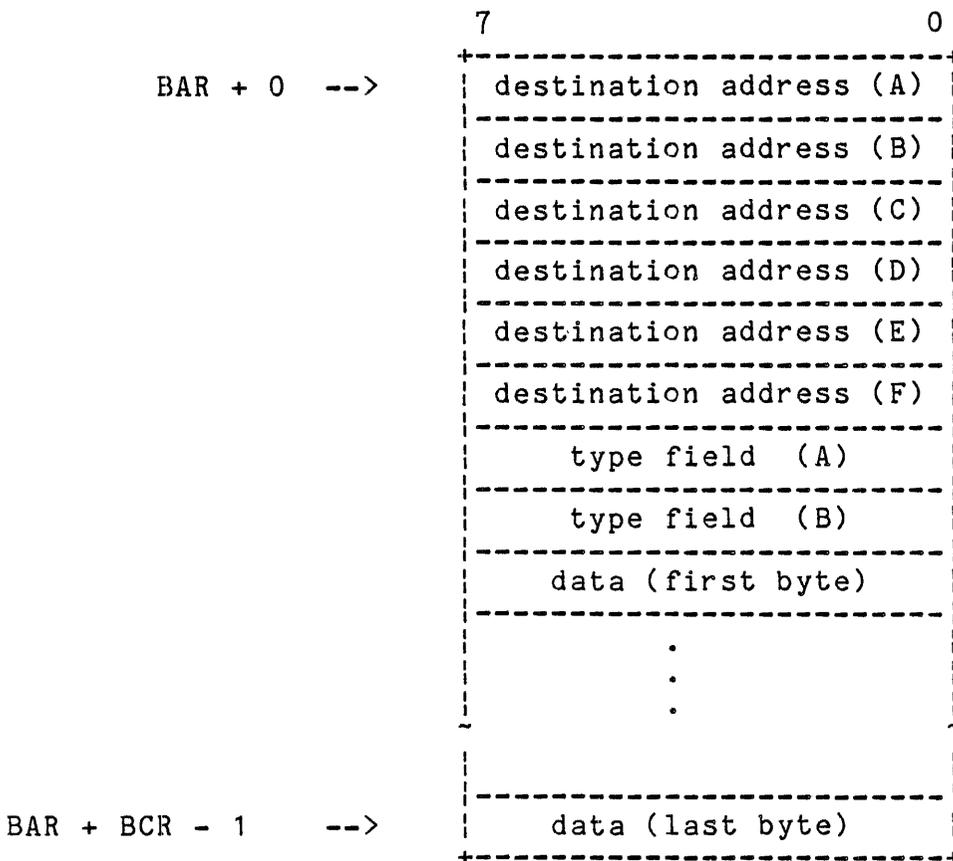


Figure 3-1

Transmit Data Block in MULTIBUS Memory.

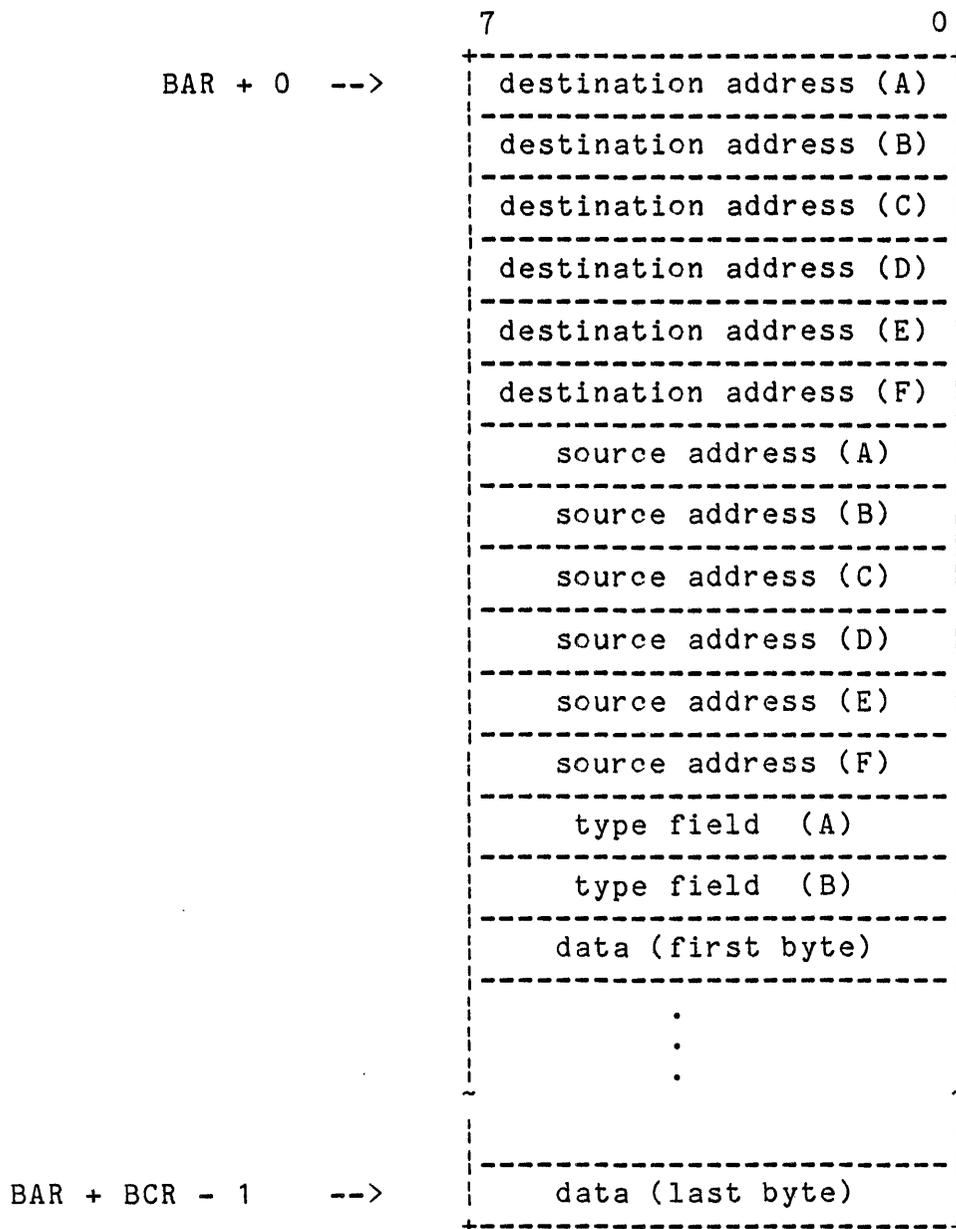


Figure 3-1a

Transmit Data Block in Memory  
(no source address insertion mode)

The minimum length of the data field in an Ethernet frame is 46 bytes. If you supply less than this number, the NI3010A pads the data field with null characters to make 46 data bytes. A null character has all its bits equal to 0.

If you supply a transmit frame of less than 8 bytes (no data field and incomplete destination and type fields), the Load Transmit Data command (28H) or the Load Transmit Data and Send command (29H) returns a status code of Frame too Small (06H).

The maximum length of the data field in an Ethernet frame is 1500 bytes. If you supply a transmit frame of more than 1508 bytes (1500 data bytes plus 8 bytes for the destination and type fields), a Load Transmit Data command (28H) or a Load Transmit Data and Send command (29H) returns a status code of Buffer Size Exceeded (05H).

Only load the NI3010A's transmit FIFO with one frame at a time. When you issue a Load Transmit Data and Send command (29H), the NI3010A transmits all the data in its transmit FIFO as an Ethernet frame. It adds the Ethernet preamble, the source address, and the CRC value.

### 3.2.2 Receive Data Format in MULTIBUS Memory

When the host accepts a receive-block-available interrupt from the NI3010A, it reserves a block in its own memory for the receive data, writes the NI3010A's bus address registers (BAR) with the block's starting address, writes the NI3010A's byte count registers (BCR) with the block's byte count, and then initiates a receive DMA.

The receive data enter host memory in a particular format. Figure 3-2 shows that format.

When the NI3010A receives an Ethernet frame, it strips off the preamble and stores the rest of the frame in its receive FIFO. The rest of the frame includes 6 bytes of destination address, 6 bytes of source address, 46 to 1500 bytes of data, and 4 bytes of CRC. The minimum receive frame is 64 bytes, and the maximum receive received frame is 1518 bytes.

When the NI3010A transfers a frame to the host, it adds 4 bytes of header. The first byte is the frame status; the second is a null byte; and the third and fourth are the frame length. The frame length is a binary value representing the number of bytes in the received frame. Figure 3-3 shows the format of the frame status, and Table 3-4 explains how to interpret it.

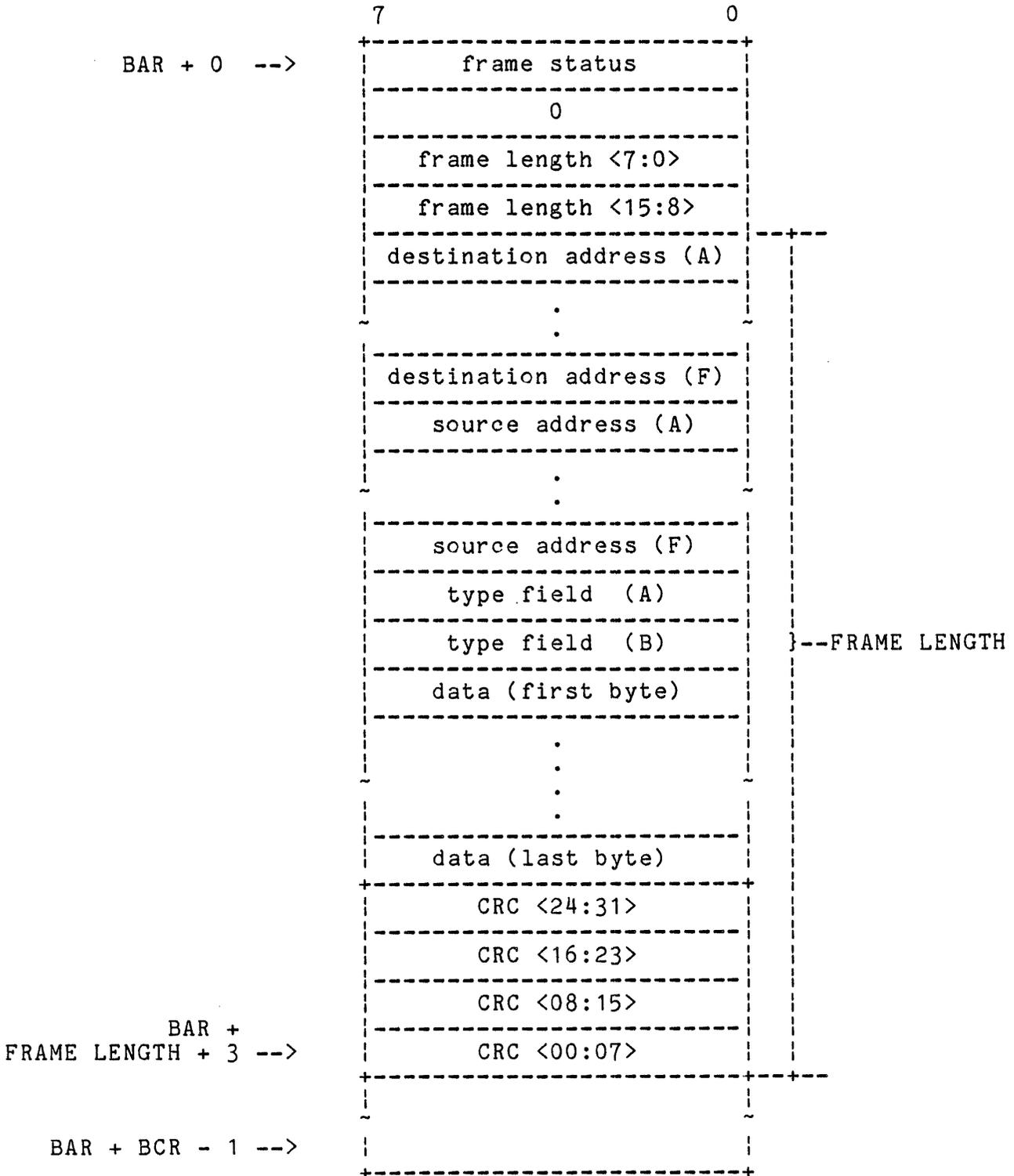


Figure 3-2

Receive Data Block in MULTIBUS Memory

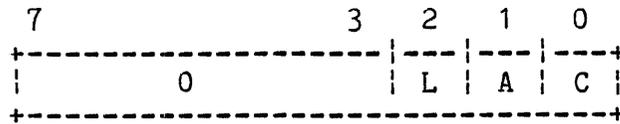


Figure 3-3  
The Frame Status Byte

Table 3-4  
Explanation of the Frame Status Byte

Bits	Name	Description
<7:3>	0	These bits are all zero.
<2>	L	Lost. When 1, indicates that the NI3010A lost one or more frames immediately prior to the reception of this frame.
<1>	A	Alignment error. When 1, indicates that this frame has an alignment error.  A frame not consisting of an integral number of bytes has an alignment error.
<0>	C	CRC error. When 1, indicates that this frame as a CRC error.

The CRC value is 32 bits (CRC<0:31>) long. The Ethernet Specification requires that an Ethernet transmitter send the most significant bit of the CRC value (CRC<31>) first. In the received frame the CRC value appears as 4 bytes. The most significant byte appears first. Bit 0 of each CRC byte contains the most significant bit of that byte, and bit 7 contains the least significant bit.

When the host initiates a receive DMA, it receives the oldest receive frame stored in the NI3010A's receive FIFO. After receiving that frame, the host once again enables a receive-block-available interrupt. If the NI3010A has another frame ready, it interrupts again.

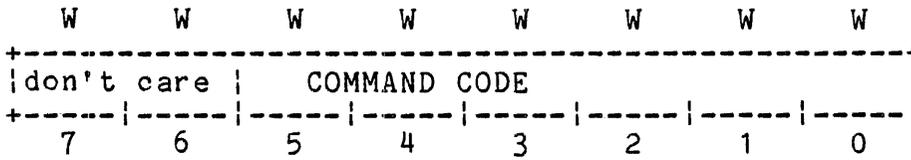
### 3.3 THE NI3010A INTERNAL REGISTERS

The NI3010A internal registers are:

the command register	(C_REG)
the status register	(S_REG)
the transmit data register	(T_DATA)
the receive data register	(R_DATA)
the interrupt status register	(IS_REG)
the interrupt enable register	(IE_REG)
the bus address register	(E_BAR, H_BAR, L_BAR)
the byte count register	(H_BCR, L_BCR)

The host accesses an NI3010A internal register by addressing one of the I/O ports assigned to the NI3010A.

3.3.1 The Command Register



Bits	Name	Description
<7:6>	don't care	The NI3010A ignores these bits.
<5:0>	command code	These bits contain one of the command codes listed in Table 3-1.

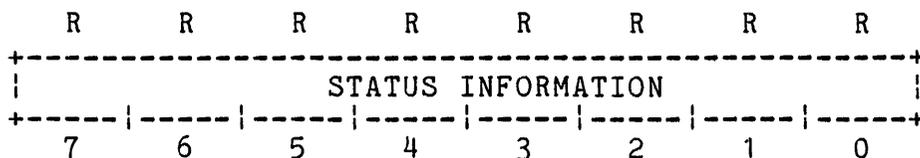
Access: Write only

I/O Port Address: The I/O port base address

Function: When the host writes this register, the NI3010A carries out the command. The host must not issue another command until it reads all bytes from the status register.

The NI3010A will not carry out the command in the command register unless the status register is empty.

3.3.2 The Status Register



Bits	Name	Description
<7:0>	status information	<p>These bits contain one of the status codes listed in Table 3-2 or one of the diagnostic status codes listed in Table 3-3.</p> <p>The NI3010A returns a diagnostic status code in response to the Run Onboard Diagnostics command (0AH) or the Reset command (3FH).</p>

Access: Read only

I/O Port Address: I/O port base address + 01H

Function: The host reads this register to determine the success of the command it just issued.

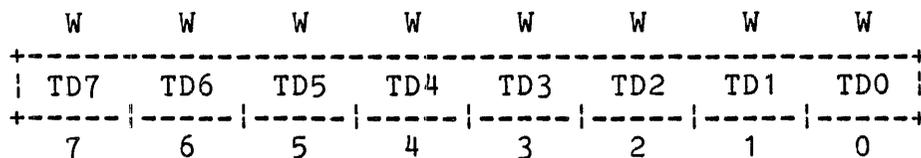
After issuing its command, the host may poll the interrupt status register to determine if the status register is ready to be read.

Alternatively, the host may have enabled a status-register-full interrupt before issuing its command and wait for the interrupt.

The NI3010A will not carry out the command in the command register unless the status register is empty (SBA/ not asserted).

The NI3010A will not refill the Receive Data Register after one status byte has been read unless the status register is empty (SBA/ not asserted).

### 3.3.3 The Transmit Data Register



Bits	Name	Description
<7:0>	TD<7:0>	<p>These bits contain the data that the host is loading into the NI3010A's transmit FIFO.</p> <p>They may contain data that the host wants to transmit on the Ethernet. The NI3010A will transmit the least significant bit (TD0) first.</p> <p>They may contain a block of multicast addresses that the host wants to load into the NI3010A's multicast address table.</p>

Access: Write only

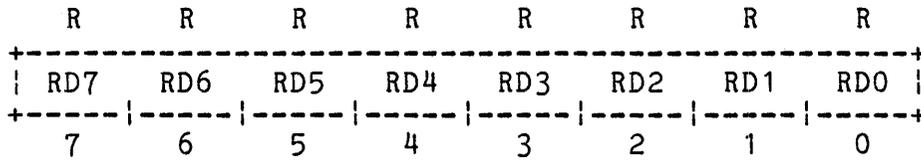
I/O Port Address: I/O port base address + 02H

Function: The host sets up a data block in its own memory and loads the NI3010A bus address and byte count registers. The host puts the block's starting address in the byte count registers. Then, the host initiates a transmit DMA by writing the interrupt enable register.

The data travel through the transmit data register and enter the NI3010A's transmit FIFO. The NI3010A interrupts the host when the transfer is complete.

Ordinarily, the host does not write the transmit data register under program control.

3.3.4 The Receive Data Register



Bits	Name	Description
<7:0>	RD<7:0>	<p>These bits contain data that make up a received Ethernet frame.</p> <p>They may contain data that the NI3010A has received from the Ethernet or loopback data.</p>

Access: Read only

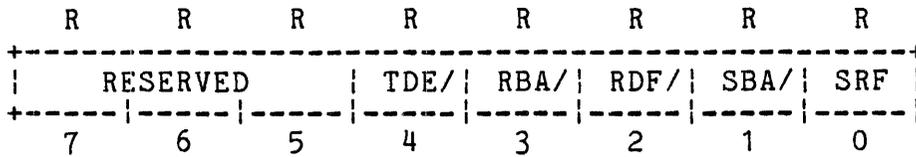
I/O Port Address: I/O port base address + 03H

Function: The host loads the bus address registers with the starting address of a block in memory where it expects the receive data. The host loads the byte count registers with the byte count of this block. Then, the host initiates a receive DMA by writing the interrupt enable register.

The data travel through the receive data register and enter host memory. The NI3010A interrupts the host after transferring the entire frame or when the byte count reaches zero.

Ordinarily, the host does not READ the receive data register under program control.

3.3.5 The Interrupt Status Register



Bits	Name	Description
<7:5>	Reserved	INTERLAN has reserved these bits. At present they have no meaning.
<4>	TDE/	<p>Transmit Data Empty. The host need not monitor this bit if it transfers data to the NI3010A by initiating transmit DMAs.</p> <p>When 1, indicates that the transmit data register is full.</p> <p>When 0, indicates that the transmit data register is empty and ready for another data byte.</p>
<3>	RBA/	<p>Receive Block Available.</p> <p>When 1, indicates that the receive data register is empty and the NI3010A does not have another frame available.</p> <p>When 0, indicates that a receive block is waiting in the receive FIFO. The receive data register has a byte ready to be read.</p>
<2>	RDF	<p>Receive Data Full. The host need not monitor this bit if it receives data from the NI3010A by initiating receive DMAs.</p> <p>When 1, indicates that the receive data register is full and ready to be read.</p> <p>When 0, indicates that the receive data register is empty.</p>

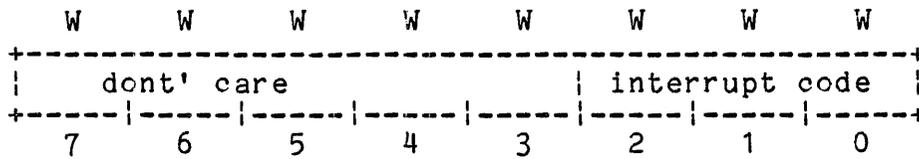
<1>	SBA/	Status Block Available.  When 1, indicates that no status data are waiting to be read.  When 0, indicates that a status block is waiting to be read. The status register has a byte ready to be read.
<0>	SRF	Status Register Full.  When 1, indicates that the status register is full and ready to be read.  When 0, indicates that the status register is empty.

Access: Read only

I/O Port Address: I/O port base address + 05H

Function: The host reads this register to determine the status of registers on the NI3010A.  
  
The host must read status data under program control. It monitors the SRF bit to determine when status data are available.

3.3.6 The Interrupt Enable Register



Bits	Name	Description
<7:3>	don't care	The NI3010A ignores these bits.
<2:0>	interrupt code	<p>By writing these bits, the host controls the nature of NI3010A interrupts.</p> <p>000 (0) Disables all NI3010A interrupts.</p> <p>001 (1) Enables a status-register-full (SRF) interrupt.</p> <p>010 (2) Enables a status-block-available (SBA/) interrupt.</p> <p>011 (3) Enables a receive-data-full (RDF) interrupt.</p> <p>100 (4) Enables a receive-block available (RBA/) interrupt.</p> <p>101 (5) Enables a transmit-data-empty (TDE/) interrupt.</p> <p>110 (6) Initiates a transmit DMA transfer and enables a transmit-DMA-done (TDD) interrupt.</p> <p>111 (7) Initiates a receive DMA transfer and enables a receive-DMA-done (RDD) interrupt.</p>

Access: Write only

I/O Port Address: I/O port base address + 08H

Function:

The host writes this register to enable a particular kind of interrupt from the NI3010A. Writing this register clears any previous interrupt enable.

When the interrupt occurs, it's latched by the NI3010A. The host clears the interrupt by writing a code of 0 to the interrupt enable register.

Because this register is write-only, you should keep a copy of it in host memory. After you write an interrupt code, update this copy before enabling the CPU for interrupts.

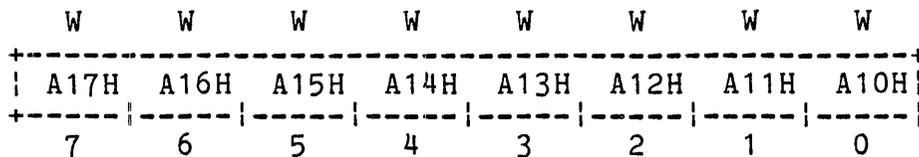
Ordinarily, the host does not enable a TDE/ or a RDF interrupt. The host transfers data to the NI3010A by loading the bus address and byte count registers, then writing a code of 6. The host receives data from the NI3010A by loading the bus address and byte count registers, then writing a code of 7.

The host receives a status block under program control. After issuing a Report and Reset Statistics command (18H) or a Report Collision Delay Times command (19H), the host reads the interrupt status register.

If SRF is 1, the host reads the status register; this read resets SRF to 0. Then, the host reads the interrupt status register again and checks SRF again.

If SRF is 0, the host checks SBA/. If SBA/ is 0, the host reads the interrupt status register and checks SRF again. If SBA/ is 1, no more status data are available.

3.3.7 The Extended Bus Address Register



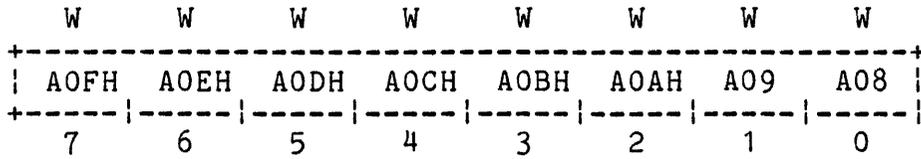
Bits	Name	Description
<7:0>	A<17H:10H>	These bits contain the upper 8 bits of a host MULTIBUS address.

Access: Write only

I/O Port Address: I/O port base address + 09H

Function: The host writes this register with the upper 8 bits of a starting address in host memory. This is the starting address of an upcoming DMA transfer.

3.3.8 The High Bus Address Register



Bits	Name	Description
<7:0>	A<OFH:08>	These bits contain the middle 8 bits of a host MULTIBUS address.

Access: Write only

I/O Port Address: I/O port base address + 0AH

Function: The host writes this register with the middle 8 bits of a starting address in host memory. This is the starting address of an upcoming DMA transfer.

3.3.9 The Low Bus Address Register

W	W	W	W	W	W	W	W
A07H	A06H	A05H	A04H	A03H	A02H	A01H	A00H
7	6	5	4	3	2	1	0

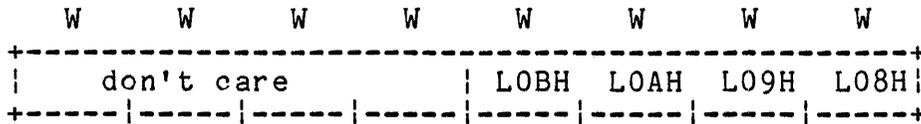
Bits	Name	Description
<7:0>	A<07H:00H>	These bits contain the lower 8 bits of a host MULTIBUS address.

Access: Write only

I/O Port Address: I/O port base address + 0BH

Function: The host writes this register with the lower 8 bits of a starting address in host memory. This is the starting address of an upcoming DMA transfer.

3.3.10 The High Byte Count Register



Bits	Name	Description
<7:4>	don't care	The NI3010A ignores these bits.
<3:0>	L<OBH:08>	These bits contain the upper 4 bits of a byte count. This byte count is the number of bytes in a host memory block.

Access: Write only

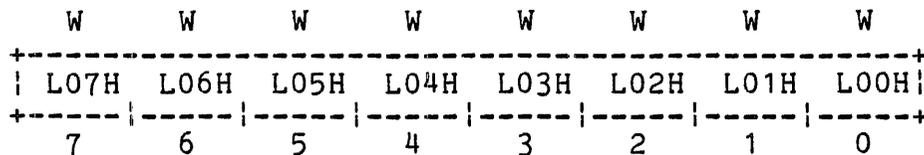
I/O Port Address: I/O port base address + 0CH

Function: The host writes this register with the upper 4 bits of a byte count.

If the host plans to initiate a transmit DMA, this is the byte count of the transmit data block in host memory.

If the host plans to initiate a receive DMA, this is the byte count of the block in host memory where the host expects the data.

3.3.11 The Low Byte Count Register



Bits	Name	Description
<7:0>	L<07H:00H>	These bits contain the lower 8 bits of a byte count. This byte count is the number of bytes in a host memory block.

Access: Write only

I/O Port Address: I/O port base address + 0DH

Function: The host writes this register with the lower 8 bits of a byte count.

If the host plans to initiate a transmit DMA, this is the byte count of the transmit data block in host memory.

If the host plans to initiate a receive DMA, this is the byte count of the block in host memory where the host expects the data.

### 3.4 NI3010A COMMAND DESCRIPTIONS

This section contains a complete description of each NI3010A command. Each command entry tell you

- what the command does
- how to issue the command
- what status codes are possible and what they mean
- the format of any related data

3.4.1 Reserved (00H)

Description: Interlan has reserved this command code. Do not  
issue it.

### 3.4.2 Set Module Interface Loopback (01H)

Description: This command puts the NI3010A in module interface loopback.

The NI3010A goes offline. When offline, the NI3010A cannot receive from or transmit to the network.

To send data through module interface loopback, the host must first put the NI3010A in module interface loopback, then load the NI3010A's transmit FIFO and issue a Load Transmit Data and Send command (29H). The transmit frame is entered into the receive memory and then to the receive data register.

RBA/ of the interrupt status register becomes 0, indicating that the receiver has a frame available. If the host previously enabled a receive-block-available interrupt, the NI3010A generates an interrupt.

RDF of the interrupt status register becomes 1, indicating that the receive data register has a data byte available. If the host previously enabled a receive-data-full interrupt, the NI3010A generates an interrupt.

The data do not go through the CRC generator, Manchester encoding/decoding, or transceiver level-shifting hardware.

Sequence: The host loads the NI3010A's command register with 01H, The command register is at the NI3010A's I/O port base address.

Status Codes: 00H Success. The NI3010A is now in module interface loopback and offline.

Data format: The data appear the same as data that were received from the Ethernet. The frame contains an FCS field, but the CRC value has no meaning.

- Notes:
1. This command is useful for testing the NI3010A's MULTIBUS hardware, the DMA controller, and the NM10A processor. Module interface loopback does not exercise the NM10A's Ethernet link management and physical channel logic.

### 3.4.3 Set Internal Loopback (02H)

Description: This command puts the NI3010A in internal loopback.

The NI3010A goes offline. When offline, the NI3010A cannot receive from or transmit to the network. The NI3010A's pass/fail LED turns on.

To send data through internal loopback the host must first put the NI3010A in internal loopback, then load the NI3010A's transmit FIFO and issue a Load Transmit Data and Send command (29H). The transmit frame is entered into the receive memory and then to the receive data register.

The NI3010A performs address recognition functions on data sent through internal loopback. If the NI3010A accepts the frame, the data await DMA transfer back to the host.

RBA/ of the interrupt status register becomes 0, indicating that the receiver has a frame available. If the host previously enabled a receive-block-available interrupt, the NI3010A generates an interrupt.

RDF of the interrupt status register becomes 1, indicating that the receive data register has a byte available. If the host previously enabled receive-data-full interrupts, the NI3010A generates an interrupt.

Sequence: The host loads the NI3010A's command register with 02H. The command register is at the NI3010A's I/O port base address.

Status Codes: 00H Success. The NI3010A is now in internal loopback and offline.

Data Format: The data appear the same as data that were received from the Ethernet. The frame contains an FCS field, but the CRC value has no meaning.

Notes:

1. This command is useful for testing the integrity of the NM10A's transmit and receive memories.
2. Frames that were in the receive FIFO before the host issued this command are not lost. The host receives these frames before receiving the loopback frames.
3. All loopback frames must satisfy the NI3010A's address detection logic to be accepted.
4. If the receive FIFO does not have enough room to accept the loopback frame, the frame is lost.

#### 3.4.4 Clear Loopback (03H)

Description: This command takes the NI3010A out of either module interface or internal loopback. The NI3010A's pass/fail LED turns off.

Sequence: The host loads the NI3010A's command register with 03H. The command register is at the NI3010A's I/O port base address.

Status Codes: 00H Success. The NI3010A is no longer in either module interface or internal loopback.

- Notes:
1. When the NI3010A leaves module interface loopback, the pass/fail LED turns off. The NI3010A remains offline.
  2. When the NI3010A leaves internal loopback, the pass/fail LED turns off. The NI3010A remains offline.

### 3.4.5 Set Promiscuous Mode (04H)

**Description:** This command puts the NI3010A in promiscuous mode. In this mode, the NI3010A receives all valid frames no matter where their destination address is.

Collision fragments (runt packets) are not valid frames.

**Sequence:** The host loads the NI3010A's command register with 04H. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The NI3010A is now in promiscuous mode.

**Notes:**

1. Promiscuous mode is useful when performing network-level diagnostic and maintenance functions.
2. When in promiscuous mode, the NI3010A receives its own transmissions. When it receives one of its own frames, however, the CRC value is not valid.

### 3.4.6 Clear Promiscuous Mode (05H)

**Description:** This command takes the NI3010A out of promiscuous mode. The NI3010A now only receives a frame if that frame's destination address matches the NI3010A's physical address, one of its multicast addresses, or the broadcast address.

**Sequence:** The host loads the NI3010A's command register with 05H. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The NI3010A is now in normal receive mode.

- Notes:**
1. Frames received while the NI3010A was in promiscuous mode may still reside in its receive FIFO.
  2. The NI3010A enters normal (default) receive mode when it powers up. In normal receive mode, the NI3010A  
  
Accepts a frame if its destination address matches the NI3010A's physical address, one of the NI3010A's multicast addresses, or the broadcast address.  
  
Rejects a frame if it has an alignment error or a CRC error.

### 3.4.7 Set Receive-on-Error Mode (06H)

**Description:** This command puts the NI3010A in receive-on-error mode. In this mode, the NI3010A accepts a frame even if it has a CRC error, an alignment error, or both. The NI3010A flags these errors in the frame's status byte.

**Sequence:** The host loads the NI3010A's command register with 06H. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The NI3010A is now in receive-on-error mode.

**Notes:**

1. Frames received with CRC or alignment errors will be counted by the statistics counters whether or not this mode is selected.

### 3.4.8 Clear Receive-on-Error Mode (07H)

Description: This command takes the NI3010A out of receive-on-error mode.

Sequence: The host loads the NI3010A's command register with 07H. The command register is at the NI3010A's I/O port base address.

Status Codes: 00H Success. The NI3010A is not in receive-on-error mode.

- Notes:
1. Frames received while the NI3010A was in receive-on-error mode may still reside in its receive FIFO.
  2. Frames received with CRC or alignment errors will be counted by the statistics counters whether or not this mode is selected.
  3. The NI3010A enters normal (default) receive mode when it powers up. In normal receive mode, the NI3010A  
Accepts a frame if its destination address matches the NI3010A's destination address, one of the NI3010A's multicast addresses, or the broadcast address  
Rejects a frame if it has an alignment error or a CRC error.

### 3.4.9 Go Offline (08H)

**Description:** This command logically disconnects the NI3010A's transmitter and receiver from the network. The NI3010A now cannot transmit frames to or receive frames from the network.

**Sequence:** The host loads the NI3010A's command register with 08H. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The NI3010A is now offline.

- Notes:**
1. The NI3010A is offline when it powers up.
  2. Data that were in the receive FIFO before the host issued this command are not affected.
  3. This command does not interrupt the NI3010A's receiver. If the NI3010A is currently receiving a frame, it completes the reception.
  4. If the NI3010A is offline and the host issues a Load Transmit Data and Send command (29H), the NI3010A returns a status code of 03H (Inappropriate Command).

3.4.10 H Go Online (09)

Description: This command logically connects the NI3010A to the Ethernet.

Sequence: The host loads the NI3010A's command register with 09H. The command register is at the NI3010A's I/O port base address.

Status Codes: 00H Success. The NI3010A is now online.

- Notes:
1. The NI3010A is offline when it powers up. The host must issue this command to put the NI3010A online.
  2. The host also must issue this command to go online after issuing a Clear Loopback command (03) or a Run Onboard Diagnostics command (0AH).

### 3.4.11 Run Onboard Diagnostics (OAH)

**Description:** This command orders the NM10A to execute its onboard diagnostic programs. The NI3010A returns a diagnostic status code in the status register.

The NI3010A goes offline. The pass/fail LED goes on when the diagnostic programs begin. The LED goes off again only if all the tests were successful. The NI3010A remains offline.

Section 5.2 describes the sequence of tests performed by the onboard diagnostic programs.

**Sequence:** The host loads the NI3010A's command register with OAH. The command register is at the NI3010A's I/O port base address.

**Status Codes:** Note that these are diagnostic status codes and different from the normal status codes.

- 00H Success. The onboard diagnostic programs executed without detecting any faults on the NM10A.
- 01H NM10A Microprocessor Memory Checksum Error. A diagnostic program detected a checksum error while testing the NM10A's program memory.
- 02H NM10A Microprocessor Memory Error. A diagnostic program detected an error while testing the NM10A's read/write memory.
- 03H Address Error. An error was detected while attempting to read the NM10A's Ethernet address.
- 04H Loopback Failure. The NM10A did not successfully transmit a test packet while in internal loopback.

- Notes:
1. This command takes about 3 seconds to execute.
  2. All information in the transmit and receive memories is destroyed.
  3. 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 (0B-0C)

Description: Interlan has reserved these command codes. Do not issue them.

NOTES: 1. 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 (0D)

**Description:** This command causes the NI3010 to insert its factory-programmed 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 3-1.

**Sequence:** Move command function code 0D into the NI3010's command register.

**Status Codes:**

00 -Success. The NI3010 will insert its preprogrammed physical address as the source address of any frame being transmitted onto the Ethernet.

**Notes:**

1. This is the default condition assumed by the NI3010 upon being powered on or after it has been reset.
2. After the issuance of this command, the on-board transfer DMA channel is restarted, thus causing any data in the transmit memory to be lost.

### 3.4.14 Clear Insert Source Address Mode (OE)

**Description:** This command will disable the insertion of the NI3010's preprogrammed physical address as the frame's source address prior to its being transmitted onto the Ethernet. In this mode, the user must include the source address as part of the data written into T\_DATA prior to its being transmitted. The frame format required by this mode for transmitting onto the Ethernet is shown in Figure 3-1a.

**Sequence:** The host loads the NI3010A's command register with 0EH. The command register is at the NI3010A's I/O port base address.

**Status Codes:**

00 -Success. The NI3010 will not insert its physical address as the source address of subsequently transmitted frames; it must be provided by the user.

**Notes:**

1. After the issuance of this command, the on-board transfer DMA channel is restarted, thus causing any data in the transmit memory to be lost.

### 3.4.15 Set Physical Address To Default (OF)

Description: This command causes the NI3010 to use its factory-programmed Ethernet address, cancelling out the effect of any other address previously provided by the user (see section 3.4.2 Load Physical Address (2C)).

While the NI3010 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 receiver's FIFO memory. This is the most efficient method of using the receiver.

Sequence: The host loads the NI3010A's command register with OFH. The command register is at the NI3010A's I/O port base address.

#### Status Codes:

00 -Success. The NI3010 will use its factory-programmed physical address for all subsequent operations, enabling its receiver physical address recognition hardware.

#### Notes:

1. This mode is assumed by the NI3010 as part of its initialization process which is invoked after power-up, or as a consequence of the execution of the Reset (3F) or Run On-Board Diagnostics (0A) commands. Issuing the Report and Reset Statistics command (18) at any time after this command has been issued will cause NI3010 to return its pre-programmed Ethernet address as its source address.
2. As mentioned above, the receiver's physical address recognition hardware is enabled when the default (factory-programmed) address is used as the NI3010 Ethernet address. The receiver's FIFO is therefore used in the most efficient manner.

### 3.4.16 Set Receive All Multicast Packets

Description: This command allows the NI3010 to receive all packets from the Ethernet whose destination addresses are multicast. The group address recognition filtering process is bypassed while the NI3010 is operating in this mode.

The user is responsible for filtering all multicast packets while the NI3010 is operating in this mode.

Sequence: The host loads the NI3010A's command register with 0FH. The command register is at the NI3010A's I/O port base address.

#### Status Codes:

00 -Success. The NI3010 will accept all multicast packets from the network if the Go Online is or has been issued, or from the user if the NI3010 has been placed in the Internal Loopback mode.

Notes: 1. On a large network with numerous "server" devices, placing the NI3010 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 (11)

**Description:** Issuing this command causes the NI3010 to filter all received multicast packets by means of 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 (10) command.

**Sequence:** The host loads the NI3010A's command register with 10H. The command register is at the NI3010A's I/O port base address.

**Status Codes:**

00 -Success. The NI3010 will accept multicast packets from the network or from internally looped back packets only if the packet's destination address matches one of those previously loaded in the group address recognition table.

- Notes:**
1. This mode is assumed by the NI3010 as part of its initialization process which is invoked after power-up, or as a consequence of the execution of the Reset (3F) or Run On-Board Diagnostics (0A) commands.
  2. The group address recognition table will contain 63 entries. If more multicast addresses need to be recognized, the host may issue the Set Receive All Multicast Packets (10) command described previously.

### 3.4.18 Perform Network Loopback Test (12)

**Description:** Issuing this command causes the NM10A 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 NM10A receives its own transmission, checks each packet for valid or invalid CRC as appropriate, and checks 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.

**Sequence:** The host loads the NI3010A's command register with 12H. The command register is at the NI3010A's I/O port base address.

**Status Codes:**

- 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 NM10A 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.
- 08 - Excessive Collisions Error; the NM10A 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:
1. The NM10A 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 (13H)

Description: Issuing this command causes the NM10A to transmit a minimum length packet through 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.

Sequence: The host loads the NI3010A's command register with 13H. The command register is at the NI3010A's I/O port base address.

#### Status Codes:

00 -Success; the NM10A 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: 1. The NM10A must be online and connected to an Ethernet transceiver which generates a heartbeat signal to successfully execute this command.

3.4.20 Reserved (14H - 17H)

Description: Interlan has reserved these command codes. Do not issue them.

### 3.4.20 Report and Reset Statistics (18H)

**Description:** This command causes the NI3010A to return a status block to host memory. This block contains network statistical data and identifies the NI3010A's network module along with its current firmware revision.

SBA/ of the interrupt status register becomes 0, indicating that a status block is available. If the host previously enabled a status-block available interrupt, the NI3010A generates an interrupt.

SRF of the interrupt status register becomes 1, indicating that the status register contains the first byte of the status block. If the host previously enabled a status-register-full interrupt, the NI3010A generates an interrupt.

The host reads the status block by reading the status register under program control. After reading the status register, the host reads the interrupt status register. If SBA/ is 0, the NI3010A has more status data available. If SRF is 1, the next status byte is in the status register, waiting to be read. When SBA/ returns to 1, the host knows that it read the last status byte.

After completing this command, the NI3010A resets all but one of its statistical counters to zero. The NI3010A does not zero the counter that records the number of frames in the receive FIFO.

**Sequence:** The host loads the NI3010A's command register with 18H. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The NI3010A has successfully transferred the status block to host memory.

**Data Format:** Figure 3-4 shows the format of the status block returned by this command.

**Notes:** 1. The NI3010A sets all its statistical counters to zero when it powers up or when the host issues a Reset command (3FH).

**Data Format:**

See Figure 3-4 on the next page.

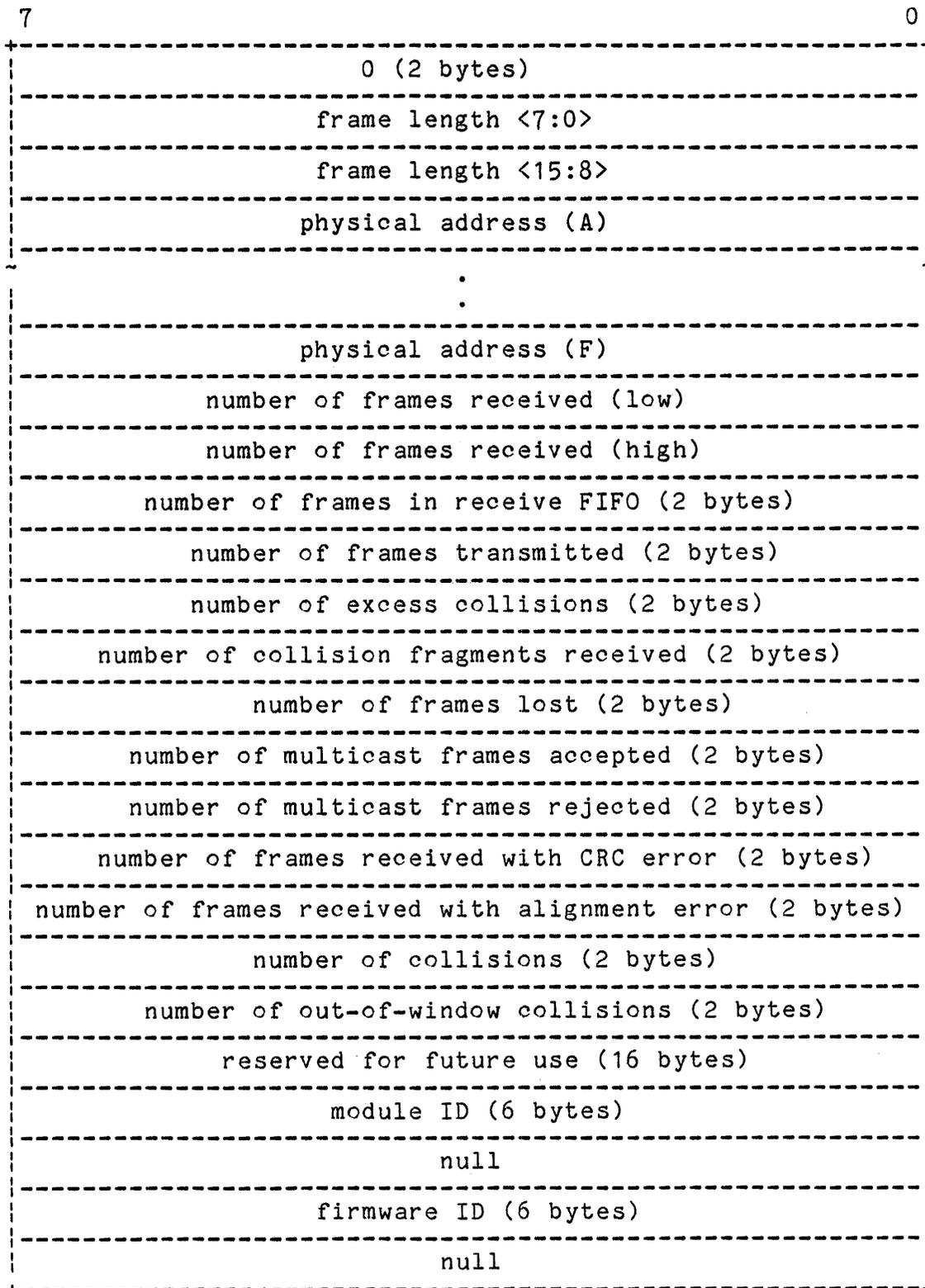


Figure 3-4. Reported Statistics

- LENGTH: a binary value for the number of bytes that follow (62 decimal).
- PHYSICAL ADDR: these 6 bytes are the NI3010A's physical address.
- 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: a binary value for the number of frames currently residing in the Receive FIFO buffer.
- NUMBER OF FRAMES TRANSMITTED: a binary value for the number of frames transmitted 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 collision fragments (ie, Runt packets) that were received and filtered by the controller.
- NUMBER OF FRAMES LOST: the number of frames not accepted by the Receive FIFO because sufficient space was not available.
- 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 Receive-on-Error has been selected.
- 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 Receive-on-Error has been selected.
- 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 first 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 NI2010 motherboard. (eg. "NM10A\_\_\_<NUL>")

FIRMWARE ID: these 8 ASCII bytes identify the firmware version installed in the NI3010A's NM10A module (eg."V3.00\_<NUL>").

NOTES:

1. The statistical counters are reset to zero (except Number of Frames in Receive FIFO) after issuance of this command or the Reset (3F) command, or the assertion of the RESET/ signal.
2. All bytes must be read from status register before attempting to read information from the receive data register.

### 3.4.21 Report Collision Delay Times (19H)

**Description:** This command causes the NI3010A to return a status block to host memory. This block contains the collision delay times for the last frame transmitted to the Ethernet.

The NI3010A measures up to 16 collision delay times for each frame before reporting its inability to transmit the frame.

The NI3010A measures a collision delay time from the start of the frame's transmission to the detection of the collision presence signal by the transceiver. The NI3010A records a collision delay time as the number of bits transmitted before sensing the transceiver's collision presence signal.

SBA/ of the interrupt status register becomes 0, indicating that a status block is available. If the host previously enabled a status-block-available interrupt, the NI3010A generates an interrupt.

SRF of the interrupt status register becomes 1, indicating that the status register contains the first byte of the status block. If the host previously enabled a status-register-full interrupt, the NI3010A generates an interrupt.

The host reads the status block by reading the status register under program control. After reading the status register, the host reads the interrupt status register. If SBA/ is 0, the NI3010A has more status data available. If SRF is 1, the next status byte is in the status register, waiting to be read. When SBA/ returns to 1, the host knows that it read the last status byte.

**Sequence:** The host loads the NI3010A's command register with 19H. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The NI3010A has successfully transferred the status block to host memory.

**Data Format:** Figure 3-5 shows the format of the status block returned by this command.

Notes:

1. An open Ethernet transmission cable causes the transmitting station's frame to collide with itself. This is due to signal reflection. In this case, all the collision delay times would be identical. The host can use these time domain reflectometry (TDR) data to calculate the approximate distance of the cable fault from the station's transceiver.

Propagation time on the Ethernet is about 5 nanoseconds/meter. The NI3010A's TDR resolution is 800 nanoseconds. Consequently, this command provides cable fault resolution to about 80 meters.

The collision delay times reported by this command include the inherent delay of the transceiver's collision delay circuit. You have to take this added delay into consideration when locating a cable fault.

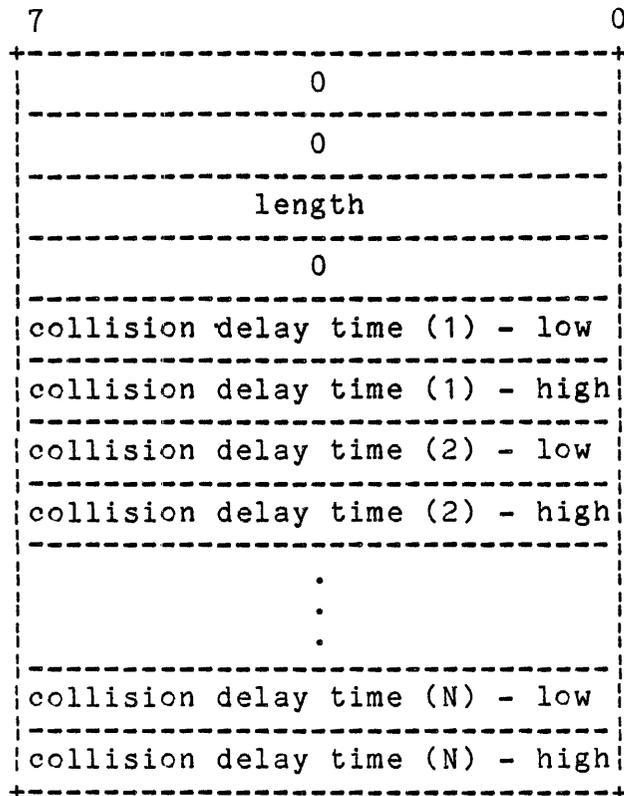


Figure 3-5. Collision Delay Times.

**LENGTH:** a binary value for the number of bytes that follow. This value, ranging from 0 to 32 (decimal), is two times the number 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 NI2010 has a timing resolution of 8 bits. (One bit represents 100 nanoseconds of transmission time.)

3.4.22 Reserved (1AH - 27H)

Description: Interlan has reserved these command codes. Do not issue them.

### 3.4.23 Load Transmit Data (28H)

**Description:** This command informs the NI3010A that it now has a block of transmit data.

The host issues this command after it has loaded the NI3010A's transmit FIFO with transmit data.

**Sequence:** The host loads the NI3010A's command register with 28H. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The NI3010A defines a block of transmit data in its transmit FIFO.

05H Buffer Size Exceeded. The amount of transmit data in the NI3010A's transmit FIFO exceeds 1508(decimal) bytes.

**Data Format:** The host must load transmit data into the transmit FIFO in the format shown in Figure 3-1. See Section 3.2 for this figure.

### 3.4.24 Load Transmit Data and Send (29H)

**Description:** This command informs the NI3010A that it now has a block of transmit data and orders the NI3010A to transmit that block on the Ethernet.

The NI3010A transmits all the data you've loaded into the transmit FIFO as a single Ethernet frame.

**Sequence:** The host loads the NI3010A's command register with 29H. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The NI3010A successfully transmitted the frame on the Ethernet without any collisions.

01H Success with Retries. The NI3010A successfully transmitted the frame on the Ethernet. While transmitting, it detected at least one collision and had to re-schedule transmission.

03H Inappropriate Command. The NI3010A is offline and not in a loopback mode.

attempting  
before the  
04H Failure. The NI3010A detected an error while to transmit. The "jabber" control timer expired frame was successfully delivered to the network.

05H Buffer Size Exceeded. The amount of data in the transmit FIFO exceeds 1508 bytes. The NI3010A purges its transmit FIFO.

06H Frame too Small. The amount of data in the transmit FIFO is less than 8 bytes. The NI3010A purges its transmit FIFO.

08H Excessive Collisions. The NI3010A detected more than 16 collisions while attempting to transmit the frame on the Ethernet. The NI3010A purges its transmit FIFO.

**Data Format:** The host must load transmit data into the transmit FIFO in the format shown in Figure 3-1. See section 3.2 for this figure.

**Notes:**

1. When it transmits the frame on the Ethernet, the NI3010A inserts its physical address into the source address field and the CRC value into the FCS field.
2. The minimum data field is 46 bytes long. If the host supplied less than 46 bytes, the NI3010A zero-fills

the data field to complete the minimum size.

### 3.4.25 Load Group Address(es) (2AH)

**Description:** This command loads multicast addresses into the NM10A's multicast address table. The host may load up to 63 multicast addresses.

The host must have previously loaded these addresses into the transmit FIFO. It may have done this by initiating a transmit DMA.

The NI3010A uses these addresses for multicast address recognition.

**Sequence:** The host loads the NI3010A's command register with 2AH. The command register is at the NI3010A's I/O port base address.

**Status Codes:**

00	Success. The group of multicast addresses is now in the onboard multicast address table.
05	Buffer Size Exceeded. Loading these addresses into the multicast address table would result in more than 63 multicast addresses. The NI3010A does not load any of the addresses into the table. It deletes them from its transmit FIFO.
0AH	Buffer Alignment Error. The set of multicast addresses stored in the transmit FIFO is not a multiple of six bytes. The NI3010A does not load any of these addresses into the table. It deletes them from its transmit FIFO.

**Data Format:** The host must load the multicast addresses into the transmit FIFO in the format shown in Figure 3-6.

**Notes:**

1. The NI3010A does not validate the supplied multicast addresses. If the host enters non-multicast addresses into the multicast address table, they will reside there, but the NI3010A won't use them for multicast address recognition.
2. When the NI3010A recognizes a frame's destination address as a multicast address and checks this destination address against its table of multicast addresses, it checks the entire 48-bit field, not any selected subfield.

data and neglects to issue a Load Transmit Data command (28H) or Load Transmit Data and Send command (29H) before loading the transmit FIFO with multicast addresses, then the NI3010A attempts to load both sets of data into its multicast address table when it receives the Load Group Address(es) command (2AH). This will probably result in a status code of 05H or 0AH.

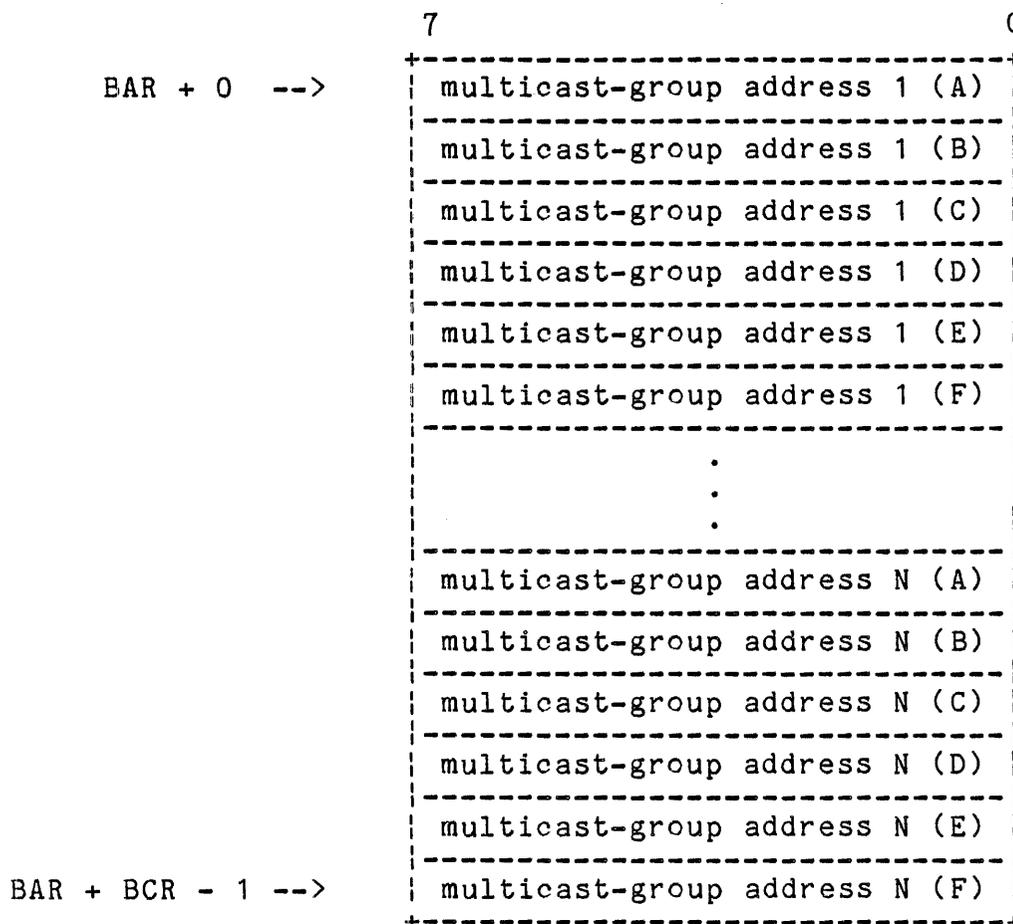


Figure 3-6.  
 Multicast address format

### 3.4.26 Delete Group Address(es) (2BH)

**Description:** This command removes selected multicast addresses from the NI3010A's multicast address table.

The host must have previously loaded the transmit FIFO with the multicast addresses it wants removed from the multicast address table. It may have done this by initiating a transmit DMA.

**Sequence:** The host loads the NI3010A's command register with 2BH. The command register is at the NI3010A's I/O port base address.

**Status Codes:** 00H Success. The multicast addresses loaded into the transmit FIFO are no longer in the multicast address table. Neither are they in the transmit FIFO.

05H Buffer Size Exceeded. The host loaded more than 63 multicast addresses into the transmit FIFO. The NI3010A does not delete any addresses from its multicast address table. The NI3010A does delete the multicast addresses from its transmit FIFO.

OAH Buffer Alignment Error. The set of multicast addresses stored in the transmit FIFO is not a multiple of six bytes. The NI3010A does not delete any addresses from its multicast address table. The NI3010A does delete the multicast addresses from its transmit FIFO.

**Data Format:** The host must load the multicast addresses in the transmit FIFO in the format shown in Figure 3-6. See the description of the Load Group Address(es) command for this figure.

- Notes:**
1. It's the host's responsibility to keep track of the current contents of the NI3010A's multicast address table. The host must delete specific addresses from the NI3010A's multicast address table.
  2. 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. The only way the host can clear this table without knowing what's in it is by issuing a Reset command (3FH).

### 3.4.27 Load Physical Address (2C)

**Description:** This command loads a user-specified Ethernet physical address into the NI3010. The factory-programmed Ethernet physical address is not used by the NI3010. The physical address recognition hardware is disabled by this command, forcing the NI3010 processor to dismiss packets whose physical addresses do not match the one loaded by this command. This process occurs prior to output to the R\_DATA register; thus, the user will not be burdened with undesired packets. Each physically addressed packet will consume space in the receiver FIFO memory until it has passed through the filtering process.

**Sequence:** Move the new physical address values into the transmit FIFO, in the order A, B, C, D, E, F (A is the first address byte transmitted). This may be done by initiating a transmit DMA transfer. Move the command function code 2C into the NI3010's command register.

**Status Codes:**

- 00 -Success. The data specified is now the new physical address of the NI3010.
- 0A -Buffer Alignment Error. The data supplied is not equal to 6 bytes in length. The physical address of the NI3010 is not changed.

- Notes:**
1. Different physical addresses may be loaded at any time by this command. Loading a new physical address while the NI3010 is online may give bizarre results (i.e., previously acceptable packets waiting in the receiver FIFO will be discarded prior to their delivery to R\_DATA).
  2. Use of this command disables the physical address recognition hardware forcing the NI3010 processor to filter all physically addressed packets prior to delivering them to R\_DATA.
  3. The NI3010's physical address recognition hardware may be enabled by issuing the Set Physical Address to Default command (0F).

3.4.28 Reserved (2DH - 3EH)

Description: Interlan has reserved these command codes. Do not issue them.

### 3.4.21 Reset (3FH)

**Description:** This command returns the NI3010A to its power-up state. When the host issues a Reset command (3FH), the NI3010A

Goes offline. The NI3010A logically disconnects its transmitter and receiver from the network.

Enters normal (default) receive mode. When the NI3010A goes back online, it will not accept frames with CRC or alignment errors.

Deletes all multicast addresses in the multicast address table.

Resets all its statistical counters to zero.

Executes its onboard diagnostic programs and returns a diagnostic status code in the status register.

**Sequence:** The host loads the NI3010A's command register with 3FH. The command register is at the NI3010A's I/O port base address.

**Status Codes:** Note that these are diagnostic status codes and different from the normal status codes.

- 00H Success. The onboard diagnostic programs executed without detecting any faults on the NM10A.
- 01H NM10A Microprocessor Memory Checksum Error. A diagnostic program detected a checksum error while testing the NM10A's program memory.
- 02H NM10A Microprocessor Memory Error. A diagnostic program detected a checksum error while testing the NM10A's read/write memory.
- 03H Address Error. An error was detected while attempting to read the NM10A's Ethernet address.
- 04H Loopback Failure. The NM10A did not successfully transmit a test packet while in internal loopback.

Notes:

1. This command is like all other Type I commands, and therefore may only be issued after status from the previous command has been read.  
The control signal, RESET/, may be asserted at any time.
2. The Reset command (3FH) takes about 1/2 second to execute.

### 3.5 NI3010A PROGRAMMING

This section contains suggestions on how to write host programs that handle the NI3010A. In addition to some programming guidelines, it contains structured pseudo-code for

- handling an NI3010A interrupt (includes receiving data)
- issuing an NI3010A command
- transmitting data to the Ethernet
- receiving a status block from the NI3010A

When you read this section, remember that the interrupt enable register is a write-only register. You have to keep a current copy of it in host memory. Reading the interrupt enable register means reading this copy.

#### 3.5.1 How to Manage NI3010A Interrupts

You'll notice that although the NI3010A is capable of generating seven kinds of interrupts, the procedures outlined here only make use of three. They are receive-block-available (RBA/), receive-DMA-done (RDD), and transmit-DMA-done (TDD) interrupts.

Because an Ethernet frame may arrive at any time, these examples assume that you enable an RBA/ interrupt when there is nothing else for the NI3010A to do. You enable an interrupt by writing an interrupt code to the interrupt enable register. Because this register is write-only, you should keep a copy of it in host memory. After you enable an interrupt, update this copy.

Often what you want to do is check your copy of the interrupt enable register, set the register to a new value, and then update your copy. To maintain a valid copy, you don't want an interrupt to occur in the midst of this operation. Be sure to disable interrupts (the CPU disable) before handling the interrupt enable register.

#### 3.5.2 When not to Write the Bus Address and Byte Count Registers

Don't write the bus address registers or the byte count registers while the interrupt enable register holds a code of 6 or 7. Doing so may cause the NI3010A to read or write incorrect areas of host memory.

An interrupt code of 6 enables a transmit-DMA-done interrupt and initiates a transmit DMA; and an interrupt code of 7 enables a receive-DMA-done interrupt and initiates a receive DMA.

When the NI3010A completes a transmit DMA, the bus address registers contain the address in host memory of the next sequential location. You may find this useful when transmitting packets from a single long buffer.

When the NI3010A completes a receive DMA, the bus address registers contain the address in host memory of the next sequential location. You may find this useful when receiving packets into a single long buffer.

### 3.5.3 When You Must Read the Status Register

After issuing a command to the NI3010A, always read the status register before issuing another command.

If the issued command returns more than one byte of status, all status bytes must be read together. After one byte of status has been read, all other registers are ignored until all status bytes have been read.

In addition, you must read the status register after the NI3010A experiences a hard reset. A hard reset occurs when the reset line on the backplane goes high. A hard reset causes the NI3010A to run its power-up tests. Approximately, 1/2 second later, the NI3010A has a diagnostic status code waiting in its status register.

### 3.5.4 Handling NI3010A Interrupts

This routine assumes that you can get three types of interrupts. They are

- \* The RBA/ interrupt. IE\_REG is 4. The NI3010A is telling the host that it has a receive frame ready. Your response is to initiate a receive DMA by setting IE\_REG to 7.
- \* The RDD interrupt. IE\_REG is 7. The NI3010A is telling the host that it completed a receive DMA. Your response is to enable an RBA/ interrupt by setting IE\_REG to 4. The routine assumes that you enable a receive-block-available interrupt when there is nothing else for the NI3010A to do.
- \* The TDD interrupt. IE\_REG is 6. The NI3010A is telling the host that it completed a transmit DMA. Your response is to enable an RBA/ interrupt by setting IE\_REG to 4. The routine assumes that you enable a receive-block-available interrupt when there is nothing else for the NI3010A to do.

The host routine that transfers data to the NI3010A sets IE\_REG to 6 and then checks IE\_REG. When IE\_REG is no longer 6, the host transfer routine knows that the NI3010A generated a TDD interrupt and that the interrupt handler changed IE\_REG.

```
disable CPU interrupts
get current IE_REG contents
set IE_REG to 0

if IE_REG was a 4 then
    load bus address registers
    load byte count registers
    set IE_REG to 7
end_if

else_if IE_REG was a 7
    wake up receive packet process
    give it this packet
    set IE_REG to 4
end_else

else_if IE_REG was a 6
    set IE_REG to 4
end_else

enable CPU interrupts
```

;if RBA/ interrupt  
;get receive buffer ready  
;initiate receive DMA  
;if RDD interrupt  
;op system dependent  
;enable RBA/ interrupt  
;if TDD interrupt  
;enable RBA/ interrupt

Handling NI3010A Interrupts

### 3.5.5 Issuing an NI3010A Command

The host writes the command register. Then, it waits until the interrupt status register shows its SRF bit equal to 1, indicating that status is available in the status register. Finally, the host reads the status register.

```
+-----+
| write C_REG           ;write command register
| read IS_REG          ;read interrupt status
|                       register
|
| repeat               ;wait until status
|   read IS_REG        ; register is full
| until SRF is 1
| read S_REG           ;read status register
+-----+
```

Issuing a Command to the NI3010A

### 3.5.6 Transmitting Data to the Ethernet

The host transfers data to the NI3010A and orders the NI3010A to transmit the data on the Ethernet.

First, the host disables CPU interrupts and checks if the NI3010A is busy by reading IE\_REG.

If the NI3010A is not busy, IE\_REG is either 0 (NI3010A interrupts disabled) or 4 (an RBA/ interrupt enabled).

If the NI3010A is busy, the host enables CPU interrupts to allow the NI3010A to finish whatever it was doing. The host transmit routine then checks IE\_REG until it sees a 0 or a 4. Remember, the interrupt handler sets IE\_REG to 4 after servicing an RDD or TDD interrupt.

When the host sees that IE\_REG is a 0 or a 4, it disables CPU interrupts and checks IE\_REG once more.

Why again? The NI3010A could have generated an interrupt while the host was checking IE\_REG and before it disabled CPU interrupts.

If IE\_REG changed, the host must once again enable CPU interrupts and let the NI3010A finish what it started. The host monitors IE\_REG. When IE\_REG is a 0 or a 4, the host disables CPU interrupts and checks IE\_REG once more.

If IE\_REG is still a 0 or a 4, the host sets IE\_REG to 0. This disables interrupts from the NI3010A.

Then, the host enables CPU interrupts.

The NI3010A may have a receive-block-available. If it does, the host still must enable an RBA/ interrupt to get one. Because IE\_REG is 0, the NI3010A can't generate any type of interrupt while the host loads the bus address and byte count registers. The host may, however, honor an interrupt from another I/O device.

Then, the host disables CPU interrupts. It initiates a transmit DMA and enables a transmit-DMA-done interrupt by setting IE\_REG to 6.

If the NI3010A has an RBA/ interrupt pending, it stays pending. That's because the host only enabled a TDD interrupt.

The host checks IE\_REG until it's no longer 6. When IE\_REG is no longer 6, the host transfer routine knows that the NI3010A generated a TDD interrupt and that the interrupt handler changed IE\_REG. When the interrupt handler services the TDD interrupt, it ends by enabling an RBA/ interrupt. If the NI3010A has an RBA/ interrupt pending, that interrupt occurs as soon as it's enabled.

Then, the host issues a Load Transmit Data and Send command (29H).

```
disable CPU interrupts

while IE_REG is not a 0 or 4 do           ;if the NI3010A is busy
    enable CPU interrupts                 ;wait until it's not
    repeat                                ;busy
        read IE_REG
    until IE_REG is a 0 or a 4
    disable CPU interrupts
        read IE_REG
end_while

set IE_REG to 0

enable CPU interrupts
load bus address registers
load byte count registers

disable CPU interrupts
set IE_REG to 6                           ;enable TDD interrupt
enable CPU interrupts

if IE_REG is a 6 then                     ;when the interrupt handler
    wait until it is not a 6             ;takes care of the TDD
end_if                                     ;interrupt, it sets IE_REG
                                           ;to 4.

issue a Load Transmit Data and Send command (29H)
```

DMA Transfers to the NI3010A

### 3.5.7 Receiving a Status Block from the NI3010A

The host receives a status block from the NI3010A.

First, the host reads IS\_REG until the SBA/ bit is 0, indicating that the NI3010A has a status block available.

Then, if IS-REG also has SRF equal to 1, indicating that a status byte is available in the status register, the host reads the status register. This read resets SRF to 0. If the NI3010A has another status byte available, it loads that byte into the status register and sets SRF to 1 again.

Then, the host checks SBA/. If SBA/ is still 0, the host checks SRF. If SRF is 1, indicating that another status byte is available in the status register, the host reads the status register. If SRF is 0, the host checks SBA/.

If SBA is still 0, the host checks SRF again. If SBA/ is 1, the host knows it has received the complete status block.

```
+-----+
| repeat                                     |
|   read IS_REG                             |;read interrupt status
|   if SRF is 1 then                         |;if status register full
|     read S_REG                             |;get status byte
|   end_if                                   |
| until SBA/ is 1                           |;until no more status data
+-----+
```

Reading the Status Block

## CHAPTER FOUR

### FUNCTIONAL DESCRIPTION

This chapter contains a functional description of the NI3010A MULTIBUS Ethernet Controller. It describes the NI3010A's architecture and the Ethernet frame format. It tells you what happens when the NI3010A transmits frames, receives frames, and runs its onboard diagnostic programs.

#### 4.1 NI3010A BOARD ARCHITECTURE

##### 4.1.1 Physical Description

The NI3010A is a single board assembly that fits into one MULTIBUS slot. It consists of the Interlan MULTIBUS Interface Board (MIB) and the Interlan NM10A Ethernet Protocol Module.

The NM10A measures 10.0"x6.75"x0.375" and is flush-mounted on the MIB. There is an insulating mylar sheet between the boards.

The NM10A connects to the MIB and to the Ethernet. It connects to the MIB through a field of 2 by 30 posts that enter through the bottom of its P1 connector. It connects to the Ethernet through a 16-pin flat cable that attaches to its P2 connector.

##### 4.1.2 Functional Overview

The MIB contains the logic necessary for transferring data between the NM10A and the host MULTIBUS system. The NM10A contains the data communications logic.

#### 4.2 THE MULTIBUS INTERFACE BOARD

The major parts of the MIB are

- the MULTIBUS "SLAVE" interface logic
- the DMA controller
- the state machine

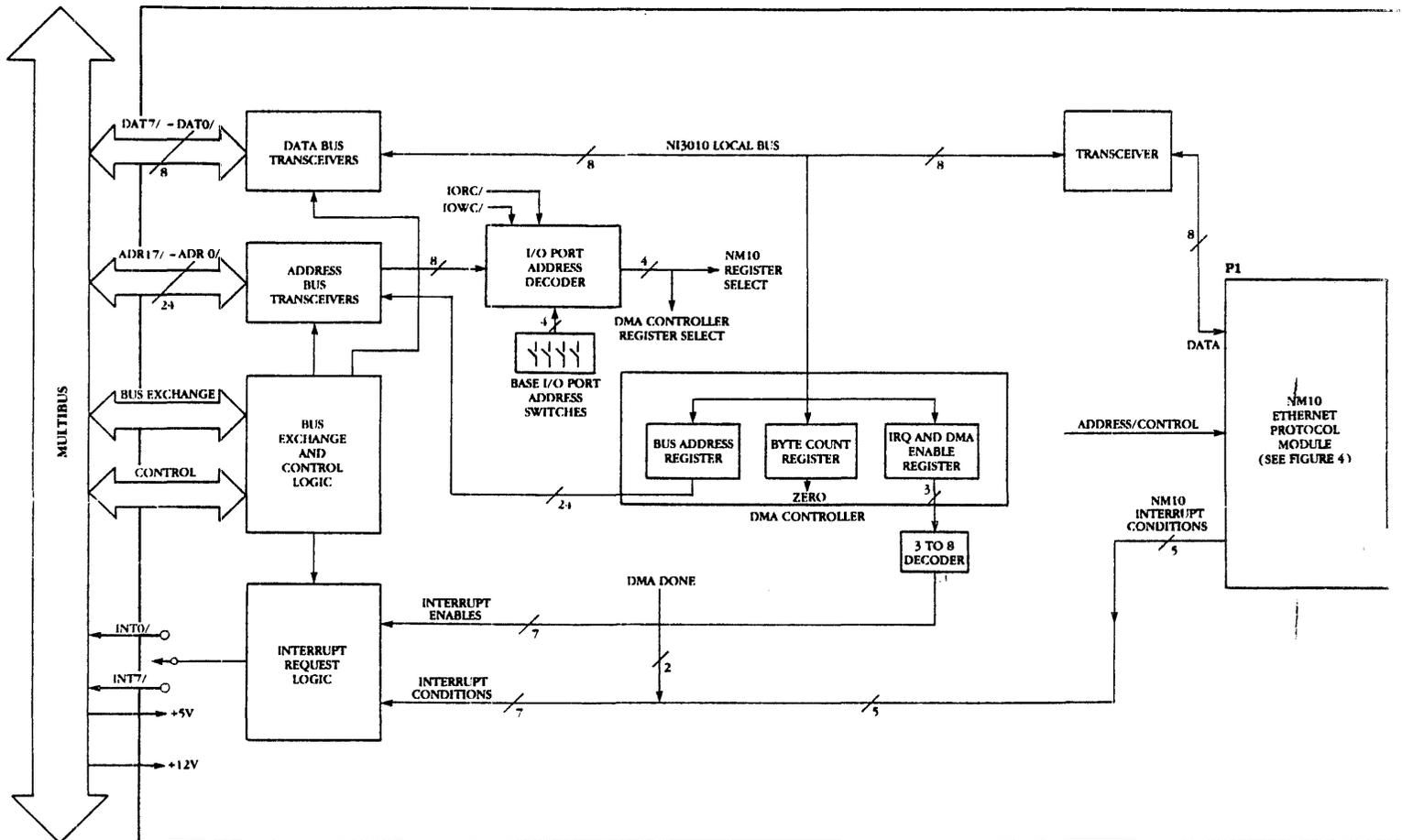


Figure 4-1. NI3010A MIB Block Diagram

#### 4.2.1 THE MULTIBUS "SLAVE" Interface

The MULTIBUS "SLAVE" interface handles data transactions initiated by the host processor. The host deals with this interface when it addresses one of the I/O ports assigned to the NI3010A and writes or reads an MIB internal register. This interface does not handle DMA (direct memory access) or interrupt sequences initiated by the NI3010A.

A MULTIBUS data transaction consists of the following steps:

1. The "SLAVE" logic determines if the I/O port address on the MULTIBUS references one of the NI3010A's internal registers. A DIP switch on each NI3010A determines its I/O port base address.  
  
Each NI3010A internal register is assigned an I/O port address that equals the base address plus an offset.
2. If the host specified a read, the "SLAVE" logic places data on the MULTIBUS.
3. If the host specified a write, the "SLAVE" logic accepts data from the MULTIBUS.

#### 4.2.2 The DMA Controller

The DMA controller contains registers that support DMA data transfers to and from host memory. These registers are the three bus address registers, the two byte count registers, and the interrupt enable register. Table 4-1 lists the DMA registers along with their I/O port addresses.

Table 4-1 The DMA Registers

Register	I/O Port Address
! interrupt enable register (IE)	! I/O port base address + 08H !
! extended bus address register (EBAR)	! I/O port base address + 09H !
! high bus address register (HBAR)	! I/O port base address + 0AH !
! low bus address register (LBAR)	! I/O port base address + 0BH !
! high byte count register (HBCR)	! I/O port base address + 0CH !
! low byte count register (LBCR)	! I/O port base address + 0DH !

DMA transfers are one data byte at a time. The NI3010A has control of the MULTIBUS during this transfer. Another device may take over the MULTIBUS between data transfers. When the NI3010A regains the MULTIBUS, it resumes its DMA transfers from where it left off.

#### 4.2.2.1 The Interrupt Enable Register

The interrupt enable register is a write-only register. Its lower 3 bits contain an interrupt code. By setting various interrupt codes, the host can

- \* Disable all NI3010A interrupt requests
- \* Enable a status-register-full, a status-block-available, a receive-data-full, a receive-block-available, or a transmit-data-empty interrupt
- \* Initiate a DMA transfer to or from the host and enable an interrupt at the completion of the transfer

#### 4.2.2.2 The Bus Address Registers

The three bus address registers are write-only registers. The host must load these registers with a 24-bit host memory address. The upper 8 bits go into E\_BAR, the middle 8 bits into H\_BAR, and the lower 8 bits into L\_BAR. The DMA controller uses this address as the starting address of a DMA transfer to or from host memory.

#### 4.2.2.3 The Byte Count Registers

The two byte count registers are write-only registers. Before initiating a DMA transfer, the host must load the byte count registers with a 12-bit byte count. The upper 4 bits of H\_BCR are don't-care bits. The host writes the upper 4 bits of the byte count into the lower 4 bits of H\_BCR and the lower 8 bits of the byte count into L\_BCR.

The DMA controller decrements the byte count each time it transfers a byte to host memory or receives a byte from host memory.

#### 4.2.4 The State Machine

A PROM-based finite state machine controls the NI3010A DMA and command functions. This state machine

- \* Transfers commands to the NM10A
- \* Generates all MULTIBUS "MASTER" signals for DMA and interrupt operation
- \* Manages data transfers between the NM10A and the MULTIBUS data bus (during DMA)

### 4.3 THE NM10A ETHERNET PROTOCOL MODULE

The NM10A contains the data communications control logic that interfaces the MIB to the Ethernet. This module complies with the Xerox/Intel/Digital Ethernet Specification, Version 1.0. Figure 4-2 is a block diagram of the NM10A.

The NM10A has a microprocessor-bus compatible interface at its 60-pin P1 connector. Data travel to and from the MIB through an 8-bit bidirectional data bus to five internal 8-bit registers. Transmit data then enter a transmit buffer and await transfer to the Ethernet. Receive data from the Ethernet enter a receive buffer and await transfer to host MULTIBUS memory.

#### 4.3.1 The NM10A Transmit and Receive Buffers

The NM10A contains two FIFO (first-in, first-out) buffers.

- \* A 1.5 Kbyte transmit buffer. This buffer allows the host to transfer each transmit packet to the NI3010A only once, independent of network traffic. To send the data out on the Ethernet, the host must issue a Load Transmit Data and Send command (29H) to the NI3010A.

If a network access collision occurs, the NM10A automatically reschedules transmission. Because the transmit frame is still available in the transmit buffer, the host need not send it again to the NI3010A.

- \* A 13.5 Kbyte receive buffer. This buffer stores receive frames. It buffers the MULTIBUS from the unpredictable arrival times of network traffic, consequently reducing the time-critical service requirements on the host MULTIBUS system.

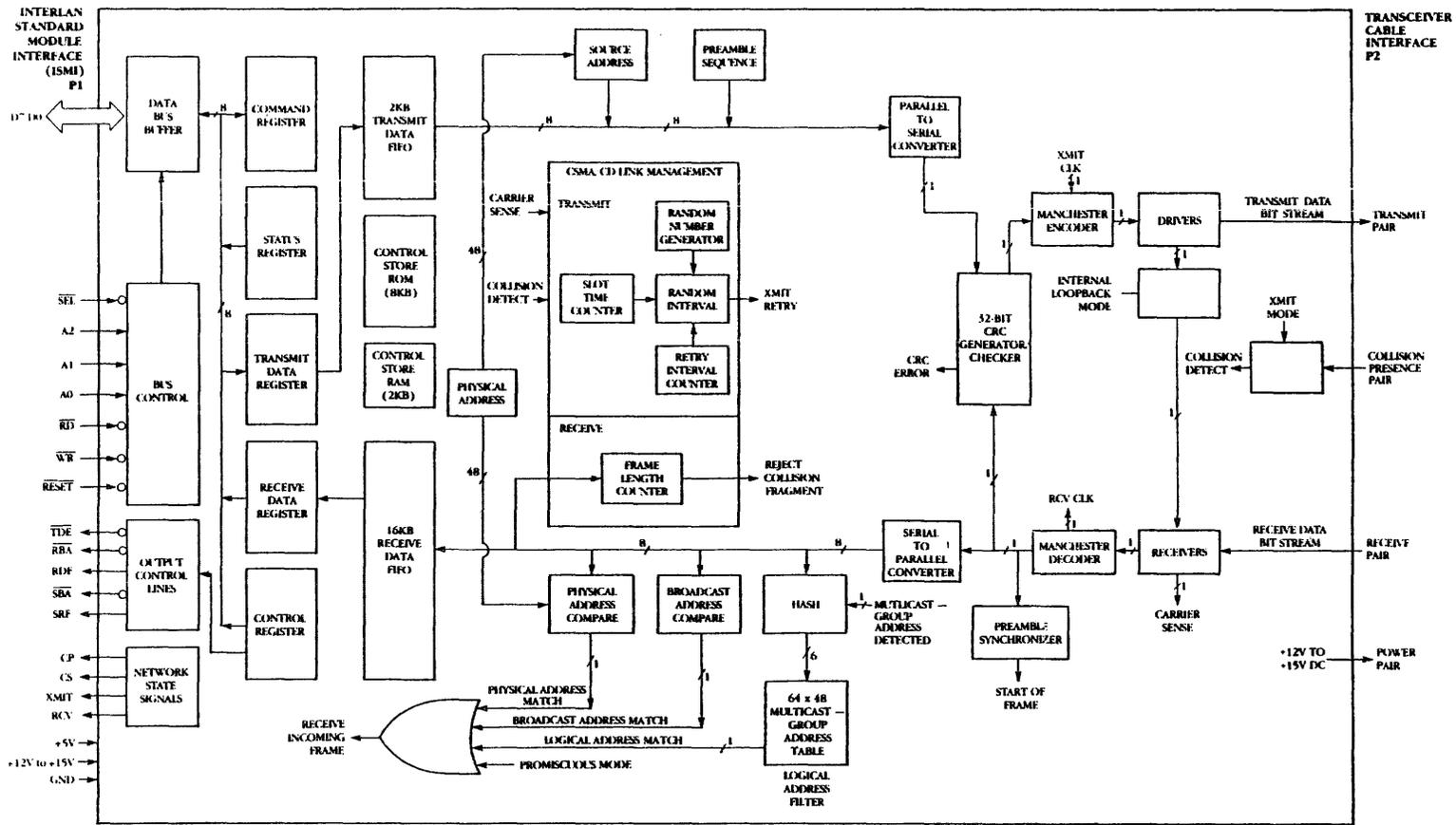


Figure 4-2. Functional Diagram of the NM10A

### 4.3.2 The NM10A Registers

Each NM10A register has an I/O port address. The MIB decodes this I/O port address into a 3-bit address that it presents to the NM10A. Table 4-2 lists the NM10A registers along with their I/O port addresses.

Table 4-2 The NM10A Registers

! Register	! I/O Port Address	!
! command register (C_REG)	! I/O port base address + 00H	!
! status register (S_REG)	! I/O port base address + 01H	!
! transmit data register (T_DATA)	! I/O port base address + 02H	!
! receive data register (R_DATA)	! I/O port base address + 03H	!
! interrupt status register (IS_REG)	! I/O port base address + 05H	!
! (also called the control register)	!	!

#### 4.3.2.1 The Command Register

The command register is a write-only register. The host sends a command to the NI3010A by writing the command register. Commands include such functions as setting loopback modes, setting receive modes, and running the onboard diagnostic programs.

#### 4.3.2.2 The Status Register

The status register is a read-only register. The host reads this register to determine the status resulting from a command. The NI3010A loads the status register with a status code after executing a host command.

#### 4.3.2.3 The Transmit Data Register

The transmit data register is a write-only register. When the host transfers data to the NI3010A via DMA, these data travel through the transmit data register.

To transfer data to the NI3010A, the host sets up a block of transmit data in its own memory, loads the NI3010A bus address registers with the block's starting address, loads the NI3010A byte count registers with the block's byte count, then initiates a DMA transfer. When the DMA controller accepts a data byte from host memory, it loads the transmit data register on the NM10A. The data byte then enters the transmit FIFO.

Ordinarily, the host does not write the transmit data register under program control. It does, however, have that capability. Instead of

initiating a DMA transfer, the host can write the data byte directly into the transmit data register. The NM10A then transfers that byte to the transmit FIFO. If the host previously enabled a transmit-data-empty interrupt, the NI3010A then interrupts the host, signifying that the transmit data register is now empty. The host must not write the transmit data register until the TDE/ bit in the interrupt status register becomes 0.

#### 4.3.2.4 The Receive Data Register

The receive data register is a read-only register. When the NI3010A transfers data to the host via DMA, these data travel through the receive data register.

When the NM10A receives a frame from the Ethernet, it stores that frame in the receive FIFO. To receive data from the NI3010A, the host then loads the NI3010A bus address registers with an address in MULTIBUS memory. This is the starting address of the block where the host will expect the receive data. The host also loads the byte count registers with the block's byte count. Then, the host initiates a DMA transfer. When the DMA controller receives a data byte from the receive FIFO, it loads the receive data register. The data byte then goes to host memory.

Ordinarily, the host does not read the receive data register under program control. It does, however, have that capability. Instead of initiating a DMA transfer, the host can read a data byte directly from the receive data register. If the receive FIFO contains more data, the NM10A loads the receive data register with the next data byte. If the host previously enabled a receive-data-full interrupt, the NI3010A then interrupts the host, signifying that the receive data register is again full. The host must not read the receive data register until the RDF bit in the interrupt status register becomes 1.

#### 4.3.2.5 The Interrupt Status Register

The interrupt status register is a read-only register. The host reads this register to determine the nature of an interrupt from the NI3010A.

#### 4.4 THE ETHERNET FRAME FORMAT

An Ethernet frame consists of five fields. They are

- the destination address
- the source address
- the type field
- the data field
- the frame check sequence (FCS)

A 64-bit preamble precedes each frame. This preamble ensures that receivers synchronize to the incoming bit stream. Each frame terminates with an interframe spacing of at least 9.6 microseconds. Figure 4-3 illustrates the format of an Ethernet frame. The preamble bits are supplied by the NM10A. The source address bytes may be supplied by the user or the NM10A. (See Set/Clear Insert Source Address Mode command.) All other information must be supplied by the user.

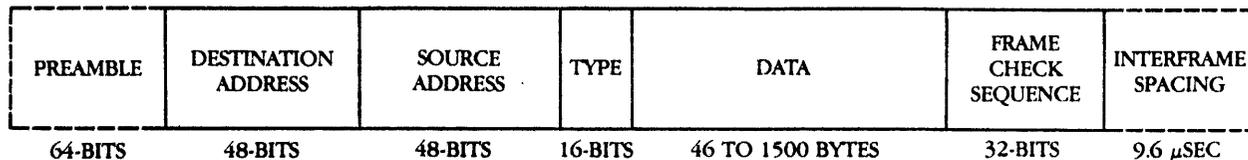


Figure 4-3. Ethernet Frame Format

##### 4.4.1 The Destination Address

The destination address is always 48 bits long. These bits specify the address of the station or stations for which the frame is intended. The NI3010A requires that all frames have a destination address.

The user provides the destination address. There are three types of destination addresses. They are

- \* A physical address. Each NI3010A is assigned a unique physical address by Interlan. The user cannot alter this address.
- \* A multicast address. This is an address assigned by the user. The user can associate a multicast address with a group of logically related stations. Each NI3010A can have up to 63 multicast addresses.
- \* The broadcast address. This consists of all ones. All active stations receive a frame with the broadcast address.

An Ethernet address consists of six bytes, labeled A, B, C, D, E, and F. The NI3010A transmits them in that order. It transmits the least significant bit of each byte first.

Bytes A, B, and C have been assigned by Xerox. Interlan has assigned bytes D, E, and F.

Byte A's least significant bit distinguishes a physical address from a multicast address. This bit is 0 for a physical address and 1 for a multicast address. Note that a physical address has an even number in byte A, and a multicast address has an odd number in byte A.

#### 4.4.2 The Source Address

The source address is always 48 bits long. These bits contain the physical address of the station that sent the frame. When transmitting a frame on the Ethernet, the NI3010A automatically inserts the source address.

#### 4.4.3 The Type Field

The type field is always 16 bits long. The user must specify the type field. The NI3010A requires that all frames have a type field.

By specifying the type field, the user can designate a higher level protocol used in the data field.

#### 4.4.4 The Data Field

The data field can range from 46 to 1500 bytes. The NI3010A accepts a data field of less than 46 bytes from the host, but it pads such a field to 46 bytes with null characters. A null character is a byte, each bit of which is 0.

The other fields of an Ethernet frame total 18 bytes. Hence, the minimum size of a frame is 64 bytes, and the maximum size is 1518 bytes. The NI3010A refuses to transmit a frame that exceeds 1518 bytes. It returns a status code of 05H, indicating that the host supplied an over-sized frame.

#### 4.4.5 The Frame Check Sequence

The frame check sequence (FCS) contains a 32-bit cyclic redundancy check (CRC) value. The NI3010A includes the destination address, the source address, the type, and the data fields (along with any inserted zeros) in its CRC calculation.

The NM10A contains shared CRC logic circuits. During transmission, the NM10A calculates the CRC value on the outgoing bit stream and appends this value to the frame's data field. During reception, the NM10A uses the same logic to calculate the CRC value on the incoming bit stream and compares this value to that the frame's FCS.

Appendix C of the Ethernet Specification, Version 1.0 is a detailed

description of CRC implementation.

#### 4.5 THE NI3010A TRANSMIT PROCESS

The NI3010A transmit process consists of obtaining data packets from host MULTIBUS memory, forming them into Ethernet frames, and successfully delivering them to the network.

The NI3010A performs the specified Ethernet data link and physical layer functions required to form and transmit a frame at 10 Megabits per second.

##### 4.5.1 The Transmit Data DMA Transfer

The host sets up a transmit packet in its own MULTIBUS memory. Here is what happens when a transmit packet goes from MULTIBUS memory to the NM10A.

1. The host writes an interrupt code of zero to the interrupt enable register on the MIB. Writing this register clears the NI3010A's interrupt line.  

Note: This step ensures that the DMA controller does not start a DMA transfer as soon as the byte count registers contain a non-zero value.
2. The host writes a 24-bit MULTIBUS memory address into the NI3010A's bus address registers.
3. The host writes the packet's byte count into the NI3010A's byte count registers.
4. The host initiates a DMA transfer by writing the interrupt enable register with an interrupt code of 6. The NI3010A will now interrupt the host processor when it completes the DMA transfer.
5. The NI3010A moves the transmit packet from host memory to its transmit FIFO. Each byte travels through the transmit data register.

After accepting each data byte, the DMA controller increments the address in the bus address registers and decrements the byte count in the byte count registers.

When the byte count reaches 0 and its transmit register is empty, the NI3010A interrupts the host processor. This is a transmit-DMA-done (TDD) interrupt. The transmit data are now stored in the transmit FIFO.

6. To transmit these data on the Ethernet, the host issues a Load transmit Data and Send command (29H). The NI3010A carries out the command, then loads the status register. The host reads the status register.

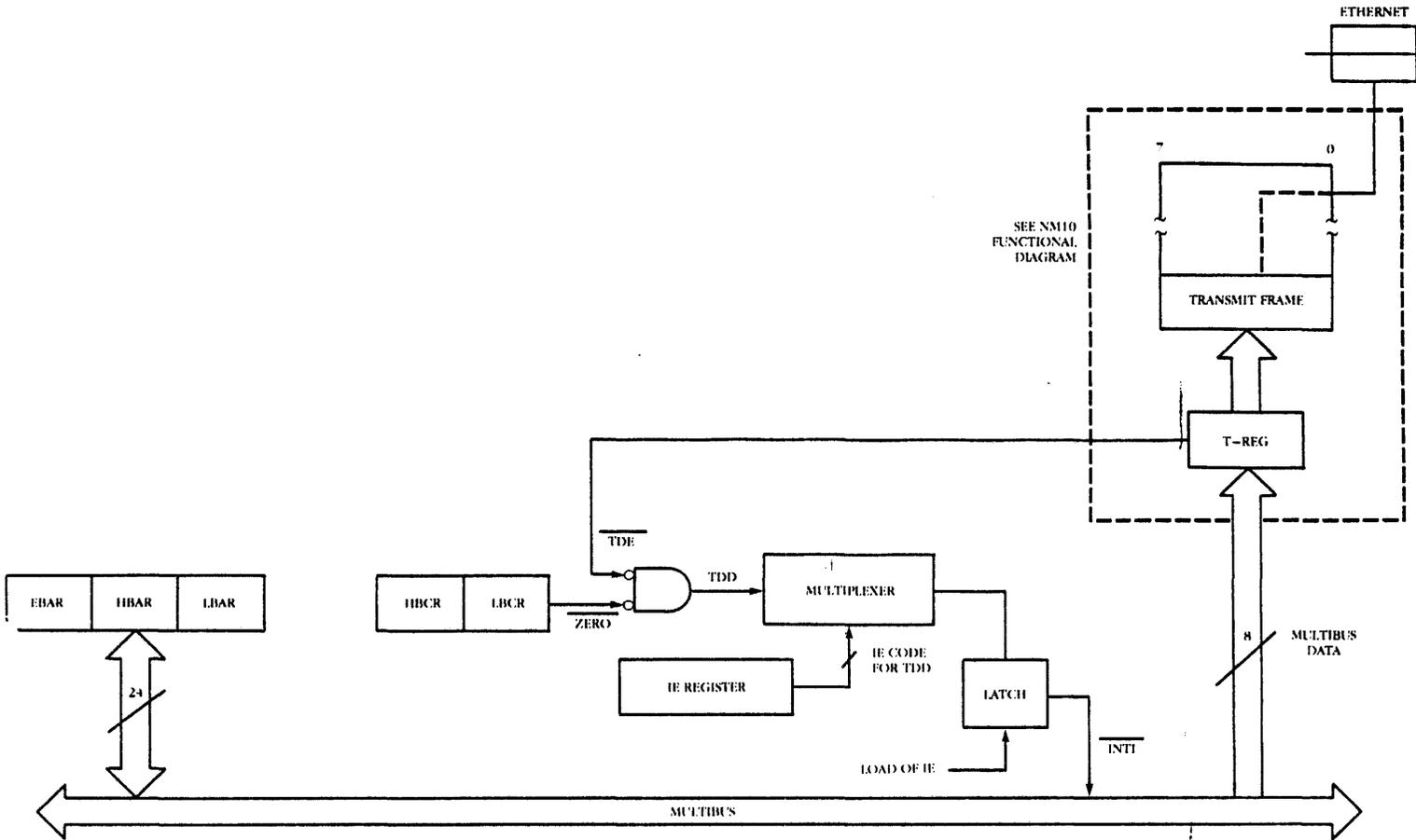


Figure 4-4. Functional Diagram of NI3010A Transmit DMA Operation

#### 4.5.2 Ethernet Frame Assembly

When the NM10A transmits data on the Ethernet, it assembles the data into an Ethernet frame.

- \* The NM10A ensures that the Ethernet is free of traffic for at least 9.6 microseconds before attempting to transmit a frame.
- \* The NM10A generates the 64-bit preamble.
- \* The NM10A inserts the NI3010A's physical address into the source field.
- \* The NM10A creates a self-synchronizing bit stream through Manchester encoding of the transmitted data.
- \* The NM10A generates a CRC value as the bit stream goes out on the Ethernet and inserts this CRC value into the frame's FCS field.

#### 4.5.3 Channel Access

The NI3010A performs carrier detection and deference, collision detection, and backoff with retransmission.

- \* Carrier detection and deference. The NI3010A monitors the physical channel and defers its transmission if the channel is busy with other traffic. When the channel becomes available, the NI3010A continues to defer its transmission for at least 9.6 additional microseconds, thus providing the proper interframe spacing.
- \* Collision detection. If another station transmits while the NI3010A is transmitting, a collision occurs. The NI3010A detects this collision.

When the NI3010A detects a collision, it jams. That means it stops transmitting data and transmits instead between 38 and 48 zeros before terminating transmission completely. The jam guarantees that all stations on the network detect the occurrence of the collision. The collision fragment is called a runt packet.

- \* Backoff with retransmission. When the NI3010A stops transmitting because of a collision, it waits a short period of time before attempting to transmit the frame again. The NI3010A determines this rescheduling of transmission by an Ethernet process called "truncated binary exponential backoff."

The NI3010A attempts to transmit a frame 16 times before reporting an error.

#### 4.6 THE NI3010A RECEIVE PROCESS

The NI3010A receive process consists of receiving data from the Ethernet into the receive FIFO, deciding whether to accept the data, formatting the data into packets, and transferring accepted packets to host MULTIBUS memory. The NI3010A deletes unaccepted frames from its receive FIFO.

The NI3010A performs the specified Ethernet data link and physical layer functions required to receive a frame from the network at 10 Megabits per second.

##### 4.6.1 Checking the Destination Address

When receiving a frame, the NI3010A synchronizes to and removes the preamble. Then, it separates (by Manchester decoding) the incoming bit stream into a receive data stream and a clock line. It runs the receive data stream into its receive FIFO.

If the NI3010A is in promiscuous mode, it accepts the frame no matter what its destination address is. If the NI3010A is not in promiscuous mode, it tests the destination address field and accepts the frame if the destination address matches

- \* The broadcast address
- \* The NI3010A's physical address
- \* One of the 63 multicast addresses that the user may have assigned to the NI3010A

The NI3010A translates a multicast address into a 6-bit pointer through the application of a many-to-few mapping technique called "hashing." This pointer identifies a location in the NI3010A's table of valid multicast addresses. The NI3010A accepts the frame only if its destination address matches the 48-bit multicast address stored in this location.

##### 4.6.2 Collision Fragment Filtering

Since collisions are a normal occurrence on the Ethernet, the NI3010A's CSMA/CD link management process filters out collision fragments by rejecting all received frames that are less than 64 bytes in length.

##### 4.6.3 Cyclic Redundancy Check

Finally, the NI3010A receives the FCS field and compares this value against its own CRC value. The user can set up the NI3010A to either accept or reject a frame with a CRC error.

#### 4.6.4 Receive Buffer Management

The NI3010A manages its receive FIFO as a ring buffer. Firmware manages the input pointer, the output pointer and the total byte count of received frames. When the NI3010A receives a frame from the Ethernet, it latches the frame's status and the frame's length.

Once the frame is in the receive FIFO, the on-board microprocessor reads the latched status and length bits, calculates a new byte count for the receive FIFO, and stores the frame's status bits along with the new byte count in a received packet status table. If the FIFO now contains insufficient space to receive a maximum-sized frame (1518 bytes), the microprocessor disables receiver hardware.

Each frame stored in the receive FIFO takes up exactly its size in bytes. The maximum number of frames that the FIFO can hold is 177. For this many frames to fit into the FIFO, the first 176 of them must be minimum-sized frames.

Occupied space in the buffer becomes available when the NI3010A transfers data to host MULTIBUS memory.

#### 4.6.5 The Receive Data DMA Transfer

Here is what happens when a receive packet goes from the NI3010A's receive FIFO to MULTIBUS memory.

1. The host issues an interrupt code of 4. This enables a receive-block-available interrupt from the NI3010A.
2. The host gets a receive-block-available interrupt. The host now knows that the NI3010A's receive FIFO has a frame awaiting transfer.
3. The host writes an interrupt code of 0 to the MIB's interrupt enable register. Writing this register clears the NI3010A's interrupt line.  

Note: This step ensures that the DMA controller does not start a DMA transfer as soon as the byte count registers contain a non-zero value.
4. The host writes a 24-bit MULTIBUS memory address into the NI3010A's bus address registers.
5. The host writes the byte count of its MULTIBUS buffer into the byte count registers.
6. The host initiates a DMA transfer. It does this by issuing an interrupt code of 7. This also enables a receive-DMA-done interrupt (RDD) from the NI3010A.
7. The NI3010A moves the received frame from its receive FIFO to host memory. The NM10A precedes the packet with a frame status byte, a null byte, and two bytes containing the frame's byte length. Each byte travels through the receive data register.

After transferring each data byte, the DMA controller increments the address in the bus address registers and decrements the byte count in the byte count registers.

The NI3010A generates a receive-DMA-done (RDD) interrupt when it finishes transferring the frame or when the byte count reaches zero.

8. The host responds to the RDD interrupt by issuing an interrupt code of zero. If the host exhausted its MULTIBUS buffer before accepting all of the NI3010A's receive data, the host must allocate another buffer and initiate another DMA transfer.

#### 4.6.6 Buffer Chaining

It's the host's responsibility to allow for the possibility of overflowing one of its MULTIBUS buffers. If the host chains buffers, it must keep track of their order. It's possible that an Ethernet frame may overlap host buffers. The host knows when this occurs because it knows the length of its own buffers and the length of each Ethernet frame. The host knows the length of each Ethernet frame from the frame length bytes.

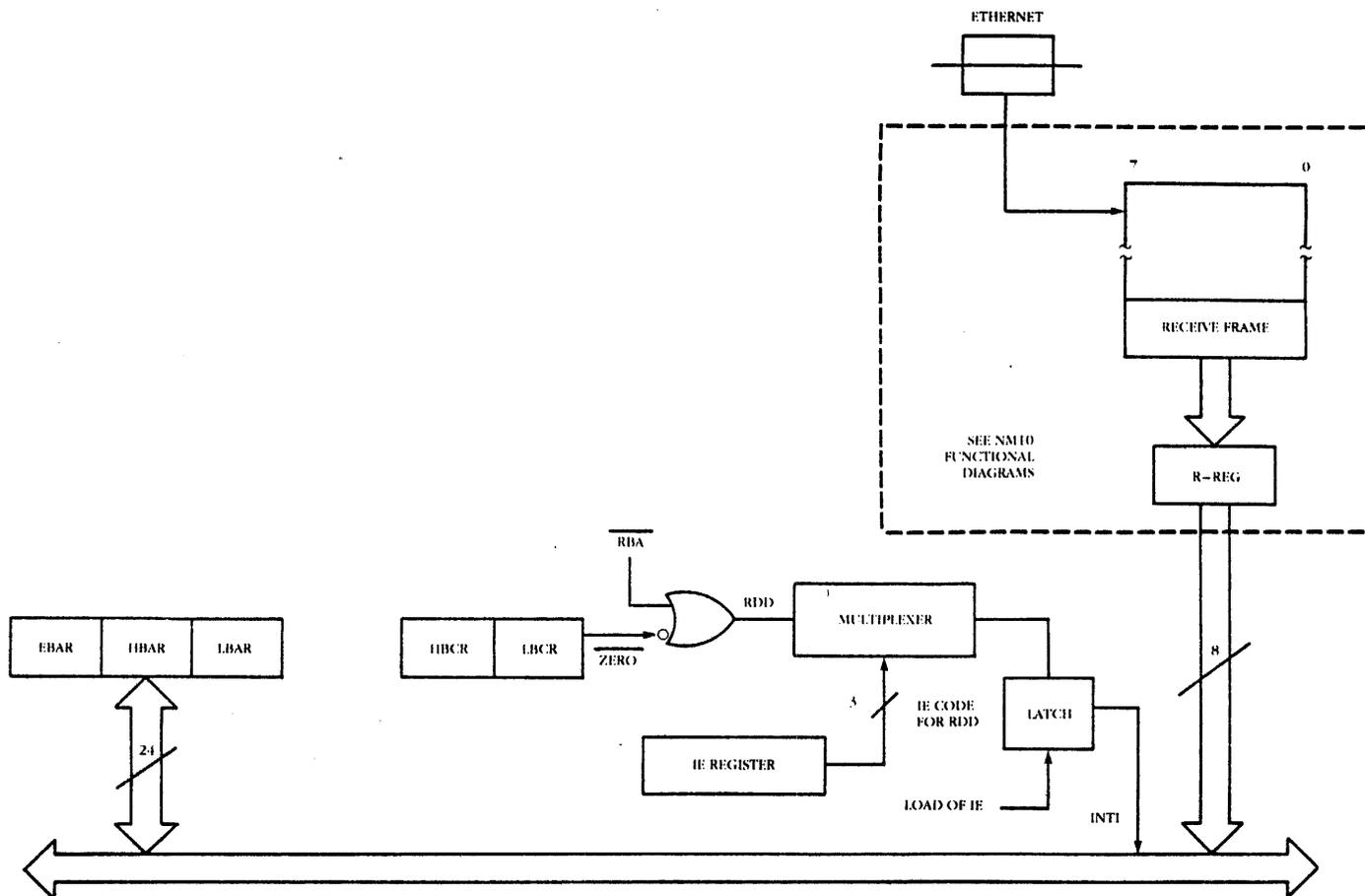


Figure 4-5. Functional Diagram of NI3010A Receive DMA Operation

#### 4.7 DIAGNOSTIC TOOLS

The NI3010A provides you with two sets of diagnostic tools.

- \* Three types of loopback operation. These are module interface loopback, internal loopback, and external loopback.
- \* Power-up self-tests with a pass/fail LED

Interlan has designed these diagnostic tools to help you isolate a network problem to a field replaceable component.

##### 4.7.1 Module Interface Loopback

The NI3010A enters this mode when the host issues a Set Module Interface Loopback command (01H). See the description of this command in Chapter 3 for more information about module interface loopback.

The host issues the Set Module Interface Loopback command (01H). Then, the host sets up a test data packet in its own memory and initiates a DMA transfer to the NI3010A. These data go into the NI3010A's transmit FIFO.

When the host issues a Load Transmit Data and Send command (29H), the data in the transmit FIFO appear as receive data. The first byte (the status byte) enters the receive data register, awaiting DMA transfer back to the host. When the host initiates a receive DMA transfer, the data enter the host's buffer.

The data appear the same as data that were received from the Ethernet. Although the data do not pass through the CRC generator, the frame does arrive at the host with four CRC bytes. The NI3010A, however, only adds these bytes to produce the correct frame length. They do not contain valid information.

##### 4.7.2 Internal Loopback

The NI3010A enters this mode when the host issues a Set Internal Loopback command (02H). See the description of this command in Chapter 3 for more information on internal loopback.

The host issues the Set Internal Loopback command (02H). Then, the host sets up a test data packet in its own memory and initiates a DMA transfer to the NI3010A. These data go into the NI3010A's transmit FIFO.

When the host issues a Load Transmit Data and Send command (29H), the data pass through the NI3010A's transmit hardware and address recognition code before entering the receive FIFO. All of the NI3010A's transmit and receive memories are required for internal loopback. If the frame's destination address matches the NI3010A's physical address, one of its multicast addresses, or the broadcast address the first byte of the

received frame (the status byte) enters the receive data register. Only frames addressed to the particular board will be accepted.

The data appear the same as data that were received from the Ethernet. Although the data do not pass through the CRC generator, the frame does arrive at the host with four CRC bytes. The NI3010A, however, only adds these bytes to produce the correct frame length. They do not contain valid information.

While in this mode, the NI3010A processes all commands normally. The only exception is that it doesn't send transmitted frames out on the Ethernet.

#### 4.7.3 External Loopback

The third loopback operation involves the transmission of an Ethernet frame over the network to the transmitting station. That is, the NI3010A 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 NI3010A 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 NI3010A 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.

All of the following occurs:

- \* Backoff on collision detect
- \* Aborting a transmission after 16 attempts
- \* Checking the frame's status

#### 4.7.4 The NI3010A Power-Up Self-Test

The NI3010A contains onboard (ROM resident) diagnostic programs. The NI3010A runs these programs

- when a hard reset occurs on the INIT line
- when the host issues a Reset command (3FH)
- when the host issues a Run Onboard Diagnostics command (0AH)

A pass/fail LED indicates if the NI3010A passed the test. This LED goes off if all the tests are successful.

See Section 5.2 for details about the onboard diagnostic programs.

## CHAPTER FIVE

### MAINTENANCE

The NI3010A does not require calibration or any special maintenance procedures. You can install it in a standard MULTIBUS slot, connect it to standard Ethernet transceivers and cables, and put it online. Interlan has provided the NM10A with onboard diagnostic programs to assist you if a problem develops.

#### 5.1 VERIFYING PROPER OPERATION

The pass/fail LED is normally off. When the tests run successfully, you'll see the pass/fail LED turn on and then off.

#### 5.2 THE NM10A ONBOARD DIAGNOSTICS

The NM10A onboard diagnostics programs are stored in ROM. The NM10A runs these diagnostics programs

- when it experiences a hard reset
- when the host issues a Rest command (3FH)
- when the host issues a Run Onboard Diagnostics command (0AH)

##### 5.2.1 After a Hard Reset or a Reset Command (3FH)

The NM10A

- \* Calculates a checksum on its ROM and compares this value to a checksum stored in the last two ROM bytes.
- \* Writes and then reads four pre-determined data patterns to its scratchpad RAM.
- \* Reads the pre-programmed Ethernet address and checks its validity.
- \* Transmits two test packets in Internal loopback mode to verify the transmit, receive and CRC logic.

If any of these tests fail, the NM10A reports a diagnostic status code as described in the Run Onboard Diagnostic command. The pass/fail LED is on only if one of these tests has failed.

### 5.2.2 After a Run Onboard Diagnostics Command (OAH)

The NM10A performs a set of tests designed to check out its state of health. While performing these tests, the NM10A maintains a status value. Failure of any test will cause the NM10A to exit the command, loading the status register with the appropriate status, see Run Onboard Diagnostic command. The host can then read the status register and interpret the value as a diagnostic status code.

Since the command does not initialize the board if a failure occurs, a Reset command should be issued before attempting any accesses. The pass/fail LED will remain on only if a failure occurs.

After successfully completing its test sequence, the NM10A re-initializes all states and variables and loads the status register with 00H. The host can then read the status register.

The pass/fail LED goes on at the beginning of these tests. If any test fails, the pass/fail LED stays on.

These tests consist of the following steps:

1. The NI3010A goes offline.
2. The pass/fail LED turns on.
3. The NM10A calculates a checksum on its ROM and compares this value to a checksum stored in the last two ROM bytes.
  - If the test succeeds, the NM10A goes to the next step, step 4.
  - If the test fails, the NM10A sets the status value to 02H and goes to step 9.
4. The NM10A writes and then reads four pre-determined data patterns to its transmit and receive RAM.
  - If the test succeeds, the NM10A goes to the next step, step 5.
  - If the test fails, the NM10A sets the status value to 03H and goes to step 9.
5. The NM10A reads the pre-programmed Ethernet address and checks its validity.
  - If the test succeeds, the NM10A goes to the next step, step 6.
  - If the test fails, the NM10A sets the status value to 04H and goes to step 9.
6. The NM10A tests its ability to send and receive two

pre-determined Ethernet packets through internal loopback.

If the test succeeds, the NM10A sets its status value to 00H and goes to the next step, step 7.

If the test fails, the NM10A sets its status value as defined in the Perform Network Loopback command and goes to step 9.

7. The NM10A initializes all its variables and data structures.

Caution. While performing this step, the NM10A loses all state information concerning data, statistics, etc.

8. If the status value is 00H (SUCCESS), the NM10A turns off the pass/fail LED.
9. The NM10A loads the status value into the status register and sets the SRF bit in the interrupt enable register to 1.

### 5.3 SUGGESTIONS FOR VERIFYING FURTHER NI3010A OPERATIONS

Interlan tests each NI3010A for correct operation in the factory. You may find the following suggestions useful when writing tests to verify all NI3010A features function properly.

1	RESET	Reset Test
2	REGTST	Power-Up Condition
3	PIOTST	Bus Addressing
4	INTTST	Verify all non-DMA interrupts
4	TDMATST	Verify transmit DMA transfer and interrupt
5	RDMATST	Verify receive DMA transfer and interrupt
6	CMDTST	Responds to Simple Commands
7	STATST	Checks NM10A Statistics
8	MLBTST	DMA Transfers (loopback)
9	ILBTST	DMA Transfer/NM10A Functions (loopback)
10	NLBTST	Perform Network Loopback Test On Line
11	LTDST	NM10A Verify of Load Transmit Data Cmd
12	CHNTST	Tests Non-Contiguous Buffering
13	NMMEM	Check NM10A 13.5K Receive FIFO
14	NMEMADD	Check NM10A 13.5K Receive FIFO addressing
15	CRCTST	Check CRC generator
16	NETADD	Check E'net address decoding
17	ADDTST	Check promiscuous mode operation

### 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, Massachusetts 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 States excludes all costs of shipping, Customs clearance, and other related 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.

## 5.6 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 package.
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 accommodate 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.

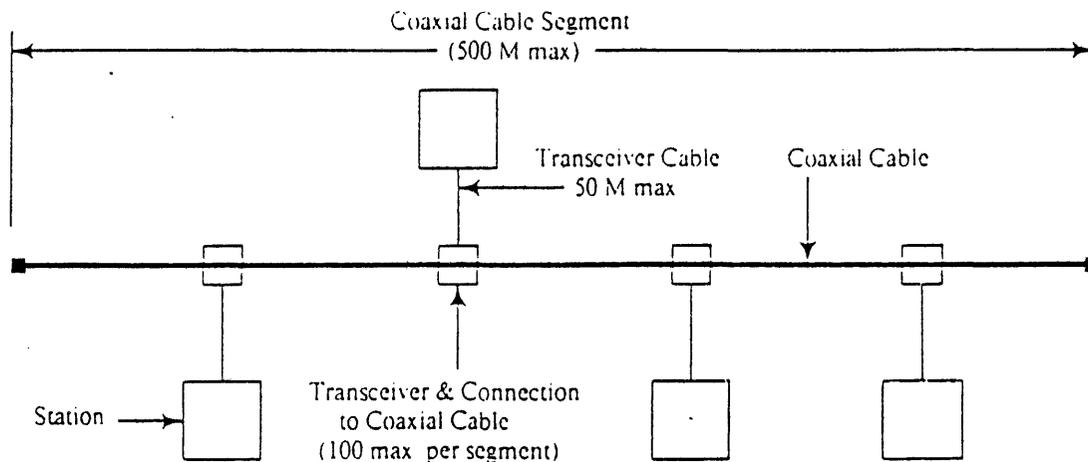


Figure A-1a: The Minimal Configuration

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.

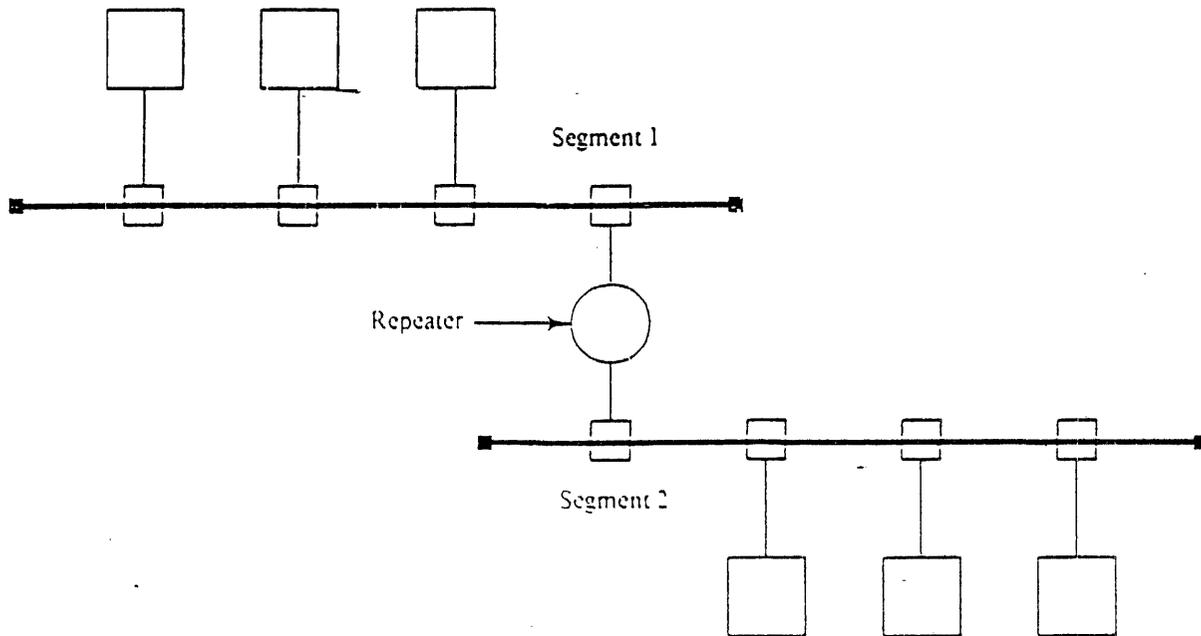


Figure A-1b. A Typical Medium-Scale Configuration

### 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).

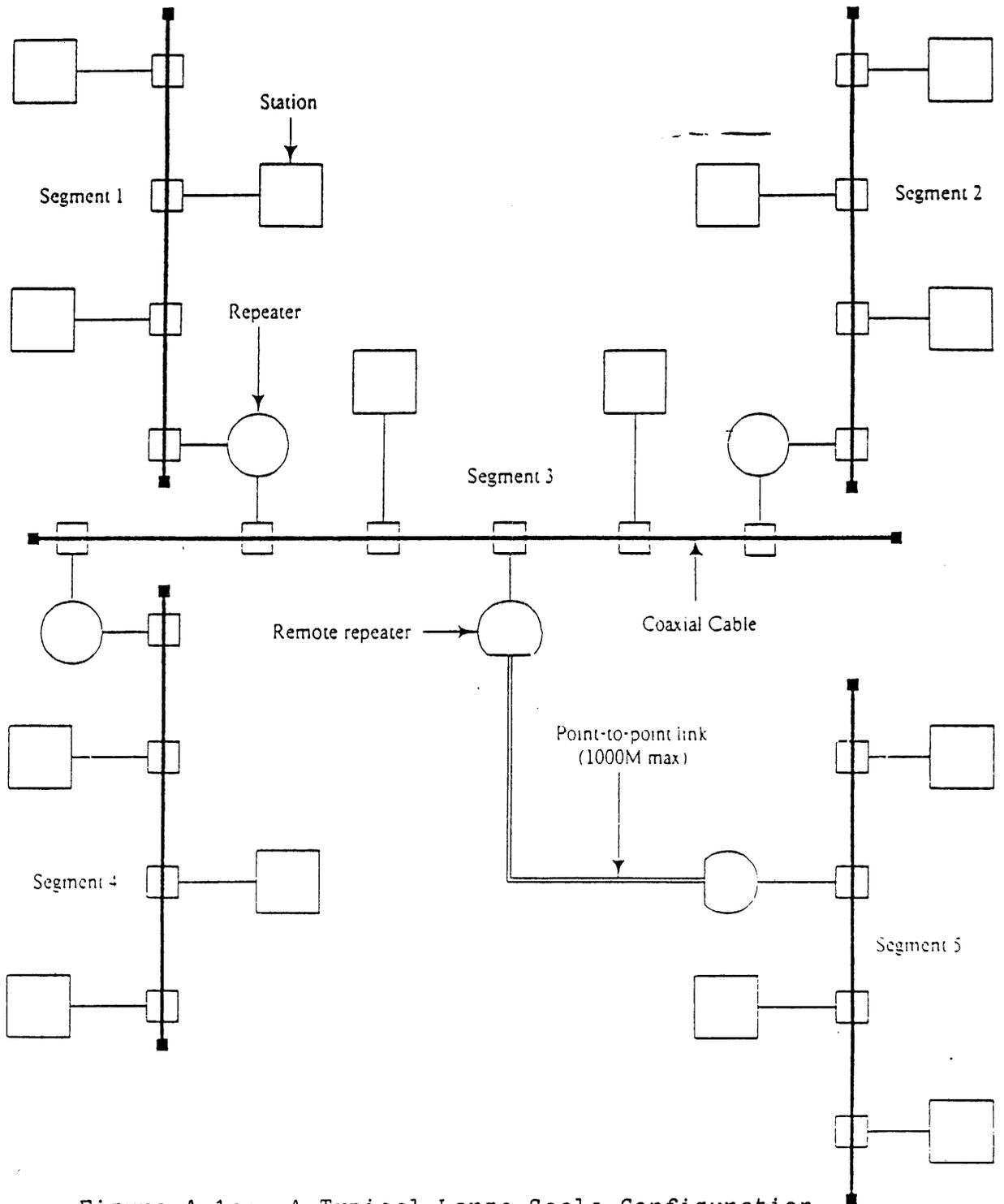


Figure A-1c: A Typical Large-Scale Configuration

### 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 energy 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.