Personal Computer
Hardware Reference
Library

Technical Reference

PC Network

6322916

IBM

Personal Computer Hardware Reference Library

Technical Reference

PC Network

First edition (September 1984)

FEDERAL COMMUNICATIONS COMMISSION RADIO FREQUENCY INTERFERENCE STATEMENT

Warning: The equipment described herein has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to the computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception. If peripherals not offered by IBM are used with the equipment, it is suggested to use shielded grounded cables with in-line filters if necessary.

INSTRUCTIONS TO USER

This equipment generates and uses radio frequency energy and if not installed and used properly, i.e., in strict accordance with the operating instructions, reference manuals, and the service manual, may cause interference to radio or television reception. It has been tested and found to comply with the limits for a Class B computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a residential installation.

If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna.
- Relocate the equipment with respect to the receiver.
- Move the equipment away from the receiver.
- Plug the equipment into a different outlet so that equipment and receiver are on different branch circuits.
- Ensure that all cables and connecting hardware are properly installed.
- If peripherals not offered by IBM are used with this equipment, it is suggested that you use shielded, grounded cables with in-line filters, if necessary.

If necessary, consult your dealer service representative for additional suggestions.

The manufacturer is not responsible for any radio or TV interference caused by unauthorized modifications to this equipment. It is the responsibility of the user to correct such interference.

CAUTION
The product described herein is equipped with a grounded plug for the user's safety. It is to be used in conjunction with a properly grounded receptacle to avoid electrical shock.

About This Book

The *IBM PC Network Technical Reference* provides network design information on the components that comprise the IBM PC Network, and information on the fundamentals of designing an IBM PC Network. This book also includes the IBM PC Network Adapter Basic Input Output System (BIOS) interface with listings.

This book is written for engineers, programmers, telecommunications professionals, and those interested in how the IBM PC Network is designed.

How This Book is Organized

This book has four chapters and five appendixes:

Chapter 1 is an introduction to the IBM PC Network:

- What the IBM PC Network is
- What the IBM PC Network does for you

Chapter 2 is a detailed reference about the software for the IBM PC Network adapter:

- Network software layers
- IBM PC Network software Network Control Block (NCB) commands

Chapter 3 is a detailed description of each of the IBM PC Network components:

- PC Network Adapter
- Translator Unit
- PC Network Cable System

Chapter 4 discusses the fundamentals of network design:

- How to calculate network parameters to obtain the desired signal levels
- Configuration considerations
- Available tools

Appendix A. IBM PC Network adapter and The IBM Translator Unit schematics

Appendix B. Tables of IBM PC Network specifications

Appendix C. The IBM PC Network protocols and pseudo code for the NCB commands

Appendix D. Adapter BIOS listings

Appendix E. Multitasking considerations

Glossary of Terms

Bibliography

IBM Personal Computer program license agreements permit the use of a program on a single machine. The customer is responsible for ensuring that each system user in the network is appropriately licensed to use any programs shared over the network.

Contents

Chapter 1. The IBM PC Network

Contents

What the IBM PC Network Is

The IBM PC Network is a broadband local area network that allows multiple Personal Computers to communicate with each other.

Some of the major features of the IBM PC Network are as follows:

- Many types of IBM Personal Computers can be connected to the network with an IBM PC Network Adapter in each computer.
- Each IBM PC Network Adapter is a highly intelligent device that can communicate on a single coaxial cable at a data rate of 2 million bits per second.
- The circuitry on the adapter includes an Intel™ 80188 microprocessor, an Intel 82586 communications controller, and other related circuitry.
- The IBM PC Network cable kit components can be used to simplify the installation of the IBM PC Network. A fully extended network using IBM components supports data transmissions for up to 72 nodes within a 1000 foot radius.
- You can choose to design your own data and video cable network. The network can contain up to 1000 nodes at a maximum distance of up to 5 kilometers with the proper cable specifications and a commercially available frequency translator. The performance of a large network depends on the load that is placed on the network by each computer.
- Communication with the IBM PC Network Adapter is through a BIOS interface that is operating system independent.

- The high level BIOS interface and a separate processor off-loads work from each computer on the network. This high level interface improves the computers performance, reduces memory requirements, and simplifies programming.

- The adapter BIOS supports a Remote Program Load feature to boot a Personal Computer from a remote server computer on the network.

- The software on the adapter is implemented as protocol layers. The software represents a broadband implementation of the lower 5 protocol layers (physical through session).

- Communication can occur by addressing other computers on the network by using ordinary names like John or Mary instead of physical addresses.

Broadband

The term broadband is used here to describe a type of local area network. Broadband networks are similar to Cable TV (CATV) networks in use today. One difference between CATV and Broadband networks is that CATV systems only transmit many one-way frequencies at the same time on a single coaxial cable. Broadband networks use different frequencies to provide a simultaneous communication link between each attached computer.

Major components used in CATV and broadband networks are similar, such as the coaxial cable, directional taps, and splitters. The difference is in the main component used in both types of networks called the frequency translator. This component does what its name implies; it translates from one frequency to another but leaves the information that is carried by the frequency intact. Most of the CATV frequency translators in use today do the same thing, but not in a bi-directional way like the IBM PC Network.

The frequency translator used in the IBM PC Network is made up of a frequency translator, a directional tap, and an 8-way splitter. These components, when connected together, make up what is called the IBM Translator Unit.

Local Area Network

This term is used to denote the physical size of the network. CATV networks, in general, cover part or all of an entire city. Broadband networks are limited in size to cover a cluster of computers within a building or even between buildings, if the cable limitations are not exceeded.

For the IBM PC Network, each computer connects to the network by a single coaxial cable connection similar to that for CATV. The physical layout of the cable can vary from installation to installation. You can think of the basic layout as a tree, with the frequency translator at the base of the tree. The main cable is the trunk of the tree A. The computers are connected to the main cable through branches B. Each computer on the network is referred to as a node C. Each node includes a personal computer, an IBM PC Network Adapter, and the necessary software. See Figure 1-1 for a diagram of a basic network.

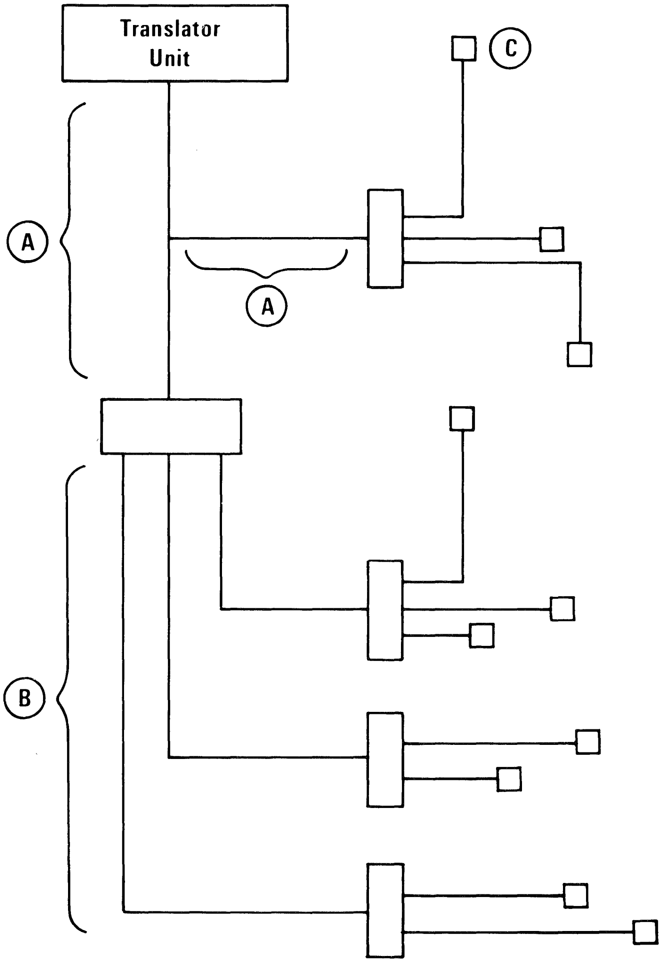


Figure 1-1 Basic Network Layout

The advantage of a tree network is that it continues to operate even though one or more of the nodes or branches are not active.

In general, the IBM PC Network is a reliable, efficient means to communicate programs, data, and messages between two or more computers. The following is a brief description of the software and hardware components that make up the IBM PC Network.

Basic Components

Basic components of the IBM PC Network are:

- An IBM PC Network Adapter and BIOS with a white adapter cable
- An IBM Translator Unit with connection hardware
- The IBM Base Expander
- An IBM Short, Medium, or Long Distance Kit
- IBM Coaxial Cable in lengths of 25, 50, 100, or 200 feet

The IBM PC Network Adapter

Each computer on the network must have an IBM PC Network Adapter. The adapter contains a radio frequency (RF) modem, a Basic Input Output System (BIOS) ROM, and two processors (Intel 80188 and 82586), to ensure reliable two-way communication with the other computers on the network. The hardware portion of the adapter is described in Chapter 3. In Chapter 2, the BIOS and programming features of the IBM PC Network Adapter are detailed. Some of the information covered in Chapter 2 is as follows:

- A layered approach to the network software

- Session and datagram support to connect the computers together
- Sample programs and aids to programming with the BIOS

The IBM Translator Unit

All transmissions pass through an electronic device called a Translator Unit. The unit receives transmissions from the devices attached to the network, amplifies them, raises their frequency into a higher range, and retransmits them over the same cable.

Each IBM PC Network must have one Translator Unit. The connection hardware that comes with the IBM Translator Unit supports up to eight computers. Each computer can be located up to 200 feet away from the Translator Unit's eight-way splitter.

The IBM Base Expander

Use this device when you need to connect more than eight computers to the network or when a distance greater than 200 feet between the Translator Unit and a computer is required. The Base Expander allows you to connect any combination of Short, Medium, or Long Distance Kits to the Translator Unit. Up to eight kits can be connected to the Base Expander.

The IBM Short, Medium, or Long Distance Kits

These kits allow you to connect up to eight additional computers to the network. Each kit must connect to the Base Expander to function properly on the network.

There are three types of kits for the IBM PC Network: Short, Medium, or Long. Each type of kit is selected for the distance you need to locate each computer from the Translator Unit.

- The **Short Distance Kit** connects directly to the Base Expander. Cable can be added between the Short Distance Kit and the computer for a distance of up to 200 feet.

- The **Medium Distance Kit** attaches to the Base Expander through 400 feet of cable. This cable must be added between this kit and the Base Expander for proper operation. Cable can be placed between the Medium Distance Kit and the computer for a distance of up to 200 more feet. The maximum allowable distance for each computer using this kit is 600 feet from the Translator Unit.

- The **Long Distance Kit** connects to the Base Expander like the Medium Distance Kit. 800 feet of cable must be added between this kit and the Base Expander for proper operation. Cable can be added between the Long Distance Kit and the computer for a distance of up to 200 additional feet. The maximum allowable distance for each computer using the Long Distance Kit is 1000 feet from the Translator Unit.

Hardware Configuration

The minimum IBM PC Network configuration consists of two computers and one IBM PC Network Translator Unit. Both computers contain an IBM PC Network adapter, and a 3–meter white adapter cable that connects them to the Translator Unit.

To expand this simple system, you can add up to 6 more computers. For more distance or more than 8 computers, you can add IBM cable, the IBM Base

Expander, and any combination of IBM Short, Medium, or Long Distance Kits. Also, you must add an IBM PC Network Adapter for each computer.

You can expand the IBM PC Network, with the kits, up to a maximum of 72 nodes by installing the hardware needed to connect them. See "The IBM PC Network Cable System" on page 3-79 for details about using the cable kits.

Because the IBM PC Network is a broadband network, you can simultaneously use the network for other applications.

The IBM Translator Unit can only be used on a network that transmits and receives data. It does not allow or support any of the following functions. To support these functions, you must acquire a different frequency translator.

- Additional data channels
- Video conferencing
- Closed circuit TV (CCTV)
- Area and building access control
- Security and fire alarm systems
- Energy management and conservation

Notes:

Notes:

Chapter 2. IBM PC Network Software Description

Contents

Introduction

The IBM PC Network is a broadband local area network designed to logically and physically connect two or more Personal Computers together. The software on the adapter presents a high level interface to the programmer eliminating the need to know network protocol details. This high level interface improves system performance by off-loading network programs onto the adapter. Also, this off-loading feature saves memory because the network programs are in the adapter's memory and not in the Personal Computer's memory. Concepts that are designed within the network are as follows:

- Peer network—This means that each member is treated equally and on a first-come, first-served basis. There is no "host" concept as in telecommunication operations. There are no required centralized facilities of any type on the network other than the Translator Unit. Peers on the network can be connected with a reliable, point-to-point connection called a virtual circuit.

- Names on the network—When each member is physically connected to the network, a name is given to represent that peer. The names used can be general names, such as "John", instead of specialized names or numbers. Names can also be clustered into logical groups.

- Session services—After the names for each peer are specified, two of the peers communicate with each other in a mode called a session. Sessions are very similar to a telecommunication reliable point-to-point, full-duplex type of connection. For the IBM PC Network, a session can also be referred to as a virtual circuit. Once the session is established, the transfer of data through the network can begin.

- Datagram service—The IBM PC Network also supports messages called datagrams. Datagram services do not provide a point-to-point connection. The datagrams are only sent once. Acknowledgment and retransmissions are the responsibility of the user.

IBM PC Network Adapter Data Transfer

This section describes the IBM PC Network Adapter software. The adapter supports all network and software protocol functions to assure that messages and data are sent from one computer to another on the network. It also provides the mechanism for returning command status to the Personal Computer following command execution.

The IBM PC Network adapter supports five layers of the data transfer protocols. Each layer comprises one or more protocol services. Each layer communicates only with the layer immediately above and below it. This structure allows a modular design of the protocols. The layers supported by the adapter are as follows:

- Physical layer
- Link layer
- Network layer
- Transport layer
- Session layer

The physical layer is implemented using the RF modem on the adapter and the interface logic to the Intel 82586 Communications Controller. For detailed information on the physical layer, refer to the Intel 82586 iAPX manual and refer to Chapter 3 for the RF Modem logic.

The link layer is primarily implemented in hardware by using the Intel 82586.

The other three layers (network, transport and session) are implemented using the Intel 80188 processor and ROMs on the adapter. Also, the layers provide a reliable virtual connection service, a name support facility, and a low-overhead datagram service.

The following figure provides a general overview of the protocol services provided by the Physical, Link, Network, Transport, and Session layers.

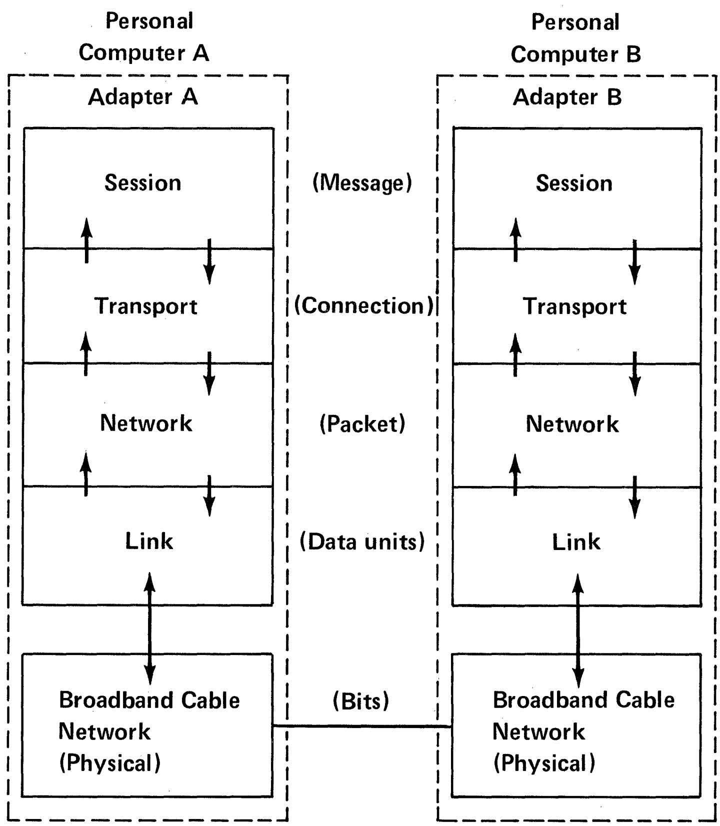


Figure 2-1. Adapter Software Layers

The following description provides a general overview of the protocol services provided by the Physical, Link, Network, Transport, and Session layers. Detailed specifications and standard message formats can be found in Appendix C of this book.

Physical Layer

The physical layer is implemented in hardware providing a 2 Mbit per second physical channel on the broadband network through a single-channel RF modem. This layer is mostly implemented using the RF transmit and receive circuits along with the Sytek Serial Interface Controller (SIC). The RF modem transmits on one channel and receives on another.

Link Layer

The link layer uses a connection-less oriented link layer protocol. This link layer protocol is similar in function to the IEEE 802.3 medium access control protocol.

The link layer is largely responsible for assembling the bits, transmitted by the network layer or received from the physical layer, into data units. When the physical layer has received a transmission of bits from the cable, it is the responsibility of the link layer to check and assemble the bits.

When the network layer has a packet to transmit, the link layer has the responsibility to put the data into the correct format and detects any errors that might occur. The data units are further organized by either adding or removing any necessary start or end bytes. A final CRC check is performed on the organized bits.

CSMA/CD

One method of regulating transmissions is called a channel arbitration protocol. One such protocol is called Carrier Sense, Multiple Access with Collision Detection or CSMA/CD. CSMA/CD operates in the following way:

The carrier sense multiple access with collision detection (CSMA/CD) technique is used to resolve contentions and allows the sharing of the common channel on the broadband cable in an orderly and equitable manner.

Carrier Sense

- Each adapter continuously monitors traffic on the channel, even when transmitting.

Multiple Access

- Anytime there is a pause on the channel (no one transmitting), any device can begin transmitting.
- If only one device begins transmitting during a pause, that device gains control of the channel and transmits its messages without interruptions.
- After that transmission is ended, the channel is clear again, and all devices waiting to transmit again contend for the channel.

Collision Detection

- If two or more devices begin transmitting during the same pause, their signals collide and the garbled data is detected by both receivers (collision detection).
- When the collision is detected, the devices stop transmitting and wait for some random time interval before retransmitting. During the time

interval before retransmitting, any other device can attempt to transmit over the network, using the CSMA/CD protocol.

The IBM PC Network uses the CSMA/CD method for all data transmissions on the network.

Network Layer

The network layer has the responsibility of correctly routing the packets. When a message is passed from the transport layer, this layer selects the correct routing convention for the message. The packet is then passed on to the link layer for processing.

When this layer has received a data unit from the link layer, the Network layer determines if the packet is a datagram or if it belongs to a virtual connection and passes the packet up to the transport layer for further processing.

Transport Layer

The transport layer primarily has the responsibility of creating a reliable point-to-point connection between two adapters on the network. This layer supports data transmissions and acknowledgments, and handles any necessary flow control or pacing required to maintain a reliable virtual circuit. This layer transmits messages for the session layer using services of the network layer.

Session Layer

The session layer presents the adapter interface to the network for all Personal Computer programs. Responsibilities of this layer include establishing a session using two names in the appropriate name tables and interpreting commands in the form of a Network Control Blocks (NCBs). The concept of an adapter being known by many names is implemented in this layer.

Chapter 2 outlines the specific interface to this layer.

Programming The IBM PC Network Adapter

The first section, "IBM PC Network Adapter Characteristics", describes basic concepts of how to program your adapter using the IBM PC Network Adapter Basic Input Output System (BIOS) and its interface to the IBM Personal Computer. Even though the network software is composed of many layers, the BIOS presents one interface to the user program. When programming the adapter, you should use the BIOS interface.

The next section, "Network Control Block (NCB)", describes how to interface your program with the BIOS by using a collection of fields to form a NCB. Once a command is presented to the BIOS in the NCB format, a response is returned to the program in the form of a return code. These return codes are also described in this section.

The last section, "Remote Program Load for the IBM PC Network", describes how to program and setup a Personal Computer as a program load computer for the network. The section also has instructions for the sample programs included with this book.

Before you program, you need to understand some additional characteristics of the adapter.

IBM PC Network Adapter Characteristics

All of the communication functions from the physical layer through the session layer are handled on the adapter.

The BIOS is a software interface between the IBM PC Network adapter and the Personal Computer programs. The BIOS places the unique features of a local area network into a standard format.

IBM PC Network security is not built into the BIOS. Instead, it is the responsibility of the operating system or application program to make sure that data or devices are secure on the IBM PC Network.

Data Transfer

Two basic types of data transfer are supported. Reliable data transfer is provided by the session layer. If data is lost or if the line drops, the BIOS will return an error code.

Data transfer using datagram support goes directly to the link layer. This type of transfer does not contain any features such as those found in the session or the transport layer. The most common use of this type of data transfer is for broadcast messages.

Name Support

You must communicate on the network by using names. Each adapter can hold up to 16 selectable names and 1 permanent node name. Each name has a length of 16 characters and all 16 characters are always used in a name. A permanent name is always present and consists of 10 bytes of binary zeros followed by the unique adapter unit ID number. The next 16 names can be added to the name table.

Using the Network

To use the network you must:

1. Add your name to the table of names on the adapter. This is the name that you are known by on the network. Skip this step if you wish to use the permanent node name.

2. Establish a session with another name on the network. This gives you a logical connection with another name. The other name can be in your name table or in a name table of another adapter.

3. Send and receive messages using that session.

As an alternative to session support, you can use datagram support.

Network Control Block (NCB)

This section describes how to create an NCB, how to handle interrupts, and how to recover from error situations.

> **Note:** The following section assumes that you are familiar with assembler language and its concepts.

Commands are presented to the BIOS in the form of a Network Control Block (NCB). The following is the basic concept and format to present NCB commands to the BIOS:

1. Build and fill in all required fields of an NCB.

   When you build a new NCB, set all fields to binary zeros. See section "NCB Field Description" on page 2-15 for the required fill character for each field.

2. Allocate any necessary buffers specified in the required NCB fields.

3. Make sure that there are at least 20 bytes of stack space left for each outstanding NCB command.

4. Place the address of the NCB in the ES:BX register pair. Issue a software interrupt to vector 5C hex (i.e. INT 5CH).

5. Once an NCB is issued, *Do Not* change or move it until it has completed.

6. After the command is processed, control is returned to the caller. The result of the process is in either the AL register or the return code field of the NCB.

NCB Field Description

The following is a list of the NCB fields and the description of each field. See the following figure for the correct format of the fields.

Note: An "@" is used to represent the word "address" in the following list of descriptions.

Network Control Block (NCB) Format

Field Name	Coding and Meaning	
NCB__COMMAND	DB 00H	; NCB command ; field
NCB__RETCODE	DB 00H	; NCB return code ; field
NCB__LSN	DB 00H	; NCB local session ; number field
NCB__NUM	DB 00H	; NCB number of ; your name
NCB__BUFFER@	DD 00000000H	; NCB pointer to ; message buffer ; address ; (offset:segment)
NCB__LENGTH	DW 0000H	; NCB buffer length ; (in bytes)
NCB__CALLNAME	DB 16 DUP(0)	; NCB name on local ; or remote adapter. ; For CHAIN SEND, ; the first 2 bytes ; indicates length ; of second buffer. ; The next 4 bytes ; indicates the second ; buffer address.

Figure 2-2 (Part 1 of 2). Network Control Block Format

Field Name	Coding and Meaning	
NCB_NAME	DB 16 DUP(0)	; NCB name on local ; adapter.
NCB_RTO	DB 00H	; NCB receive timeout ; value
NCB_STO	DB 00H	; NCB send timeout ; value
NCB_POST@	DD 00000000H	; NCB pointer to post ; routine ; (offset:segment)
NCB_LANA_ NUM	DB 00H	; NCB adapter number ; for first adapter. ; Use 01H for second ; adapter.
NCB_CMD_ CPLT	DB 00H	; NCB command ; status field ;
NCB_RESERVE	DB 14 DUP (0)	; NCB reserved area

Figure 2-2 (Part 2 of 2) Network Control Block Format

NCB_COMMAND

A 1-byte field for the command code to execute. Each command can be executed in either a wait or no-wait mode. If the high order bit is set to 1, the no-wait option is selected. If the high order bit is is set to 0, the wait option is selected. The remaining 7 bits are used to specify the command that you want the adapter to execute.

For maximum throughput between the IBM PC Network Adapter and the Personal computer, the no-wait option is preferable. In addition, the no-wait option allows multiple commands to be queued for execution within the adapter.

The programming interface to the IBM PC Network Adapter BIOS is different depending upon the selection of the wait/no-wait option. For example; when issuing the instruction INT 5CH using the wait option, control is not returned to the next instruction until the adapter completes the command. When the command does complete, check either the AL register or the NCB__RETCODE field for the status of the completed command.

If you choose the no-wait option for the INT 5CH instruction, you will receive 2 return codes. One code is returned immediately after you issue the instruction. The following is a list of the possible return codes that can be found in the AL register:

Immediate Return Codes

- 00H—Good return
- 03H—Invalid command
- 21H—Interface busy
- 22H—Too many commands outstanding
- 23H—Invalid number in NCB_LANA_NUM field
- 24H—Command completed while cancel occurring
- 26H—Command not valid to cancel
- 4XH—Unusual network condition
- (50–FE)H—Adapter malfunction

If the immediate return code is 00H, a final return code is posted to the user when the adapter has executed the command. The posting of the return code can be done by having the adapter interrupt the user application program or by the user application program checking the NCB__CMD__CPLT field. If the NCB__POST@ field is non-zero, the adapter interrupts the user application program at the address specified in the NCB__POST@ field. If the NCB__POST@ field is zero, the adapter does not interrupt the program and command completion must be determined by checking the NCB__CMD__CPLT field.

When the adapter interrupts the user application program upon command completion, the final return code can be obtained from either the AL register or the NCB__RETCODE field. If checking the

NCB_CMD_CPLT field, a change in value from FFH (pending status) indicates command completion. This value represents the final return code. The final return code varies from command to command.

If the immediate return code is other than 00H, the adapter cannot execute the requested command and adapter processing terminates. See "Error Recovery" on page 2-90 for the definitions and the recommended actions for each return code.

NCB_RETCODE

A 1-byte field indicating the return code of a command. If the return code is 00H, the operation was successful. Any other number means that the operation failed or has not completed. If the no-wait option is used without being interrupted on command completion, the NCB_CMD_CPLT field not the NCB_RETCODE field contains the final return code. When a command has not completed, the return code is FFH. See section "Error Recovery" on page 2-90 in this chapter for the definitions and the recommended actions for each return code.

When a command is completed, your routine is interrupted by the adapter at the post address. You can choose to check the AL register instead of the return code field.

Never go into a program loop on the NCB_RETCODE field looking for a command to complete. Loop on the NCB_CMD_CPLT field for this purpose.

NCB_LSN

A 1-byte field indicating the local session number. This is the number of the session you have with another name on the network. This is only valid after a CALL or LISTEN command has successfully completed. For

SEND and RECEIVE commands under session support, this field must always be correctly filled in. For datagram support, the LSN does not apply.

The NCB_LSN field is assigned a number in a round-robin Modulo 254 technique ranging from 1 to 254. 00H and FFH are never returned.

The RESET command uses this field to indicate the maximum number of sessions supported.

NCB_NUM

A 1-byte field indicating the number returned to you after an ADD NAME, ADD GROUP NAME command is executed. This number, not the name, must be used with all datagram support commands and and for RECEIVE ANY commands.

The NCB_NUM field is assigned a number in a round-robin Modulo 255 technique ranging from 2 to 254. The permanent node name number is always 1. 00H and FFH are never retuned.

The RESET command uses this field to indicate the maximum number of command blocks to be supported.

NCB_BUFFER@

> **Note:** An "@" is used to represent the word "address" in the following list of descriptions.

A 4-byte field indicating a pointer to the buffer you wish to use with a command. This field is in Define Double-Word format (offset:segment) and must be a valid address in memory. See the *IBM Macro Assembler* manual for references to define double-word format (DD).

NCB_LENGTH

A 2-byte field indicating the length, in bytes, of the data you want transferred.

For a RECEIVE, RECEIVE ANY, ADAPTER STATUS, SESSION STATUS, RECEIVE BROADCAST DATAGRAM, and RECEIVE DATAGRAM command, this field is used and updated to indicate the number of bytes that are actually received. For a SEND, CHAIN SEND, SEND DATAGRAM, and SEND BROADCAST DATAGRAM command, this field is used to indicate the number of bytes to be sent.

NCB_CALLNAME

A 16-byte field indicating the name with whom you want to communicate. All 16 bytes are used. The name can be either on your adapter or any other adapter.

For a CHAIN SEND command, the first 6 bytes are used to specify the second buffer. In these 6 bytes, the first 2 bytes specify the length and the last 4 bytes specify the buffer address. These are specified in the same format as the NCB_LENGTH and NCB_BUFFER@ fields. The remaining bytes in this field are reserved.

NCB_NAME

A 16-byte field indicating the name that you are known by on the network. All 16 bytes must always be used. The table on the adapter can hold up to 16 names. You are always known by the permanent node name on the network. The permanent node name is 10 bytes of binary zeros, followed by 6 bytes returned by the ADAPTER STATUS command.

NCB_RTO

A 1-byte field used by the CALL and LISTEN commands to specify a time-out period for all RECEIVES associated with that session. The receive time-out is the maximum amount of time allowed before a RECEIVE command returns a time-out error. The time-out value is specified in increments of 500 ms. If binary zero (00H) are specified, the default is no time out. The time-outs can also be different for each session, but are fixed once the session is established.

NCB_STO

1-byte field used by the CALL and LISTEN commands to signify a send time-out. The send time-out is the maximum amount of time allowed before a SEND command returns a time-out error. The time-out is specified in increments of 500 ms. SEND time-outs should be used with caution because they will always drop the session if they expire. If binary zero (00H) are specified, the default value is no time-out.

NCB_POST@

A 4-byte field indicating the address of the routine that is to be executed when the adapter has completed processing a command. This field is used in no-wait options only. This field is in Define Double-Word format (offset:segment) and must be a valid address in memory. Your post routine must establish DS and any other registers you need. Only AL, CS, ES, and BX registers are set for the NCB being completed. The post routine is called by the adapter interrupt level with interrupts masked. To return, issue an IRET. Your post routine should be short and return immediately, unless you unmask interrupts.

If the post address is specified as all binary zero, a post does not occur. This allows a program to do other work and then loop waiting for the NCB_CMD_CPLT field to see when it changes from FFH. When the change from FFH occurs, either the command completed or an error code is returned. Do not check the NCB_RETCODE because it will change before the command is actually complete. This can be useful in a BASIC program by using PEEK and POKE.

NCB_LANA_NUM

A 1-byte field indicating which adapter you want to use. A value of 00H directs the command block to the first IBM PC Network adapter. A value of 01H directs the command block to the second IBM PC Network adapter.

NCB_CMD_CPLT

A 1-byte field indicating the command status. A value of 0FFH in this field indicates the command is pending. A value of 00H means that the command is complete. Any other value means that the command completed with an error. See section "Error Recovery" on page 2-90 for a complete description of the error codes.

This byte is only useful if the NCB_POST@ was specified as all zero's and you are using a no-wait option command. Otherwise, check the NCB_RETCODE field.

NCB_RESERVE

A 14-byte reserved field. This space should be allocated because the BIOS uses this field to store temporary variables.

NCB Commands

NCB commands control the adapter on the network. The commands are divided into four categories:

- General
- Name support
- Session support
- Datagram support

Within each category, all of the commands, except for the RESET and CANCEL commands, are further divided into wait and no-wait options. The wait option means that when you issue the command, the processor waits until the command is completed before returning to the next instruction. The no-wait option means that the processor returns immediately after processing the command and is interrupted at the post address when the command is completed.

You must fill in either all required parameters or use the default values for some of the commands in the Network Control Block (NCB). Command codes and return codes are represented by hexadecimal values.

Wait Option

When you use the wait option, check either the NCB__RETCODE field or the AL register for the return code. See Appendix E for multitasking considerations.

No-Wait Option

When using the no-wait option, the issued command returns to the next instruction after the INT 5C. The

AL register should be checked for a good return code 00H. Refer to the description of each command issued for any other values returned.

If the command was accepted with AL = 00H, an interrupt will occur when the command is completed. Either the AL register or the NCB_RETCODE may be checked for the return value. If the command is accepted but not complete, the NCB_RETCODE field should contain a FFH.

Your program should handle one special case of the no-wait option. The no-wait post is on an interrupt level with interrupts masked. It is possible to get the command complete interrupt before your main routine has finished processing the accepted command.

Be sure to specify each NCB_COMMAND field correctly or you will receive a 03H return code.

General Commands

General commands are used to enable your adapter on the network, to read status, or to control other outstanding commands.

RESET

Reset local adapter status and clear the name and session tables. This command allows you to change the number of sessions and NCB command blocks supported by the adapter. You can specify a value from 1 to 32 for sessions and a value from 1 to 32 for NCB command blocks. At power-on time, the default values are 6 sessions and 12 commands. Session and NCB command blocks take space away from the data buffers on the adapter and reduce the packet size on the network. For best performance, only configure the number of sessions and commands that you actually need. If the specified values exceed the limits, the maximum values are used. If binary zeros are specified, the default values are used.

Once a RESET is completed, the ADAPTER STATUS command can be used to see the resulting maximum data packet size allowed by the adapter. Since overall performance is related to the number of packets sent on the network, you can choose to optimize your message size to fit into one packet. If two adapters are reset to different command and session sizes, the resulting packet sent on the network will always be the smaller of the two.

This command does not reset traffic and error statistics. Only a power-on reset will reset these statistics.

Cmd code	32H—Wait for the command to be completed.
Fields Required	NCB__LSN (Number of sessions to be supported.)

	NCB_NUM (Number of command blocks to be supported.) NCB_LANA_NUM (Number of the adapter you want to reset.)
Field Returned	NCB_RETCODE

Return Codes:

Final Return Codes

00H—Good return

03H—Invalid command

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

CANCEL

This command requests that the command, whose NCB can be found at the address given by NCB_BUFFER@, be canceled.

Commands that are not valid to CANCEL are: ADD NAME, ADD GROUP NAME, DELETE NAME, SEND DATAGRAM, SEND BROADCAST DATAGRAM, SESSION STATUS, RESET, and CANCEL. Use caution when canceling a SEND command because completing it will always drop the session.

Cmd Code	35H—Wait for the command to be completed.
Fields Required	NCB_LANA_NUM (Number of the adapter you want to cancel.) NCB_BUFFER@ (Address of the NCB you want canceled.)
Field Returned	NCB_RETCODE

Return Codes:

Final Return Codes

00H—Good return

03H—Invalid command

23H—Invalid number in NCB_LANA_NUM field

24H—Command completed while cancel occurring

26H—Command not valid to cancel

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

ADAPTER STATUS

Receive adapter status. This command gives the status information for a local or remote adapter by specifying the name in the NCB__CALLNAME field. If an * is specified in the first byte of the NCB__CALLNAME field, the information for the local adapter is returned. This information is placed in the specified buffer address, NCB__BUFFER@, and the length field is updated to indicate the number of bytes of information received.

The minimum number of bytes in the status buffer is 60 bytes. The maximum number of bytes required to hold the status buffer is 348 bytes when 16 names are in use. In general, 60 + 18(X) = the number of bytes that are required, where X is the number of names in use.

> **Note:** A return code of 06H is posted in the NCB__RETCODE field if the receive buffer is not large enough for the data. The remaining data is lost at this point.

Cmd code	33H—Wait for the command to be completed. B3H—Return immediately and post when the command is completed.
Fields Required	NCB__BUFFER@ NCB__LENGTH NCB__CALLNAME (local, remote, or an * for local) NCB__POST@ (If no-wait option used) NCB__LANA__NUM (Number of the adapter for the status.)
Field Returned	NCB__RETCODE

Data that is returned in the data buffer field contains the following information.

* Unit identification number—6 bytes

This number is part of the permanent node name. The unit identification number is represented as follows:

Byte 0: Low word, low byte
Byte 1: Low word, high byte
Byte 2: Middle word, low byte
Byte 3: Middle word, high byte
Byte 4: High word, low byte
Byte 5: High word, high byte

* External jumper status—1 byte

The status of the external jumpers are represented as follows:

7	6	5	4	3	2	1	0
W2	W1	X	X	X	X	X	X

Where:
X = Reserved

W2 = 1 Jumper W2 on adapter
W2 = 0 Jumper W2 off adapter

W1 = 1 Jumper W1 on adapter
W1 = 0 Jumper W1 off adapter

Figure 2-3 Jumper Status Byte

* Results of last self-test—1 byte

See "Power-On Self-Tests (POST)" on page 3-63 for a description of the tests.

* Software version—2 bytes

The software version of the protocol layers are represented as follows:

Byte0: Major version number
Byte1: Minor version number

* Traffic and error statistics—48 bytes

1. Duration of reporting period (in minutes)—2 bytes After the counter reaches a value of 0FFFFH, it will roll over to 0.

2. Quantity of CRC errors received—2 bytes After the counter reaches a value of 0FFFFH, they will not increment further.[1]

3. Quantity of alignment errors received—2 bytes After the counter reaches a value of 0FFFFH, they will not increment further.[1]

4. Quantity of collisions encountered—2 bytes After the counter reaches a value of 0FFFFH, it will roll over to 0.

5. Quantity of aborted transmissions—2 bytes A transmission can be aborted due to excessive collisions or for some other cause.[1] After the counter reaches a value of 0FFFFH, it will roll over to 0.

6. Number of successfully transmitted packets—4 bytes After the counter reaches a value of 0FFFFFFFFH, it will roll over to 0.

---

1 This is supplied by the Intel 82586 chip. See the Intel 82586 Reference Manual, "System Control Block" section for further information.

7. Number of successfully received packets—4 bytes After the counter reaches a value of 0FFFFFFFFH, it will roll over to 0.

8. Number of retransmissions—2 bytes After the counter reaches a value of 0FFFFH, it will roll over to 0.

9. Number of times the receiver exhausted its resources—2 bytes After the counters reach a value of 0FFFFH, they do not increment further.[1]

*** Adapter resource statistics**

1. Reserved for internal use—8 bytes

2. Free command blocks—2 bytes

3. Configured maximum NCBs—2 bytes

4. Maximum number of command blocks free command blocks—2 bytes

5. Reserved for internal use—4 bytes

6. Pending sessions—2 bytes A pending session is either a CALL-pending, a LISTEN-pending, a session established, session aborted, HANG UP-pending, or HANG UP (complete).

7. Configured maximum pending sessions—2 bytes

8. Total maximum of possible sessions—2 bytes

9. Maximum session data packet size—2 bytes

*** Quantity of names in the local name table—2 bytes**

*** Local name table—16 entries of 18 bytes each**

The first 16 bytes of each entry represent the name, and the last 2 bytes represent the name status. This first byte is equal to the name number. The second byte denotes the status when it is masked with an 87H. The mask is used to get the most significant bits and the last 3 bits of the byte. The other bits are reserved and can have nonzero values.

- NXXXX000 = Trying to register a name
- NXXXX100 = A registered name
- NXXXX101 = A de-registered name
- NXXXX110 = A detected duplicate name
- NXXXX111 = A detected duplicate name with de-register pending

  Where:

  - X = Reserved bit
  - N = 0 The name is a unique name
  - N = 1 The name is a group name

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in `NCB_LANA_NUM` field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

01H—Illegal buffer length

05H—Command timed out

06H—Message incomplete

0BH—Command canceled

19H—Name conflict detected

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

UNLINK

This command is only used with the remote program load (RPL) feature. The command applies only if a call to IBMNETBOOT was made at power up time of this computer. The session with IBMNETBOOT is dropped when this command is issued. The BIOS also ends the INT 13 redirector to the network. For more information refer to the Remote Program Load section in this book.

Cmd code	70H—Wait for the command to be completed.
Fields Required	NCB_LANA_NUM (Number of the adapter you want to unlink.)
Field Returned	NCB_RETCODE

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

Name Support

Name support commands allow your personal computer to be known by a name on the network. A name can be a unique name or a group name on the network. The adapter checks to see if a name is unique on an ADD NAME and returns an error if anyone else is using the name you want to add. When using an ADD GROUP NAME, the same name can be added by many adapters on the network.

The adapter can have up to 16 names in the local name table. A permanent node name is always present and consists of 10 bytes of binary zero followed by the unique adapter unit ID number. This permanent node name is also unique on the network. You can find its value by issuing an ADAPTER STATUS NCB for a local status by putting an asterisk (*) in the callname field. Look at the first 6 bytes returned in the buffer specified. Append this number to 10 bytes of binary zeros to make a total of 16 bytes for the permanent node name. This permanent name does not show up as an entry in the local name table returned by the ADAPTER STATUS NCB command.

The RESET command deletes all names from the specified adapter with the exception of the permanent node name.

Reserved Names

The following names are reserved and cannot be added or deleted:

- Any name starting with an * in ASCII or 00H.

- It is recommended that you should not use any name starting with IBM.

ADD NAME

Add a 16-character name to the table of names. The name you add cannot be used by anyone else on the network.

When the adapter processes this command, it sends a broadcast request on the network repeatedly. If no reply is received, the name is assumed to be unique and is added to the table of names. The command returns to you the number of your name in the NCB__NUM field. This number is used in datagram support and for RECEIVE ANY commands.

Cmd Code	30H—Wait for the command to be completed. B0H—Return immediately and post when the command is completed.
Fields Required	NCB__NAME NCB__POST@ (If no-wait option used) NCB__LANA__NUM (Number of the adapter for the add name.)
Fields Returned	NCB__RETCODE NCB__NUM

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

0DH—Duplicate name in local name table

0EH—Name table is full

15H—Name not found, cannot specify an * , or 00H

16H—Name in use on remote adapter

19H—Name conflict detected

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

ADD GROUP NAME

Add a 16-character name to the table of names. The name you add cannot be used by anyone else on the network as a unique name but can be added by anyone as a group name.

When the adapter processes this command, it sends a broadcast request on the network repeatedly. If no unique name replies, the name is added. The command returns to you the number of the name in the NCB__NUM field. This number is used in datagram support and for RECEIVE ANY commands.

Cmd Code	36H—Wait for the command to be completed. B6H—Return immediately and post when the command is completed.
Fields Required	NCB__NAME NCB__POST@ (If no-wait option used) NCB__LANA__NUM (Number of the adapter for the add group name.)
Fields Returned	NCB__RETCODE NCB__NUM

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

25H—Reserved name specified

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

0DH—Duplicate name in local name table

0EH—Name table is full

15H—Name not found, cannot specify an * , or 00H

16H—Name in use on remote adapter

21H—Interface busy

22H—Too many commands outstanding

25H—Reserved name specified

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

DELETE NAME

Delete a 16-character name from the table of names kept in the adapter. You should use the HANG UP command before you delete the name. If the name has active sessions when this command is issued, the name is flagged as de-registered and the status, "command completed, name has active sessions" is returned to the user. The DELETE delays until all sessions associated with the name are closed or abnormally terminated. If the name has only pending non-active session commands when the DELETE NAME command is issued, the name is removed and the "command completed" status is returned to the user. The pending non-active session commands are terminated immediately with the "name was deleted" status. Non-active session commands are: LISTEN, RECEIVE ANY, DATAGRAM RECEIVE, RECEIVE BROADCAST DATAGRAM.

A name flagged as de-registered continues to occupy an entry in the local name table until the de-registration is completed.

Cmd Code	31H—Wait for the command to be completed. B1H—Return immediately and post when the command is completed.
Fields Required	NCB__NAME NCB__POST@ (If no-wait option used) NCB__LANA__NUM (Number of the adapter for the delete name.)
Field Returned	NCB__RETCODE

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

0FH—Command completed, name has active sessions and is now de-registered

15H—Name not found, cannot specify an * , or 00H

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

Session Support

Session support commands allow you to establish a logical connection (session) on the network, send and receive messages, end sessions, and read session status. More than one command can be outstanding because the connection is in two-way simultaneous transmission mode.

Sessions are established between any two names on the network. These names can be on your adapter or any other adapter. Names are used to establish sessions, but a 1-byte number is used to refer to each session after they are established. This number is found in the NCB__LSN field that is returned when a session is established. A maximum of 32 sessions are allowed. The same name pair can be used to establish more than one session. The difference between the session pairs are the different LSN fields. If you create a local session, two session entries are used instead of one. One side of the local session has a LSN number associated with it and the other side has a different LSN number. To establish a session with yourself, make the CALLNAME equal to the NAME field.

Session support gives you reliable data transfer and receipt of a message. Messages can range from 0—65,535 characters in length. The RESET command aborts all sessions.

CALL

This command opens a session with another name specified by the NCB__CALLNAME field using the local name specified by the supplied NCB__NAME. The name that you call must have a LISTEN command outstanding for the session to be established. You can establish a session with either a local or a remote name. Multiple sessions can be established with the same pair of names. All SEND and RECEIVE commands for this session will abort if they are unsuccessful after the specified time-out intervals. The time-out intervals are specified in 500 millisecond units (a value of zero means that no time-out will occur). The CALL command will abort, if unsuccessful after the system time-out intervals. The system time-out intervals and retry count are constants in the adapter software. When the CALL is completed, a local session number (LSN) is assigned and used thereafter to refer to the established session.

Local session numbers (NCB__LSN) are assigned in a round-robin technique, starting from the next available value within the range of 1 to 254.

Cmd Code	10H—Wait for the command to be completed. 90H—Return immediately and post when the command is completed.
Fields Required	NCB__CALLNAME NCB__NAME NCB__RTO (Specified in 500 ms increments. If the field is set to 00H, no receive time-out occurs.) NCB__STO (Specified in 500 ms increments. If the field is set to 00H, no send time-out occurs.) NCB__POST@ (If no-wait option used)

	NCB_LANA_NUM (Number of the adapter you want to call.)
Fields Returned	NCB_RETCODE NCB_LSN

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

05H—Command timed-out

09H—No resource available

0BH—Command canceled

11H—Local session table full

12H—Session open rejected

14H—Cannot find name called or no answer

15H—Name not found, cannot specify an * , or 00H

18H—Session ended abnormally

19H—Name conflict detected

1AH—Incompatible remote device

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

LISTEN

This command enables a session to be established with the name specified in the NCB__CALLNAME field, using the name specified by the NCB__NAME field. If the CALLNAME field has a name starting with an * , a session is established with any network node that issues a CALL to the local name.

LISTEN for a specific name has priority over a LISTEN for any name. Sessions can be established with either a local or a remote name. Multiple sessions can be established with the same pair of names.

All SEND and RECEIVE commands for this session abort if they are unsuccessful after the specified time-out intervals. If a SEND times-out, the session is abnormally terminated.

The time-out intervals are specified in 500 millisecond units (a value of zero means that no time-out will occur). A LISTEN command does not time-out but, a LISTEN occupies a session entry and is considered a pending session in information returned on an adapter status command. Local session numbers (LSN) are assigned in a round-robin technique starting with the next available value within the range from 1 to 254. Also, if an * is used for the called name, the name that made the call will be returned in the CALLNAME field.

The error "Name conflict detected" is returned if, during the completion of a LISTEN command, a unique name exists in more than one table. All nodes with the name registered, except for the one where the LISTEN command has returned successfully, will report the error "Name conflict detected".

Cmd Code	11H—Wait for the command to be completed. Use this carefully because it does not

time out and your program hangs until the command is satisfied.
91H—Return immediately and post when the command is completed.

Fields Required

NCB__CALLNAME (This can be specified in the first byte as an * . The * is used to listen for a call from anyone to your name. If a name is specified in this field, it takes priority over a name of * .)
NCB__NAME
NCB__RTO (Specified in 500 ms increments. If the field is set to 00H, no receive time-out occurs.)
NCB__STO (Specified in 500 ms increments. If the field is set to 00H, no send time-out occurs.)
NCB__POST@ (If no-wait option used)
NCB__LANA__NUM (Number of the adapter you want to listen.)

Fields Returned

NCB__RETCODE
NCB__LSN
NCB__CALLNAME (If listen any used. Specified with an * .)

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

09H—No resource available

0BH—Command canceled

11H—Local session table full

15H—Name not found, cannot specify an * , or 00H

17H—Name was deleted

18H—Session ended abnormally

19H—Name conflict detected

1AH—Incompatible remote device

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in `NCB_LANA_NUM` field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

HANG UP

This command closes the session with another name on the network indicated by the local session number. A "Good return" status is returned on a normal close and a "Session closed" status or an illegal local session number is returned if the session is already closed or never existed.

When a HANG UP command is received, all pending local RECEIVE commands are terminated and returned to the issuer with "Session closed" in the NCB__RETCODE field. The termination is valid regardless of whether or not any data had been transferred by the pending command. If a local SEND command is pending, the HANG UP commands delays until the SEND is completed or until approximately 20 seconds have elapsed. This delay occurs whether or not the command has begun to transfer data or is waiting for the remote computer to issue a RECEIVE command. The HANG UP is performed if any of the following conditions occur:

- The SEND is completed
- The SEND has aborted
- The SEND fails because the session was terminated by the other computer with a HANG UP.
- The SEND fails because of the time-out specified when the session was opened.

If one of the above conditions does not occur within 20 seconds after the HANG UP command is executed, the HANG UP command is returned with a "Command timed-out" status and the session is aborted.

When a session closes, all SEND and RECEIVE commands pending on the closed session are returned to the issuer with a "Session closed" status. Also, if a RECEIVE ANY command is pending on the local name used by the session, it is returned to you with a "Session closed" status. Only a single RECEIVE ANY command is returned even though many RECEIVE ANY commands may be pending. Even though a single RECEIVE ANY command is returned, many SEND or RECEIVEs can be returned when pending.

When a session abnormally terminates, all outstanding commands on that session are returned to the issuer with a "Session ended abnormally" status.

Cmd Code	12H—Wait for the command to be completed. 92H—Return immediately and post when the command is completed.
Fields Required	NCB_LSN NCB_POST@ (If no-wait option used) NCB_LANA_NUM (Number of the adapter you want to hang up.)
Field Returned	NCB_RETCODE

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

05H—Command time-out

08H—Illegal local session number

0AH—Session closed

0BH—Command canceled

18H—Session ended abnormally

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

SEND

This command sends data by the session number indicated in the local session number (LSN). The data is taken from the buffer indicated by the NCB__BUFFER @ for the indicated number of bytes. The buffer is limited to a size starting with 0 and up to 65,535 bytes in length.

When a session is closed by the remote computer, all SEND commands pending on the closed session are returned with a "Session closed" status. If a local HANG UP command is issued with any pending SEND commands, the HANG UP is delayed until the SEND commands are completed.

If a session aborts, a "Session ended abnormally" status is returned. If the SEND time-out expires, the session is aborted and a "Command timed-out" status is returned. Time-out values for the SEND are associated with the session when a CALL or LISTEN was issued and cannot be specified here.

If more than one SEND is pending, the data is transmitted in a first-in, first-out (FIFO) order within a session.

If the SEND cannot complete for any reason, the session ends abnormally and the session is dropped. The reason for this is to guarantee data integrity.

SEND commands without corresponding RECEIVEs, consume resources on the adapter. It is not advisable to issue many SENDs without corresponding RECEIVEs.

Cmd Code	14H—Wait for the command to be completed. 94H—Return immediately and post when the command is completed.
Fields Required	NCB__LSN

	NCB_BUFFER@ NCB_LENGTH NCB_POST@ (If no-wait option used) NCB_LANA_NUM (Number of the adapter you want to send.)
Field returned	NCB_RETCODE

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

05H—Command timed-out

08H—Illegal local session number

0AH—Session closed

0BH—Command canceled

18H—Session ended abnormally

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in `NCB_LANA_NUM` field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

CHAIN SEND

This command sends data by the session number indicated in the local session number (LSN). The data is taken from the buffers for the indicated number of bytes. Two buffers can be chained together with this command.

The data in the second buffer is concatenated to the data in the first buffer and sent as a single message. The NCB__CALLNAME is used to specify the length and address of the second buffer. The length must be specified in the first 2 bytes and the second buffer address is the next four bytes.

When a session is closed by the remote computer, all CHAIN SEND commands pending on the closed session will be returned with a "Session closed" status. If a local HANG UP command is issued with any pending CHAIN SEND commands, the HANG UP is delayed until the SEND commands are completed.

If a session abnormally terminates, a "Session ended abnormally" status is returned. If the CHAIN SEND time-out expires, the session is aborted and a "Command timed-out" status is returned. Timeout values for the SEND are associated with the session when a CALL or LISTEN is issued.

Messages are limited to a size starting with 0 and up to 65,535 bytes in length.

If more than one CHAIN SEND is pending, the data is transmitted in a first-in, first-out (FIFO) order within a session.

Cmd Code	17H—Wait for the command to be completed. 97H—Return immediately and post when the command is completed.
Fields Required	NCB__LSN

NCB_BUFFER@
NCB_LENGTH
NCB_CALLNAME (The format for the second buffer is specified as follows:)
  1) NCB_LENGTH2
     DW 0000H
  2) NCB_BUFFER2@
     DD 00000000H
NCB_POST@ (If no-wait option used)
NCB_LANA_NUM (Number of the adapter you want to chain send.)

Field returned NCB_RETCODE

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

05H—Command timed-out

08H—Illegal local session number

0AH—Session closed

0BH—Command canceled

18H—Session ended abnormally

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in `NCB_LANA_NUM` field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

RECEIVE

Receive data from a specified session. If more than one RECEIVE command is outstanding, they are posted according to the following hierarchy: RECEIVE, RECEIVE ANY for a specified name, and RECEIVE ANY for any name. Once the commands are sorted according to hierarchy, all of the RECEIVE commands are processed in a first-in, first-out order. Time-out values for RECEIVE are specified during a CALL or LISTEN and cannot be specified here.

When a session is closed, either by a local session close command or by the remote adapter closing the session, all pending NCBs for that session are returned with a session closed status.

> **Note:** A return code of 06H is posted in the NCB__RETCODE field if the receive buffer is not large enough for the message being sent. You can issue another receive to obtain the rest of the information before a time-out occurs.

Cmd Code	15H—Wait for the command to be completed. 95H—Return immediately and post when the command is completed.
Fields Required	NCB__LSN NCB__BUFFER@ NCB__LENGTH NCB__POST@ (If no-wait option used) NCB__LANA__NUM (Number of the adapter you want to receive.)
Fields Returned	NCB__RETCODE NCB__LENGTH

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

05H—Command timed-out

06H—Message incomplete

08H—Illegal local session number

0AH—Session closed

0BH—Command canceled

18H—Session ended abnormally

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

RECEIVE ANY

Receive data from anyone with whom you have a session. You must use your name number instead of your name when issuing this command. Your name number (NCB__NUM) was returned when you issued the ADD NAME or ADD GROUP NAME command. If more than one RECEIVE command is outstanding, they are completed in a first-in, first-out (FIFO) order according to the following hierarchy: RECEIVE, RECEIVE ANY for a specified name, and RECEIVE ANY for any name.

If a session is closed by the local or remote computer, or session aborted, one RECEIVE ANY or RECEIVE name will be posted with "session closed" or "session aborted" status regardless of the number of session receives that can be pending. If a RECEIVE ANY or RECEIVE name is pending, it is posted as "Session closed" with the LSN field posting the session that closed. A RECEIVE ANY with no name specified is posted only if no RECEIVE ANY name is pending for the session with that name.

> **Note:** A return code of 06H is posted in the NCB__RETCODE field if the receive buffer is not large enough for the message being sent. You can issue another RECEIVE to obtain the rest of the information.
>
> Application programs should not use a RECEIVE ANY to any name because this command can receive messages from other programs running in the Personal Computer.

Cmd Code	16H—Wait for the command to be completed. Use this carefully because it does not time-out. 96H—Return immediately and post when the command is completed.

Fields Required	NCB_BUFFER@ NCB_LENGTH NCB_NUM (If this field = FFH, then receive from any remote name, that you have a session with, for any of your names.) NCB_POST@ (If no-wait option used) NCB_LANA_NUM (Number of the adapter you want to receive for any name.)
Fields Returned	NCB_LSN NCB_RETCODE NCB_NUM (If FFH is specified.) NCB_LENGTH

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

06H—Message incomplete

0AH—Session closed

0BH—Command canceled

13H—Illegal name number

17H—Name deleted

18H—Session ended abnormally

19H—Name conflict detected

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

SESSION STATUS

Receive status of all active sessions for your name. This command optionally gives the status for all of the names in the local name table if an * is specified in the first byte of the NCB__NAME field. The minimum buffer length possible is 4 bytes.

> **Note:** A return code of 06H is posted in the NCB__RETCODE field if the receive buffer is not large enough for the data being sent. The remaining data is lost at this point.

Cmd Code	34H—Wait for the command to be completed. B4H—Return immediately and post when the command is completed.
Fields Required	NCB__BUFFER@ NCB__LENGTH NCB__NAME (Specify an * for all names.) NCB__POST@ (If no-wait option used) NCB__LANA__NUM (Number of the adapter for session status.)
Fields Returned	NCB__RETCODE NCB__LENGTH

Data areas returned contain the following:

1. Name number of sessions being reported—1 byte

2. Number of sessions with this name—1 byte

3. Number of RECEIVE DATAGRAM and RECEIVE BROADCAST DATAGRAM commands outstanding—1 byte

4. Number of RECEIVE ANY commands outstanding—1 byte

5. Information that is returned about a session—36 bytes for each session

   a. Local session number—1 byte

   b. State of the session—1 byte

   This byte is represented as follows:

LISTEN pending	01H
CALL pending	02H
Session established	03H
HANG UP pending	04H
HANG UP complete	05H
Session Aborted	06H

   c. Local name—16 bytes

   d. Remote name—16 bytes

   e. Number of RECEIVE commands outstanding—1 byte

   f. Number of SEND and CHAIN SEND commands outstanding—1 byte

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

01H—Illegal buffer length

06H—Message incomplete

15H—Name not found, cannot specify an * , or 00H

19H—Name conflict detected

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in `NCB_LANA_NUM` field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

Datagram Support

Datagram support commands allow you to send a message to a name, a group name, or to broadcast a message to everyone. These commands also allow you to receive a datagram message from a name, a group name, or from anyone on the network. Datagram support differs from session support in several ways. The message is never acknowledged by the receiver's adapter, so it is up to the sender and receiver to agree on their own network protocols. Messages are limited in size starting with 0, up to 512 bytes in length. If you specify more than 512 bytes for a RECEIVE DATAGRAM or RECEIVE BROADCAST you will only receive the maximum that is allowed for a SEND DATAGRAM or SEND BROADCAST.

Datagrams are smaller than session SENDs and require additional protocol interaction for reliable data transmissions. For reliable transmissions, sessions should always be used.

SEND DATAGRAM

Send datagram to a unique name or group name for receipt at a local node or remote node.

Cmd Code	20H—Wait for the command to be completed. A0H—Return immediately and post when the command is completed.
Fields Required	NCB_BUFFER@ NCB_LENGTH NCB_NUM NCB_CALLNAME NCB_POST@ (If no-wait option used) NCB_LANA_NUM (Number of the adapter for the send datagram.)
Field Returned	NCB_RETCODE

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

01H—Illegal buffer length

13H—Illegal name number

19H—Name conflict detected

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in `NCB_LANA_NUM` field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

SEND BROADCAST DATAGRAM

Send a message to everyone who has a RECEIVE BROADCAST DATAGRAM command outstanding. If the remote adapter does not have a RECEIVE BROADCAST DATAGRAM outstanding, it does not get the message. If a computer issues a SEND BROADCAST DATAGRAM and the computer has a RECEIVE BROADCAST DATAGRAM command outstanding, the adapter receives its own message. If the adapter has several broadcast messages pending the next SEND BROADCAST command issued satisfies all RECEIVE BROADCAST commands.

Cmd Code	22H—Wait for the command to be completed. A2H—Return immediately and post when the command is completed.
Fields Required	NCB_BUFFER@ NCB_LENGTH NCB_NUM NCB_POST@ (If no-wait option used) NCB_LANA_NUM (Number of the adapter for the send broadcast datagram.)
Field Returned	NCB_RETCODE

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

01H—Illegal buffer length

13H—Illegal name number

19H—Name conflict detected

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

RECEIVE DATAGRAM

Receive a datagram message from any name or anyone on the network directed to you. There is no time-out associated with this command. If you do not have a RECEIVE DATAGRAM command outstanding at the time a SEND DATAGRAM is issued, you will lose data.

This command does not receive a broadcast datagram but will receive a group name.

> **Note:** A return code of 06H is posted in the NCB__RETCODE field if the receive buffer is not large enough for the data being sent. The remaining data is lost at this point.

Cmd Code	21H—Wait for the command to be completed. Use with care since all processing halts until the datagram is received. A1H—Return immediately and post when the command is completed.
Fields Required	NCB__BUFFER@ NCB__LENGTH NCB__NUM (If this field = FFH, then receive a datagram from any other name for any of your names.) NCB__POST@ (If no-wait option used) NCB__LANA__NUM (Number of the adapter for the receive datagram.)
Fields Returned	NCB__RETCODE NCB__LENGTH NCB__CALLNAME

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in `NCB_LANA_NUM` field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

06H—Message incomplete

0BH—Command canceled

13H—Illegal name number

17H—Name deleted

19H—Name conflict detected

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

RECEIVE BROADCAST DATAGRAM

Receive a message from anyone who issues a SEND BROADCAST DATAGRAM command. There is no time-out for this command.

> **Note:** A return code of 06H is posted in the NCB__RETCODE field if the receive buffer is not large enough for the data being sent. The remainder of the data is lost.

Cmd Code	23H—Wait for the command to be completed. Use with care since all processing halts until the datagram is received. A3H—Return immediately and post when the command is completed.
Fields Required	NCB__BUFFER@ NCB__LENGTH NCB__NUM NCB__POST@ (If no-wait option used) NCB__LANA__NUM (Number of the adapter for the receive broadcast datagram.)
Fields Returned	NCB__RETCODE NCB__LENGTH NCB__CALLNAME

Return Codes:

Immediate Return Codes

00H—Good return

03H—Invalid command

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Final Return Codes

00H—Good return

03H—Invalid command

06H—Message incomplete

0BH—Command canceled

13H—Illegal name number

17H—Name deleted

19H—Name conflict detected

21H—Interface busy

22H—Too many commands outstanding

23H—Invalid number in NCB_LANA_NUM field

4XH—Unusual network condition

(50–FE)H—Adapter malfunction

Note: The X can be any number.

Remote Program Load for the IBM PC Network

To remotely load an IBM Personal Computer on the IBM PC Network you must have a program written to service the Remote Program Load (RPL) request issued by the adapter and an IBM PC Network adapter that has jumper W1 removed. See "Configurable Hardware Options" on page 3-70 for a detailed description of the jumper settings. Included on the diskette are sample programs that can be used to service the remote program load request. See "Sample Programs" on page 2-82 for a description of the sample programs.

The BIOS on the IBM PC Network Adapter provides the capability to load a computer from the IBM PC Network. The IBM PC Network BIOS redirects the initial diskette read requests to the network if there are no other drives enabled and the RPL jumper on the adapter is removed. The boot request goes to a special IBM name on the network called IBMNETBOOT. The IBMNETBOOT name must be active and it must handle the RPL requests from remote IBM Personal Computers. This function does not depend on the operating system and can operate with any operating system that uses RPL during bootstrap. The only restriction is that the operating system must use only INT 13 requests and not try to use the diskette hardware directly.

RPL Request Format

If there are no drives ready on the Personal Computer and the RPL jumper is removed, the following requests are issued:

1. The BIOS issues a RESET command to allow the maximum number of sessions and commands.

2. A CALL to IBMNETBOOT is issued by the BIOS. If this step fails, the computer will come up in ROM BASIC.

3. If the call to IBMNETBIOS succeeds, the BIOS sends an 11-byte message to IBMNETBOOT with the following format:

Start					End
AX	CX	DX	ES	BX	Carry flag

Figure 2-4 Boot Record Format

> **Note:** The BIOS intercepts and reformats all INT 13 requests to the system. When the reformatting is done, the above format is used to send the information to IBMNETBOOT.

The BIOS will then issue a RECEIVE for an 11-byte reply message.

4. In order to issue the request on the network, the BIOS uses 1K of RAM from the highest memory address. This 1K memory location is assigned using the memory size word from the ROM BIOS data area during power-up.

5. After the above steps are finished, every INT 13 request is sent to IBMNETBOOT. If the request is a Read, a RECEIVE is issued for an 11-byte header plus the data. For a Write request, 11-bytes plus data are sent.

6. The following INT 13 requests are not sent by the BIOS:

   - All format requests (diskette and fixed disk)

   - Read long (fixed disk)

7. The INT 13 redirection to the network can be turned off by using the UNLINK command. This will allow a program to use other forms of a drive redirection to the network.

Sample Programs

Two sets of sample programs are included on the diskette in this book. The first sample is a listing with a simple example of how to use the BIOS. The second set of programs is an example of a working network using the BIOS and the remote program load features of the adapter.

Sample Program Set 1

This example uses two files on the Network Sample Program diskette in the back of this book. Print out the files CALL.LST and LISTEN.LST with the DOS TYPE command and refer to the following description of the programs.

To get started:

- You do not need to issue a RESET command if the default parameters are acceptable.
- Create an NCB with your name in the name field and issue an ADD NAME command. All other fields should be zero.
- Call NETBIOS by issuing an INT 5C.

To send data to someone using sessions:

- Create a CALL NCB inserting binary zeros in all unused fields.
- Issue INT 5C.

- Check the return code in the AL register when the command is finished.
- You can use the old CALL NCB, change the command to SEND and then fill in the correct fields. Wait or no-wait options can be used.
- Point the NCB buffer address (offset:segment) to the data you wish to send.
- Issue INT 5C
- When you are finished sending data, issue a HANG UP command.

To receive data, using the no-wait option, from someone who calls you:

- Issue a LISTEN NCB using the no-wait option.
- When someone calls you, the post address in the NCB is your starting address. You will be executing on an interrupt level with interrupts masked.
- After you are called, issue a RECEIVE command to the name of the caller and obtain data. Either wait or no-wait options can be used.
- A RECEIVE ANY command can also be used to obtain data from anyone you have a session with.

To receive status about active sessions, adapter status, or errors:

Issue either a session status or adapter status NCB request.

To broadcast a message:

Issue a SEND BROADCAST DATAGRAM command NCB

To Receive a Broadcast message:

- Issue a RECEIVE BROADCAST NCB.
- A RECEIVE BROADCAST command must be outstanding or the message is lost.

Sample Program Set 2

This set of programs is an example of a working network using the remote program load (RPL) feature on the adapter. These programs are only intended to be examples and may not work in all environments or operating systems. The intent of this example is to allow a programmer to learn more about the network and the BIOS.

Before you begin, the environment needs to be defined. The following lists is the equipment needed to perform the example network:

- One Personal Computer with 128K, a fixed disk, and an IBM PC Network Adapter. This computer is used as a dedicated computer to serve the network. When this computer is properly configured, it is known on the network as the Server computer.

- At least one Personal Computer with 64K and with an IBM PC Network Adapter. These computers can be any configuration. Also, each IBM PC Network Adapter must have jumper W1 removed to enable the RPL feature. See "Configurable Hardware Options" on page 3-70 for the location of the W1 jumper.

- An IBM PC Network Translator Unit.

- IBM PC Network cable.

- DOS 3.0 diskettes.

- Blank diskettes.

Getting Started

Once the network hardware is set up, the following steps will help you configure the software for the example:

1. Start with the Server computer. Install the IBM PC Network Adapter in this computer.

2. Using the DOS 3.0 Make Directory command (MD), create a directory called IMAGES in the root directory of the fixed disk.

3. Use the DOS COPY command to copy the programs IMGUTIL.COM and RPLS.EXE from the sample program diskette to the IMAGES directory.

4. At the C> prompt, format a diskette and copy the system onto it. Copy the programs MUSIC.BAS and BASIC.COM from the DOS 3.0 diskettes onto this diskette.

5. Type CD \ IMAGES at the C> prompt.

6. At the C> prompt, type IMGUTIL. This program creates an image of a diskette on the fixed disk. The entire diskette is copied, so be sure there is enough room on the fixed disk for the images you want to store.

7. The image utility prompts you for the name you want the image to have. At this prompt, type MUSIC. Insert into drive A the diskette you created from the previous step. Now you have an image called MUSIC on the Server's fixed disk. This image contains the directory of the diskette, including any programs, data, and free space that was on the diskette.

8. After the program has copied the diskette to the Server computer's fixed disk, type RPLS at the C> prompt. Now the Server computer is dedicated to serve each computer loading from the network.

9. For the other computers in the network, install an IBM PC Network Adapter in each with jumper W1 removed. See "Configurable Hardware Options" on page 3-70 for information about the jumpers.

10. Now switch one of the remote computers on the network ON. If the jumper W1 is removed, the computer will try to load in the following order:

    a. From the diskette
    b. From the fixed disk (if available)
    c. From the network
    d. From ROM BASIC

11. When operating correctly, the remote computer should display the word MUSIC at the top of the screen. If you created other images on the Server's fixed disk, these images would also be displayed on the screen. These images are on the Server's fixed disk and can be used as if the files were on your A: drive. For example, if one of the images displayed was called NETWORK, you could use the DOS DIR command and display the directory of that image.

12. You should have the A> prompt on the bottom of the screen. Type DIR MUSIC at the A> prompt. You should see the contents of the image you just selected.

If you want to copy more diskette images onto the Server's fixed disk, repeat step 4 for each diskette you create. Each diskette must have DOS copied onto it in order for the program to work. The maximum number of images you can store on the Server's fixed disk is 6.

The above set of programs is an example of how the network uses the RPL feature, so these programs must be used with caution. You can have errors when two or more computers are trying to write to the same image at the same time because there is no protection built into the sample program.

Two Adapter Cards In the Same Personal Computer

You can install two IBM PC Network adapters in the same Personal Computer. The second adapter must be set so the BIOS knows which adapter is the second adapter. Jumper positions are identified in Chapter 3, "Configurable Hardware Options" on page 3-70.

Whenever you issue an NCB, the NCB__LANA__NUM field must be 00H to "talk" to the first adapter. See "NCB Field Description" on page 2-15 for a description of the NCB__LANA__NUM field. Use 01H to "talk" to the second adapter.

Adapter Presence Test

A test can be issued in software to check for a working adapter in a computer. The following is the method to check for an adapter.

Test Method

1. Check interrupt vector 5C.

2. If the location contains all binary zeros, an adapter is not present in the computer.

3. If the location contains a value other than all binary zeros, issue either command 7FH or FFH.

4. If a code of 03H is returned, the adapter is present.

5. If a code other than 03H is returned, check the list below to see if the code returned is one of the following to make sure that the adapter is present.

a. 00H—Good return
b. 23H—Invalid number in NCB_LANA_NUM field
c. 4XH—Unusual network condition
d. (50–FE)H—Adapter malfunction
e. FFH—Command pending status

Note: The X can be any number.

Error Recovery

The following table lists the return codes and recommended actions.

Hex Val	Return Code Name	Return Code Meaning	Recommended Actions
00H	Good return	Command complete	No action required. This is normal after each successful command.
01H	Illegal buffer length for SEND DATAGRAM, SEND BROADCAST, ADAPTER STATUS, or SESSION STATUS.	A SEND BROADCAST or SEND DATAGRAM cannot send more than 512 bytes. For ADAPTER and SESSION STATUS, the buffer length specified was less than the minimum required.	Specify the correct size for the buffer and try again.
03H	Invalid command	The command code used was incorrect.	Reissue the correct command code.

Figure 2-5 (Part 1 of 7). Return Codes and Recommended Actions

Hex Val	Return Code Name	Return Code Meaning	Recommended Actions
05H	Command timed-out	The return code field has following meanings: 1)For a CALL or for 2)ADAPTER STATUS, the system time-out period has elapsed. 3)For a SEND or for 4)RECEIVE, the time-out period specified for the CALL or LISTEN has elapsed. 5)For a HANG UP, the time-out period has expired for an outstanding SEND to complete.	1)For a CALL, try again later. 2)For an ADAPTER STATUS, make sure you are using a correct name. 3)For a SEND, the session has been terminated abnormally. Establish another session and reissue a SEND. 4)For a RECEIVE, reissue the command. 5)For a HANG UP the session has been terminated abnormally.
06H	Message incomplete	You received part of a message because your specified buffer length is not big enough to receive the full message.	You must reissue another RECEIVE or RECEIVE ANY command to get the rest of the message before the remote computer times-out. For ADAPTER STATUS, SESSION STATUS, RECEIVE DATAGRAM, and RECEIVE BROADCAST DATAGRAM, the remaining data is lost.

Figure 2-5 (Part 2 of 7). Return Codes and Recommended Actions

Hex Val	Return Code Name	Return Code Meaning	Recommended Actions
08H	Illegal local session number	The session number you specified is not one of the active sessions.	Specify an active session number when you issue a command.
09H	No resource available	Not enough space available in the adapter for the session.	Reissue the command at a later time.
0AH	Session closed	The session has been closed from either the local or remote computer.	No action is required. This is notification for a pending SEND or RECEIVE command that the session has been closed. For a HANG UP, the session was closed by the remote computer.
0BH	Command canceled	Notification received that the command was canceled. If the command that was canceled was a SEND or a CHAIN SEND,the session is abnormally terminated.	No action is required.
0DH	Duplicate name in local name table	You tried to specify a name that is already in the local name table.	Specify another name.
0EH	Name table full	Up to 16 names have already been added.	Wait until a delete name is issued so an entry will become available.

Figure 2-5 (Part 3 of 7). Return Codes and Recommended Actions

Hex Val	Return Code Name	Return Code Meaning	Recommended Actions
0FH	Command completed, name has active sessions and is now de-registered.	The name to be deleted is active in a session now, but is de-registered. When the name is marked de-registered and has active sessions, it still occupies a slot in the table. Name is unusable.	Close all sessions using this name for the DELETE command to complete.
11H	Local session table full	There are no available entries in the session table. (The number of sessions for a table is user- specified.)	1) Wait until a session has closed so an entry becomes available. 2) Refer to the RESET command to alter values.
12H	Session open rejected	No LISTEN command is outstanding on the remote computer.	Wait until a LISTEN is issued on the remote computer.
13H	Illegal name number	Invalid name number.	You must use the original name number that was assigned to the name.
14H	Cannot find name called or no answer	The call name specified cannot be found or did not answer.	Verify that the call name used is correct. Retry with the correct or a different call name or reissue if the remote computer is busy.

Figure 2-5 (Part 4 of 7). Return Codes and Recommended Actions

Hex Val	Return Code Name	Return Code Meaning	Recommended Actions
15H	Name not found, cannot specify * , or 00H.	Either the name you specified was not in the table or you specified an asterisk in column 1 of the name field or you specified 00H.	An asterisk or 00H in column 1 is not allowed. Retry with another name and verify that it is the correct name.
16H	Name in use on remote adapter.	Unique names can only be used once on the network.	Specify another name.
17H	Name deleted	This occurs when a name is deleted and there are no outstanding LISTEN, RECEIVE ANY, RECEIVE DATAGRAM, or RECEIVE BROADCAST DATAGRAM commands for that name.	No action required.
18H	Session ended abnormally	Either the remote computer is powered off, the cable link is broken, the session SEND or CHAIN SEND has timed-out, or the SEND or CHAIN SEND was canceled, or a HANG UP timed-out waiting for a SEND to complete.	1) Check the remote end for status and check the cable. 2) For a SEND or CHAIN SEND, or RECEIVE or RECEIVE ANY, reestablish the session.

Figure 2-5 (Part 5 of 7). Return Codes and Recommended Actions

Hex Val	Return Code Name	Return Code Meaning	Recommended Actions
19H	Name conflict detected	Network protocol has detected two or more identical names on the network.	Everyone on the network should delete that name immediately.
1AH	Incompatible remote device	Unexpected protocol packet received.	Verify that all units on the network agree with the network protocols.
21H	Interface busy	You called the BIOS out of an interrupt handler routine in process.	Return from the interrupt handler and try again later.
22H	Too many commands outstanding	The maximum number of commands are outstanding.	If not at maximum number, refer to RESET. If at maximum number, retry at a later time.
23H	Invalid number in NCB__LANA__ NUM field.	You tried to specify a number other than 00H or 01H.	Specify either 00H for the first adapter or 01H if you have and want to use the second adapter. Correct the number and try again.
24H	Command completed while cancel occurring	You tried to cancel a command that already completed, or never existed.	No action required.
26H	Command not valid to cancel	You tried to cancel a command that is invalid to cancel.	See CANCEL command for the list of commands not valid to cancel.

Figure 2-5 (Part 6 of 7). Return Codes and Recommended Actions

Hex Val	Return Code Name	Return Code Meaning	Recommended Actions
4XH	Unusual network condition Note: The X can be any Hex value.	The BIOS has detected an unusual condition in the network.	Either retry or reset the command. If the error is displayed again, refer to your *IBM Guide to Operations* for the appropriate action.
50-FEH	Adapter malfunction	The adapter has detected an internal problem.	Retry the operation. If you receive the code again, contact your authorized dealer.
FFH	Command pending status	The command is still pending.	No action is required. See NCB__POST@ and NCB__RETCODE for description of this return code.

Figure 2-5 (Part 7 of 7) Return Codes and Recommended Actions

Notes:

Notes:

Chapter 3. IBM PC Network Hardware Description

Contents

Introduction

This chapter provides a description of each of the components that make up the IBM PC Network. The IBM PC Network Adapter connects a Personal Computer to a broadband local area network. The IBM PC Network Translator Unit is used by all of the Personal Computers to communicate with each other. The Translator Unit also has connection hardware for up to eight Personal Computers. Up to 72 nodes within a 1000 foot radius are supported by using the IBM Translator Unit and the IBM cable kits.

The IBM PC Network Coaxial Cable system is also detailed in this chapter.

IBM PC Network Adapter

The adapter plugs into one of the I/O slots within the Personal Computer. All required network signals and protocols are controlled by the adapter.

The adapter has two major sections, Digital and RF Modem. The following block diagram shows how the internal data path is structured within the adapter.

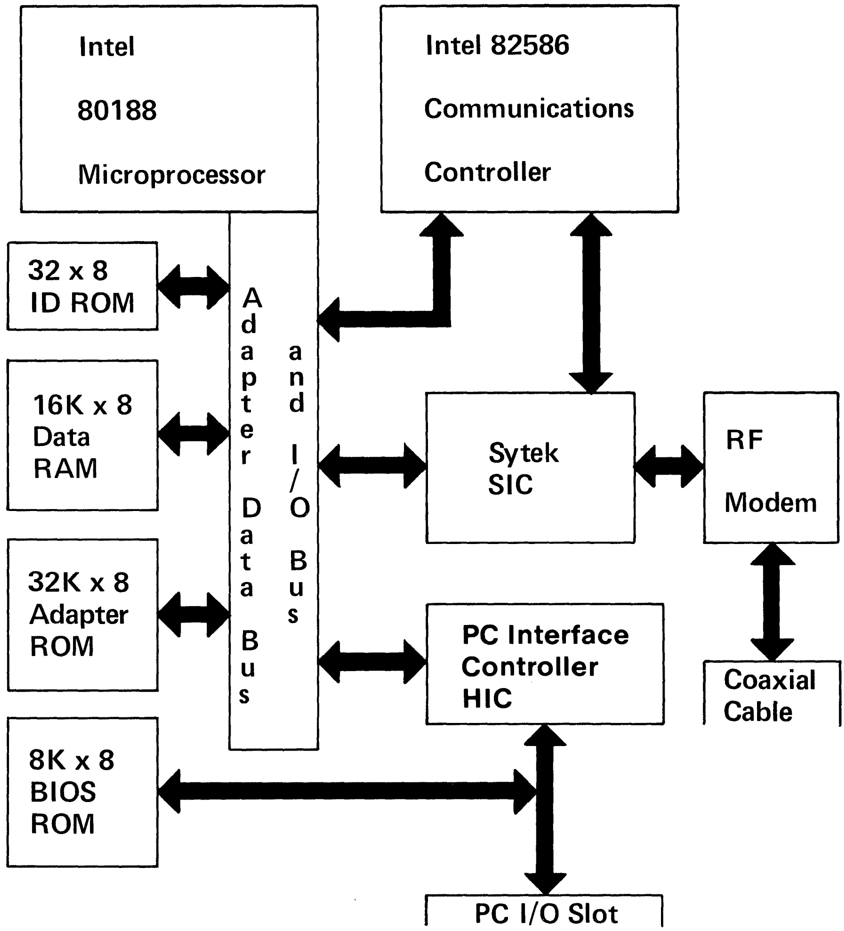


Figure 3-1 Functional Block Diagram

Digital Section

This section consists of the following components:

- Intel 80188 microprocessor
- Personal Computer / Host Interface Controller (HIC)
- Intel 82586 Local Communications Controller (LCC)
- Sytek Serial Interface Controller (SIC)
- A 16K x 8 of Data RAM
- A 32K x 8 Adapter ROM
- A 32 x 8 ID PROM
- An 8K x 8 BIOS ROM

(K = 1024 bytes)

All components in this section are connected within the adapter by an 8-bit wide internal data bus except the 8K x 8 BIOS ROM. The 8K x 8 ROM is directly accessed through the HIC.

The 80188 Microprocessor

The 80188 contains two independent direct memory access (DMA) channels, programmable timers, a programmable interrupt controller, and bus interface logic on a single integrated circuit. Additional information may be obtained in the publications listed in the bibliography. Communication between the Personal Computer and the adapter logic is accomplished through the Personal Computer / Host Interface Controller (HIC).

The microprocessor uses one of the two DMA channels to transfer data between the Personal Computer's interface registers and the local buffer memory. The microprocessor accepts four external peripheral interrupt requests. Two of these interrupts come from the Personal Computer interface. Another interrupt comes from the Sytek Serial Interface Controller (SIC) and the other is from the Intel 82586 (LCC). The interrupt priority scheme is programmable, but is fixed by the adapter's microcode.

The 80188 provides three memory select signals to select three different ROM/RAM devices. Also, there are three peripheral select signals to select the local communication controller, the Personal Computer HIC, or the Sytek SIC.

Personal Computer Interface Circuits

The Personal Computer interface circuit comprises the HIC and several TTL bus transceivers and drivers. It allows the IBM PC Network Adapter to appear as an array of I/O address spaces in the Personal Computer. These circuits contain a set of interface registers and necessary control logic to transfer commands and data between the Personal Computer memory and the local buffer memory.

RF Modem Interface Circuits

The Sytek Serial Interface Controller (SIC) and the 82586 (LCC) comprise the interface to the RF Modem portion of the IBM PC Network Adapter. Together, they interface data and commands from the parallel internal data bus to the serial network. They also implement the link layer protocols required by the network. Additional information on the 82586 can be obtained in the publications listed in the bibliography.

Adapter RAMs

The IBM PC Network Adapter uses 16K x 8 of RAM as its buffer memory. This is dynamic RAM and the circuitry on the adapter does the refresh operation. The circuitry is controlled by a programmable timer in the 80188.

This memory is used for the IBM PC Network Adapter's internal scratch memory, stacks, protocol control, and for buffering transient data to and from the Personal Computer interface registers.

Adapter ROMs and PROM

The BIOS ROM is located on the adapter in a single 24-pin, 8K x 8 bit integrated circuit. It is an extension of the Personal Computer's software for network support. It can be accessed by the Personal Computer and can be disabled by removing jumper W8 on the adapter. See "Configurable Hardware Options" on page 3-70 for a description of the jumper positions.

The ROM has separate output control and enable control lines. Whenever an access is made to the adapter BIOS by the Personal Computer, this ROM is selected by the control lines.

The 32K x 8 Adapter ROM contains program and protocol information for the 80188 microprocessor.

The 32 x 8 ID PROM contains information about the adapter's ID. This number is the network node address of the adapter and is unique for each adapter. The address is also referred to as the permanent node name.

IBM PC Network Adapter Characteristics

The following paragraphs describe the characteristics of the IBM PC Network Adapter. These characteristics also apply to many different types of Personal

Computers that can have an IBM PC Network Adapter installed. The following paragraphs refer to this as either a "host" or a "host computer", meaning any Personal Computer with an IBM PC Network Adapter installed.

Data Transfer

The fundamental software interface between a host's processor and the IBM PC Network Adapter's processor is provided by a set of interface registers in the HIC. One of the interface registers is the data register, a bidirectional, 2-byte FIFO register. The data register holds data to be transferred between a host and the IBM PC Network Adapter. The data register is readable and writable by both the host and the adapter. Specific operations being performed determine which processor reads and which processor writes.

There are three basic methods to transfer data through the data register from host memory to adapter memory. Note that the transfer of data from adapter to host is analogous to the transfer of data in the opposite direction. Both the host processor and the adapter processor can choose to use any of these three methods independently. The three methods are:

- Polled I/O
- Interrupt I/O
- DMA transfers

DMA Operations

For best performance, the primary method for transferring data between the host and adapter is accomplished by DMA operations. There are two separate DMA controllers that can access this interface. The host DMA controller is used for transferring data between host memory and the data register. The two-channel DMA controller within the Intel 80188 is used in the adapter for transferring data between adapter memory and the 2-byte FIFO data register. The maximum transfer rate of the Intel 80188 integrated DMA controller is 1.0 Mbytes per second.

DMA Scheme

The Personal Computer HIC provides two DMA requests, one to the IBM PC Network Adapter processor and one to the Personal Computer processor. The Personal Computer's DMA controller is responsible for moving data and commands between the Personal Computer's system memory and the adapter interface data register. When instructed, the 80188's DMA controller moves data and commands between the local buffer memory and the interface data register.

To ensure reliable data transfer, handshake signals such as DMA REQUEST, DATA REGISTER FULL, and DATA REGISTER EMPTY are used to synchronize the data flow to and from both sides.

Upon completion of a host DMA transfer, a DMA complete interrupt can be sent to the host processor through the adapter.

The Intel 80188's integrated DMA controller is controlled by the adapter processor. It accepts a 20-bit address, a 2-byte transfer length and an I/O address from the adapter processor. Also upon completion of DMA transfer, a DMA complete interrupt is sent to the adapter processor.

Interrupt Structure

Another method of communication between the host and the adapter is through vectored interrupts. These are asynchronous events, which cause a change in the flow of program control of the processor (either the host or the adapter) being interrupted. Each processor can be interrupted both by events external to it, such as a request for service from the interface or by internal events such as the completion of an operation initiated by the same processor.

A processor may, for various reasons, want to prevent interrupts from occurring. For example, interrupts usually are not allowed while a critical data structure or piece of status information is being updated. Also, a particular application program or operating system may not be equipped to deal with interrupts. For this reason, each processor has the ability, by the setting of certain bits in its interface registers, to enable and disable the various interrupts on an individual basis.

There are six distinct types of interrupts that each processor may receive. These are discussed separately for the host and for the adapter.

Host Interrupts

1. The first type of interrupt received by the host informs it that it has received control of the interface to the adapter and may place data in the Interface registers.

   The purpose of this interrupt is to free the host from having to wait for the adapter to yield control of the interface. The host processes some other task until the interface becomes free. Then the host is notified by the interrupt. Upon receipt of this interrupt, the host issues a command to the adapter and then relinquishes control of the interface. This interrupt occurs when the following three conditions are satisfied:

   a. The host has requested control of the interface by setting the Host Control Request (HCR) bit in the Host Interface register.

b. The adapter has allowed host control of the interface by setting the Host Control Enable (HCE) bit in the Adapter Interface register.

c. The host has enabled this interrupt by setting the Host Control Interrupt Enable (HCI) bit in the Host Interface register.

2. The second kind of interrupt the host can receive is caused by a request from the adapter for service or attention. This interrupt usually occurs as a result of the host's having instructed the adapter to perform a function. While the adapter is executing a function, it requires support from the host. For example, if the host orders the adapter to send a message to another point on the network, the adapter must request that the host transfer the data to the adapter. This request is necessary because the adapter is not permitted to control the host's data bus, and thus is unable to perform the transfer without the cooperation of the host.

   This interrupt to the host occurs when the adapter, after gaining control of the interface, sets the GO bit in the Status register. To receive the interrupt, the host must have enabled it by setting the Go Interrupt Enable (GI) bit in the Host Interface register. After servicing the adapter's request, the host clears the GO bit, and then the adapter relinquishes control of the interface.

3. The third type of interrupt sent to the host is triggered by the same bit as the previous one. When the adapter, after executing a command issued by the host, clears the GO bit in the Status register, the host receives an interrupt. This interrupt is received only if the host has enabled it by setting the GI bit in the Host Interface register. Upon receipt of the interrupt, the host either issues another command to the adapter, or relinquishes control of the interface by clearing the Host Control Request (HCR) bit in the Host Interface register.

4. The next two types of interrupts seen by the host are related to the transfer of data between the adapter and the host. All such transfers take place through the Data register.

   When one processor writes data into the Data register, whether under control of the processor itself or of its DMA hardware, the interface hardware sets the Data Register Full (DRF) bit in the Status register. This informs the other processor that 1 or 2 bytes of data are available for reading.

   Similarly, when the Data register is read by the other processor, the hardware sets the Data Register Empty (DRE) bit in the Status register, indicating that one or two bytes of data may now be written to the Data register.

   Two bits in the Host Interface register control interrupts that are related to data transfer. The first of these is the Data Transfer Interrupt Enable (DTI) bit. Setting this bit causes an interrupt when the Data Register Empty or Full (DRE or DRF) bits are set depending on the setting of the second bit.

   The second bit is the Data Direction (DD) bit, which indicates the direction of data flow through the Data register.

   By setting the Data Transfer Interrupt Enable (DTI) and the Data Direction (DD) bit in the Host Interface register, the host can arrange to receive an interrupt every time a byte of data is written to the Data register by the adapter. Likewise, setting the DTI bit but clearing the DD bit causes the host to receive an interrupt each time a byte of data written to the Data register by the host is read by the adapter. Both the Data Register Full and the Data Register Empty (DRF and DRE) bits may be set at the same time; the two conditions are not mutually exclusive.

5. The final type of interrupt sent to the host is caused by the termination of a DMA operation between the host and the adapter. Such operations are to send command blocks from the host to the adapter, to return completed command blocks to the host, or to transfer message between the host and the adapter. If the host transfers data to and from the adapter through DMA operations (it does not need to do so), it has the option of receiving an interrupt upon completion of the operation. This can be done by setting both the Terminal Count Interrupt Enable (TCI) and the Data Transfer DMA Enable (DTD) bits in the Host Interface register. Whenever the DMA transfer mode is being used, the adapter interface detects the host signal indicating DMA completion on DMA channel 3 and sets the Terminal Count (TC) bit in the Status register. If the TCI bit is set, this will cause an interrupt to the host.

The adapter can receive a set of interrupts similar to that seen by the host. These interrupts are described in the following paragraphs.

Adapter Interrupts

1. The first type of interrupt that can be sent to the adapter is analogous to the one received by the host when it has acquired control of the interface to the adapter. When the adapter requires control of the interface, it sets the Host Relinquish Interrupt Enable (HRI) bit in the Adapter Interface register. When the host yields the interface by clearing the Host Control Request (HCR) bit in the Host Interface register, an interrupt is sent to the adapter. The adapter should have previously cleared the Host Control Enable (HCE) bit in the Adapter Interface register so that once the host gives up control of the interface it is prevented from regaining it until the adapter finishes using the interface and again sets

the HCE bit. Like the corresponding host interrupt, this interrupt allows the adapter to continue other processing while waiting for control of the interface.

2. The second kind of interrupt sent to the adapter results from the host's sending the adapter a request for service or attention. These requests usually take the form of a command block constructed by the host's application program. The adapter receives this interrupt, if it is enabled, when the host, after acquiring control of the interface, sets the GO bit in the Status register. The adapter enables the interrupt by setting the Go Interrupt Enable (GI) bit in the Adapter Interface register. After the adapter has received the interrupt and serviced the host's request, it clears the GO bit; the host then gives up control of the interface.

3. The third type of interrupt sent to the adapter is also triggered by the GO bit in the Status register. When the host, after executing a command issued by the adapter, clears the GO bit in the Status register, the adapter receives an interrupt. This interrupt is received only if the adapter has enabled it by setting the GI bit in the Adapter Interface register. Upon receipt of the interrupt, the adapter can elect either to issue another command to the host or to relinquish control of the interface by setting the HCE bit in the Adapter Interface register.

4. The next two interrupts that the adapter can receive correspond to the host's Data register-related interrupts. If the Data Transfer Interrupt (DTI) bit is set in the Adapter Interface register an interrupt is sent to the adapter each time the DRF or DRE bits are set in the Status register, depending upon the setting of the DD bit in the Adapter Interface register.

5. The last type of interrupt seen by the adapter is an internal one, not dependent on the host. When the adapter performs a DMA transfer to or from the host, it instructs its own DMA controller hardware as to the source and destination addresses, length of transfer and mode of operation. If instructed to do so, the DMA hardware sends an interrupt to the adapter when the transfer count has been decremented to zero. This completes the DMA operation.

Summary

As discussed above, each processor can receive six different types of interrupts related to the interaction between the host and the adapter. The following rules govern interrupts between host and adapter.

- Each of these interrupts can be disabled by the host or adapter on an individual basis.
- Each processor can receive an interrupt when it acquires control of the interface registers.
- Each processor can be interrupted by a request from the other processor for service or attention, and the recipient is responsible for clearing the GO bit.
- Each processor can receive an interrupt, when the other processor clears the GO bit, in response to the setting of the GO bit by the first processor.
- Each processor can be informed when a new data byte is available in the Data register and when a byte it has written there has been read by the other processor.
- Finally, each processor can receive an interrupt when its DMA hardware has completed a transfer.

Adapter Initialization

The IBM PC Network Adapter executes initialization and self-test routines whenever it detects a hardware reset signal from the Personal Computer during power-up or when it detects that the software control reset bit in the Personal Computer HIC is set. The Personal Computer controls the reset and initialization of the IBM PC Network Adapter during system operations. The following is a description of what occurs during a software reset.

The adapter is held in the reset state whenever the Reset Adapter (RES) bit in the Host Interface register is set. The Reset Adapter bit is set when the RESET DRV signal from the host is active. Setting this bit does not require that the host have control of the interface, because the reason for performing the reset may be to force an errant adapter to give up interface control. This capability must be used carefully.

When the adapter is released from reset by clearing the reset bit, the following actions occur:

1. A self-test is performed.

2. All internal data structures and command queues are reinitialized.

3. All names are deleted from the name table.

4. All sessions are aborted.

The adapter then sends an Adapter Initialization Complete primary command to the host. The host acknowledges the command by clearing the GO bit in the Adapter Interface register, allowing the host to gain control of the interface.

Programming Interface

This section discusses the programming interface between the host computer and the adapter. This includes the interface protocol, primary and secondary commands, DMA considerations, interrupts, aliases, broadcasting, resetting the adapter, and aborting commands.

Secondary Commands

These commands are known as Network Control Blocks (NCBs). See Chapter 2 for a complete description of the NCBs.

Interface Protocol

This section describes the interface protocol between the adapter and the host computer's BIOS (Basic Input Output System). It discusses the basic transactions between the adapter and the host and the steps required to conduct these transactions. Interrupts and interface registers are mentioned, but not described in detail. Detailed descriptions of these aspects are contained in later sections.

There are three main transactions between the host and the adapter as follows:

1. The host sends a command block to the adapter.

2. Adapter asks host to transfer message data.

3. The adapter asks the host to accept an updated command block.

These transactions are described in the following paragraphs.

Host Sends a Command Block to the Adapter

Most of the communication between the host and the adapter is initiated by the host. The host requests the adapter to execute a particular command, such as opening a session. These commands are sent as *command blocks* to be interpreted by the adapter. The command blocks are sent using the following steps:

1. The host's application program constructs a network command block (NCB) and invokes BIOS. This method of construction is specified in detail in Chapter 2 and is not described here.

2. BIOS requests control of the adapter interface by setting the Host Control Request bit in the Host Interface register. If the Host Control Interrupt Enable bit in the Host Interface register is set, an interrupt is issued to the host when interface control is granted.

3. After receiving control of the interface, BIOS writes the 1-byte primary command code followed by the address of the command block (32 bit address in DD format) and the length of the command block (2 bytes) to the Parameter register. The BIOS then sets the GO bit in the Status register. This issues the command by causing an interrupt, when enabled, to be sent to the adapter.

4. BIOS sends the command block to the adapter through the Data register. This may be done by setting up a DMA transfer or through programmed I/O (the host processor writes a byte at a time to the Data register), at the option of the host BIOS. If using programmed I/O, the host can also choose whether to receive an interrupt as each byte is read by the adapter.

5. The adapter finds, either through an interrupt or by polling, that the GO bit is set.

6. The adapter reads the information in the Parameter registers.

7. The adapter obtains the command block, either through programmed I/O (with or without using interrupts) or through DMA.

8. The adapter clears the GO bit and loads the primary command completion code in the Status register.

9. The host finds, either through an interrupt or by polling, that the GO bit was cleared; the host then releases the interface by clearing the Host Control Request bit in the Host Interface register. This may cause an interrupt to the adapter, informing it that the interface is now available.

10. The adapter queues the new command internally. If the queue is full, the adapter clears the Host Control Enable bit in the Adapter Interface register and sets the Set Command Queue Full (SQF) flag. This prevents the host from issuing any more commands until some commands have been processed.

Adapter Asks Host To Transfer Message Data

Most commands involve a transfer of message data either from the host to the adapter (for example, session send) or from the adapter to the host (for example, session receive). In either case, the adapter must ask the host to transfer the data, since the adapter cannot gain control of the host's data bus. When the adapter wishes to transfer data between itself and the host, the following events occur:

1. The adapter gains control of the interface by clearing the Host Control Enable bit in the Adapter Interface register and waits for the Host

Control bit in the Status register to be cleared. If the adapter chooses, it receives an interrupt when interface control is granted.

2. The adapter loads the command followed by the address of the data buffer and length of the transfer (previously obtained from the command block) into the Parameter register. It then sets the GO bit in the Status register. This issues the command by causing an interrupt, if enabled, to be sent to the host.

3. The adapter starts its part of the data transfer through the Data register, either through programmed I/O (with or without interrupts) or through DMA.

4. The host finds, either through an interrupt or by polling, that the GO bit is set.

5. The host reads the information in the Parameter registers.

6. The host performs the data transfer, either through programmed I/O (with or without interrupts) or through DMA.

7. The host clears the GO bit and loads the primary command completion code in the Status register.

8. The adapter finds, either through an interrupt or by polling, that the GO bit was cleared; the adapter then releases the interface by setting the Host Control Enable (HCE) bit in the Adapter Interface register. This can cause an interrupt to the host if it had previously requested control of the interface.

Adapter Asks Host To Accept Updated Command Block

The final operation in completing a command is the transfer of the updated command block to the host. This is done with the following steps:

1. The adapter updates the command block with completion status and other information, depending on the particular command.

2. The adapter gains control of the interface by clearing the Host Control Enable bit in the Adapter Interface register and waits for the Host Control bit in the Status register to be cleared. If the adapter wishes, it receives an interrupt when interface control is granted.

3. The adapter writes the command code 43H followed by the address and length of the command block to the Parameter register; it then sets the GO bit in the Status register. This issues the command, causing an interrupt, if it is enabled, to be sent to the host.

4. The adapter starts its part of the data transfer through the Data register, either through programmed I/O (with or without interrupts) or through DMA.

5. The host finds, either through an interrupt or by polling, that the GO bit is set.

6. The host reads the information in the Parameter register.

7. The host transfers the command block to its memory, either through programmed I/O (with or without interrupts) or through DMA.

8. The host clears the GO bit in the Status register.

9. The adapter finds, either through an interrupt or by polling, that the GO bit was cleared; the adapter then releases the interface by setting the Host Control Enable bit in the Adapter Interface register. This can cause an interrupt to the host if it has previously requested control of the interface, and has enabled this interrupt.

10. The host BIOS returns the updated command block to the host application program.

Interface Control

There are four types of interface registers in the Personal Computer interface controller:

- Status register
- Parameter register
- Data register
- Host/Adapter Interface register

Each register has two addresses depending on the setting of jumpers W5, W6 and W7. See section "Configurable Hardware Options" on page 3-70 for a detailed description of the jumper positions. The Personal Computer I/O register addresses are as follows:

Register name	Low address range	High address range
Status register	360H	368H
Parameter register	361H	369H
Data register	362H	36AH
Host/Adapter Interface register	363H	36BH

Figure 3-2 Register Addresses

The adapter interface registers are used to pass commands, parameters, and data between the host and the adapter. Both the host and adapter may desire to pass information through this interface. There is only one set of registers, so a mechanism to resolve conflicts over register use must be included in the interface. Determination of who is in control of the interface is made by examining the Host Control (HC) bit in the Status register. When this bit is set, the interface is under control of the host. When the HC bit is not set the interface is either under control of the adapter or is idle.

When the adapter desires control of the interface, it clears the Host Control Enable (HCE) bit in the Adapter Interface register. Clearing the HCE bit inhibits the HC bit from being set. However, if the HC bit is already set the adapter must wait until the host has cleared the Host Control Request (HCR) bit, causing the HC bit to be cleared. When the HCE and HC bits are both cleared, the adapter is in control of the interface and can use the interface registers. When the adapter is finished with the interface, it sets the HCE bit in the Adapter Interface register. This allows the host to gain control of the interface.

When the host desires control of the interface it sets the Host Control Request (HCR) bit in the Host Interface register. If the HCE bit has been set by the adapter in the Adapter Interface register, the Host Control (HC) bit is immediately set in the Status register. If the HCE in the Adapter Interface register is not set, the HC bit is not set until the adapter sets the HCE bit. Once the HC bit has been set, the interface is considered to be under control of the host. When the host is finished with the interface it clears the HCR bit in the Host Interface register. This causes the Host Control (HC) bit to be cleared in the Status register.

The following table gives the state of the interface for various combinations of the HCE, HCR, and HC bits.

HC	HCR	HCE	State
0	0	1	Idle
0	0	0	Adapter in control
0	1	0	Adapter in control, host waiting for control
1	1	1	Host in control
1	1	0	Host in control, adapter waiting for control

Figure 3-3 Interface Control States

Data Transfer

Bulk data transfers between the host and the adapter occur through the Data register. Each side of the interface can elect to transfer data through this register using DMA, interrupts, or polling. The Data Direction (DD), Data Transfer Interrupt (DTI), and Data Transfer DMA (DTD) bits in the interface registers are used to select the transfer mode independently for each side. The following table summarizes the use of these bits.

DD	DTI	DTD	Operation
X	0	0	Polled I/O
0	1	0	Host to adapter using interrupts
1	1	0	Adapter to host using interrupts
0	0	1	Host to adapter using DMA
1	0	1	Adapter to host using DMA
X	1	1	Illegal

Figure 3-4 Interface Registers Transfer Control Bits

In addition the Terminal Count Interrupt (TCI) bit in the Host Interface register can be set to interrupt the host when terminal count has been reached in the host DMA.

Status Register (SR)

Addresses:

Host: 360H or 368H
Adapter: 00H

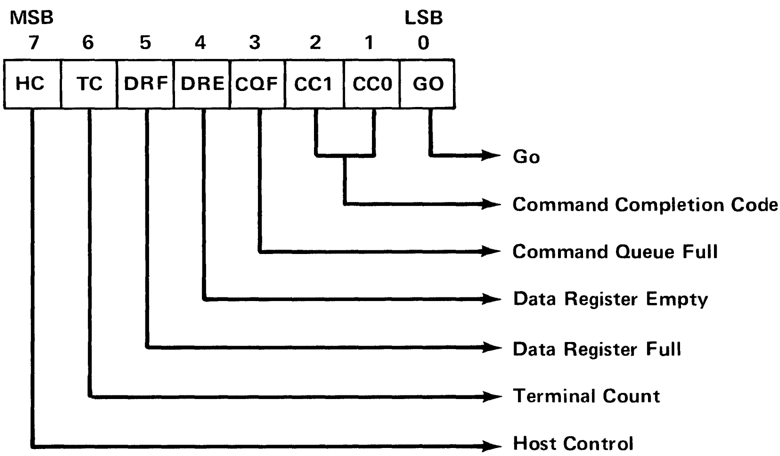


Figure 3-5 Status Register

Go (GO)

Indicates the command is now ready for execution. If the GI bit is set for a side, that side is interrupted whenever the other side sets or clears this bit. It is cleared by the side receiving the primary command when the command has been completed. This bit can be read and written to by both sides and is cleared upon a reset.

Command Completion Code (CC0-CC1)

Used to pass the primary command completion code. These bits are cleared when GO is set on command initiation, and set to the appropriate value when the GO bit is cleared by the side receiving the command. The following completion codes are as follows:

00H - Primary command completed successfully.

01H - Invalid primary command or parameter.

02H - Unable to complete primary command.

03H - Reserved

These bits can be read and written to by both sides and are cleared upon a reset.

Command Queue Full (CQF)

This bit is set as a result of setting the Set Command Queue Full (SQF) bit in the Adapter Interface register and indicates to the host that the adapter cannot accept any commands at this time. The bit is read-only for both sides, and is cleared upon a reset.

Data Register Empty (DRE)

This bit is set by the interface whenever a byte can be written to the data register. This bit can cause interrupts or DMA cycles to occur to the host or the adapter if the appropriate DTD, DTI, and DD bits are set as described in Figure 3-8 on page 3-34. This bit is always read-only and is set upon a reset.

Data Register Full (DRF)

This bit is set by the interface whenever a byte can be read from the data register. This bit can cause interrupts or DMA cycles to occur to the host or the adapter if the appropriate DTD, DTI, and DD bits are set as described in Figure 3-8 on page 3-34. This bit is always read-only and is cleared upon a reset.

Terminal Count (TC)

This bit indicates that terminal count has been reached on the host DMA channel while transferring data to or from the adapter. For this bit to be set the DTD bit must be set in the Host Interface register and the TC and DACK3 signals must be asserted on the host peripheral bus. If the TCI bit is set in the Host Interface register when this bit is set, an interrupt is sent to the host. This bit is cleared when the DTD bit is cleared in the Host Interface register. This bit is always read-only and is cleared upon a reset.

Host Control (HC)

This bit indicates that the host has control of the interface. The setting of this bit causes an interrupt to the host if the Host Control Interrupt (HCI) is set in the Host Interface register. The HC bit is cleared when the host clears the HCR bit in the Host Control Register. If the Host Relinquish Interrupt (HRI) bit is set in the the Adapter Interface register, the clearing of the HC bit causes an interrupt to be sent to the adapter. This bit is always read-only, and is cleared upon a reset.

Parameter Register (PR)

The Parameter register is a 7-byte shift register used to pass primary commands and their parameters. Because this register operates as a true shift register and not as a first-in, first-out (FIFO), it is necessary that 7 bytes always be written when passing parameters. This register is read/write by both sides but should only be written to by the side controlling the interface.

Addresses:

Host: 361H or 369H
Adapter: 02H

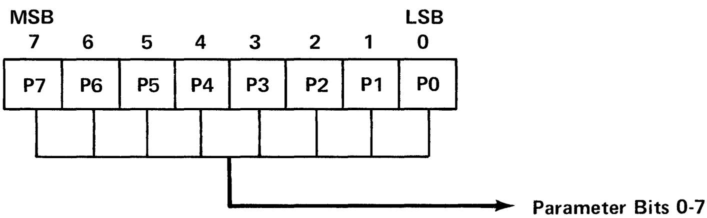


Figure 3-6 Parameter Register

Data Register (DR)

The Data register is a 2-byte FIFO used to pass data between the host and the adapter. When data is available to be read from this register, the Data Register Full (DRF) bit is set in the Status register. When data can be written to this register, the Data Register Empty (DRE) bit is set in the Status register. Because this register is double buffered, both of these bits can be set at the same time. The setting of these bits can cause interrupts and DMA cycles to occur depending on the setting of enable bits in the interface registers. This register is read/write by both sides. Depending on the command being performed, a determination is made as to which processor should read or write this register.

Addresses:

Host: 362H or 36AH
Adapter: 04H

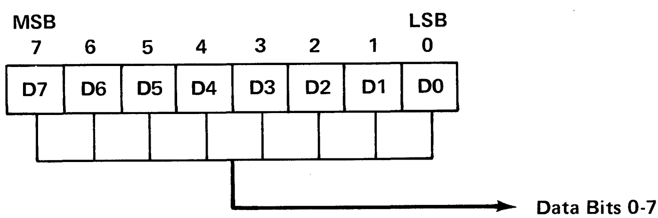


Figure 3-7 Data Register

Host Interface Register (HIR)

The Host Interface register is used by the host to control interrupt and DMA requests going to the host, to force the adapter to be reset, and to request control of the interface. This read/write register is only accessible by the host. Interrupts to the host occur on either IRQ2 or IRQ3, depending on the setting of jumpers W3 and W4 on the adapter. DMA requests occur on DRQ3 with acknowledgments on DACK3.

Addresses:

Host:	363H or 36BH
Adapter:	Inaccessible

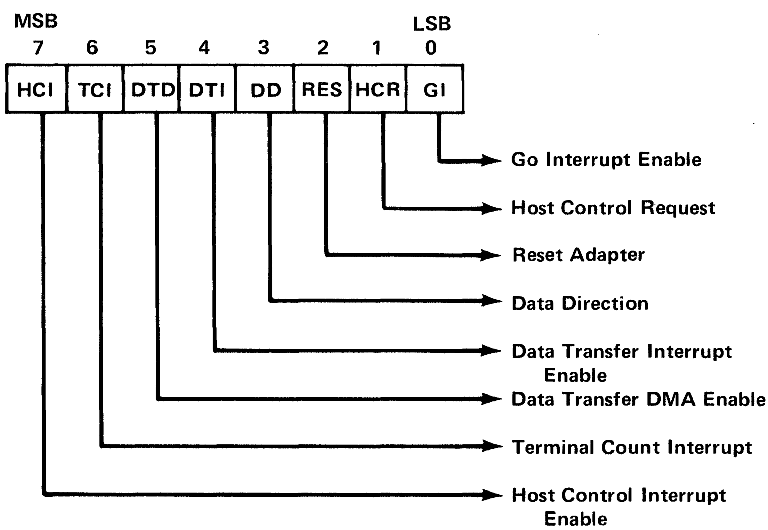


Figure 3-8 Host Interface Register

Go Interrupt Enable (GI)

If this bit is set, an interrupt is sent to the host if the GO bit is set or cleared by the adapter. The interrupt is cleared by the host's reading the Status register. This bit is cleared upon a reset.

Host Control Request (HCR)

This bit is set by the host when it wants to gain control of the interface. When host control is granted, the Host Control (HC) bit is set in the Status register. Clearing this bit clears the HC bit in the Status register. This bit is cleared upon a reset.

Reset Adapter (RES)

If this bit is set by the host, the adapter executes a hardware reset and is held in the dormant state until this bit is cleared by the host. This bit is set anytime the RESET DRV signal from the host channel interface becomes active.

Data Direction (DD)

This bit is set by the host to indicate the direction of data transfer between the host and adapter and works in conjunction with the DTI and DTD in controlling interrupts and DMA requests to the host. If this bit is set, data transfer is from the host to the adapter. If this bit is cleared, transfer is from the adapter to the host.

Data Transfer Interrupt Enable (DTI)

If this bit is set, an interrupt is sent to the host whenever the Data Register Empty (DRE) or Data Register Full (DRF) bit is set, depending on the setting of the Data Direction (DD) bit. If the appropriate DRE or DRF bit is already set when this bit is set, an immediate interrupt occurs. This interrupt is cleared by the host's accessing the Data register or by clearing the DTI bit. This bit is cleared upon a reset.

Data Transfer DMA Enable (DTD)

If this bit is set, a DMA request is sent to the host whenever the Data Register Empty (DRE) or Data Register Full (DRF) bit is set depending upon the setting of the Data Direction (DD) bit. If the appropriate DRE or DRF bit is already set when this bit is set, an immediate DMA request occurs. This DMA request is cleared by the DACK3 signal or clearing the DTD bit. This bit is cleared upon a reset.

Terminal Count Interrupt (TCI)

If this bit is set, an interrupt is sent to the host whenever the Terminal Count (TC) bit is set in the Status register. If the TC bit is already set when this bit is set, an immediate interrupt occurs. The interrupt is cleared by the host's reading the Status register or clearing the TCI bit. This bit is cleared upon a reset.

Host Control Interrupt Enable (HCI)

If this bit is set, an interrupt is sent to the host whenever the Host Control (HC) bit is set in the Status register. If Host Control is already set, an immediate interrupt occurs. This interrupt is cleared by the host's reading the Status register. This bit is cleared upon a reset.

Adapter Interface Register (AIR)

The Adapter Interface register is used by the adapter to control interrupt and DMA requests going to the adapter, and to control acquisition of the interface by the host. The adapter can read/write this register, but it is inaccessible from the host.

Addresses:

Host: Inaccessible
Adapter: 06H

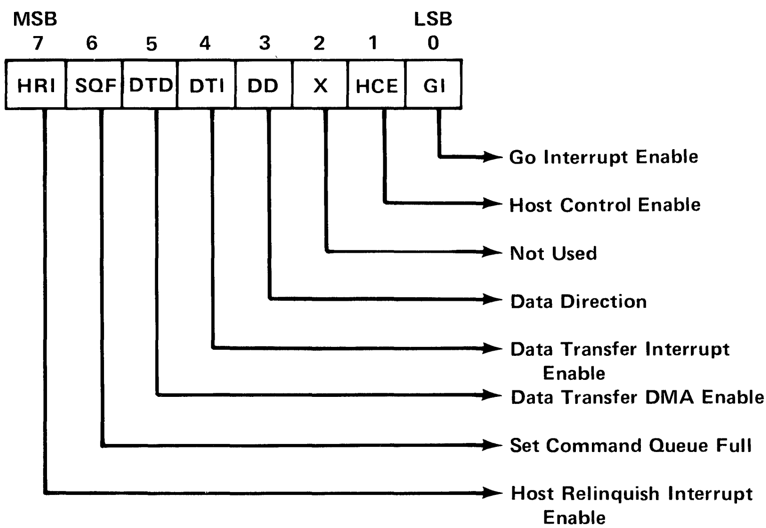


Figure 3-9 Adapter Interface Register

Go Interrupt Enable (GI)

If this bit is set, an interrupt is sent to the adapter if the GO bit is set or cleared by the host. The interrupt is cleared by the adapter's reading the Status register. This bit is cleared upon a reset.

Host Control Enable (HCE)

This bit allows the adapter to control the granting of interface control to the host. The HC bit is not set in the Status register in response to a Host Control Request (HCR) unless this bit is set. If the HCR bit is already set in the Host Interface register when this bit is set, the HC bit is immediately set in the Status register. Clearing this bit does not cause the HC bit to be cleared if it has already been set, but prevents it from being set again after it is cleared. The adapter must clear this bit before using the interface to prevent a control conflict. This bit is cleared upon a reset.

Data Direction (DD)

This bit is set by the adapter to indicate the direction of data transfer between the host and adapter, and works in conjunction with the DTI and DTD in controlling interrupts and DMA requests to the adapter. If this bit is set, data transfer occurs from the host to the adapter. If this bit is cleared, transfer is from the adapter to the host.

Data Transfer Interrupt Enable (DTI)

If this bit is set, an interrupt is sent to the adapter whenever the Data Register Empty (DRE) or Data Register Full (DRF) bit is set, depending on the setting of the Data Direction (DD) bit. If the appropriate DRE or DRF bit is already set when this bit is set, an immediate interrupt occurs. This interrupt is cleared by the adapter's accessing the Data register or clearing the DTI bit. This bit is cleared upon a reset.

Data Transfer DMA Enable (DTD)

If this bit is set, a DMA request is sent to the adapter whenever the Data Register Empty (DRE) or Data

Register Full (DRF) bit is set, depending on the setting of the Data Direction (DD) bit. If the appropriate DRE or DRF bit is already set when this bit is set, an immediate DMA request occurs. This DMA request is cleared by the adapter's accessing the Data register or clearing the DTD bit. This bit is cleared upon a reset.

Set Command Queue Full (SQF)

The setting of this bit is reflected in the Command Queue Full (CQF) bit in the Status register. The adapter sets this bit to indicate to the host that commands cannot currently be accepted by the adapter. This bit is cleared upon a reset.

Host Relinquish Interrupt Enable (HRI)

If this bit is set, an interrupt is sent to the adapter whenever the Host Control (HC) bit is cleared in the Status register. If the HC bit is already cleared, an immediate interrupt will occur. The interrupt is cleared by the adapter's reading the Status register. This bit is cleared upon a reset.

Primary Commands

Primary commands are those passed directly through the adapter's Parameter register. These commands are used to perform the most primitive level of communication between the adapter and the host.

Before a command is issued by either the host or the adapter, it must gain control of the interface. When the adapter desires control, it clears the Host Control Enable (HCE) bit in the Adapter Interface register and waits for the Host Control Request (HCR) bit in the Status register to be cleared.

When the host desires control, it requests control by setting the HCR bit in the Host Interface register and waits for the Host Control (HC) bit to be set in the Status register. The adapter can prevent the host from obtaining control of the interface by clearing the Host Control Enable bit in the Adapter Interface register. Normally the adapter prevents the host from gaining control only when the adapter is itself using the interface. However, when there is no memory space in the adapter to hold any more command blocks, the adapter can prevent the host from gaining control of the interface until memory space becomes available.

The primary commands and their parameters are passed through the Parameter register. This register holds the primary command followed by up to 6 bytes of parameter data in a 7-byte shift register. Because of the nature of this register, exactly 7 bytes must be written to it. In cases where there are less than 7 bytes of parameter data, additional dummy bytes must be written after the real parameters to bring the total number of bytes written to 7. When reading this register it is not necessary to read these dummy bytes.

The parameter data consists of a true 32-bit address of host memory.

After interface control has been obtained and the command and parameters have been written to the Parameter register, the GO bit is set in the Status register.

The side receiving the command can read the command and parameters and perform the desired action. If the command requires data to be transferred, the data is transferred through the Data register using the Data Register Full (DRF) and Data Register Empty (DRE) bits in the Status register to provide synchronization. On both sides of the interface, data can be transferred using DMA, interrupts, or polled I/O with each side able to independently select its mode of transfer.

When the side receiving the command has completed the command, it clears the GO bit to indicate the primary command has been completed and includes the primary command completion code in the Status register. Clearing the GO bit does not mean that secondary commands, that were passed by the primary command, have been accepted or completed.

Host-Initiated Commands

Transfer Command Block to Adapter

Command code: 01H

Parameters:

Command block address: low word, low byte
Command block address: low word, high byte
Command block address: high word, low byte
Command block address: high word, high byte
Length low byte
Length high byte

This command requests the adapter to accept a command block of the specified length that is stored in the host memory at the specified address.

After issuing the command, the host begins writing data to the Data register, whenever the Data Register Empty (DRE) bit is set in the Status register, until the entire command block has been transferred. The host waits for the GO bit to be cleared by the adapter.

The adapter, after reading the command, reads the parameters and performs whatever setup is required. It then reads the Data register when the Data Register Full (DRF) bit is set in the Status register and stores the data in its memory until the command block has been transferred. The adapter then clears the GO bit and loads the command completion code in the Status register to indicate the transfer has been completed.

At some later time, the command block is examined by the adapter and any secondary command is executed, if possible.

Abort Secondary Command

Command code: 02H

Parameters:
Command block address: low word, low byte
Command block address: low word, high byte
Command block address: high word, low byte
Command block address: high word, high byte
Secondary command code byte

This command requests the adapter to abort the secondary command found in the command block identified by the specified host address. After the abort has been attempted, the GO bit is cleared and the result placed into Status register bits CC0 and CC1.

Reconfigure Adapter

Command code: 05H

Parameters:
Number of sessions
Number of command blocks

This command determines the allocation of the adapter's RAM. It specifies the number of sessions and command blocks that the adapter should allocate (and hence the number of data buffers it can allocate). This command causes all current sessions and names to be removed from the name table.

A maximum of 32 sessions and 32 command blocks can be specified. If more are specified, only 32 are allocated. If 0 is specified, the adapter uses its internal default value, (6 sessions and 12 command blocks).

Adapter–Initiated Commands

Initialization Complete

Command code: 41H

Parameters: Initialization status in the Parameter register
80H–Successful completion
81H–Processor test failed
82H–ROM checksum test failed
83H–Unit ID PROM test failed
84H–RAM test failed
85H–Host interface test failed
86H–± 12V test failed
87H–Digital loopback test failed
8EH–Possible constant carrier
8FH–Analog loopback test failed

This command is issued by the adapter when it has completed its initialization after being released from the reset. Before issuing any commands to the adapter, the host waits for the adapter to issue this command. See "Power-On Self-Tests (POST)" on page 3-63 for a description of the tests.

Transfer Data To Host

Command code: 42H

Parameters: Buffer address: low word, low byte
Buffer address: low word, high byte
Buffer address: high word, low byte
Buffer address: high word, high byte
Length low byte
Length high byte

This command requests the host to accept the number of bytes of data specified by the length field, and to store this data starting at the specified address. After issuing the command the adapter begins writing data to the Data register, whenever the Data Register Empty (DRE) bit is set in the Status register, until all the data has been transferred. The adapter then waits for the GO bit to be cleared.

After reading the command, the host reads the parameters and performs whatever setup is required. It then reads the Data register when the Data Register Full (DRF) bit is set in the Status register and stores the contents of the Data register in memory until all the data has been transferred. The host then clears the GO bit in the Status register to indicate the command has been completed, and places the command completion code in the Status register.

Transfer Command Block To Host

Command code: 43H

Parameters:
Command block address: low word, low byte
Command block address: low word, high byte
Command block address: high word, low byte
Command block address: high word, high byte
Length low byte
Length high byte

This command operates in the same manner as the Transfer Data to Host command except that a command block is being passed back. The appropriate host response to this command is to clear the GO bit.

The GO bit is cleared after the transfer has been completed and before the control block is examined. This is to prevent tying up the adapter interface unnecessarily.

Transfer Data To Adapter

Command code: 44H

Parameters:
Buffer address: low word, low byte
Buffer address: low word, high byte
Buffer address: high word, low byte
Buffer address: high word, high byte
Length low byte
Length high byte

This command requests that the host transfer the number of bytes of data specified by the length and starting at the specified address to the adapter. After issuing the command, the adapter waits for the data to be written to the Data register and continues reading the Data register until all the data has been received. The adapter then waits for the GO bit to be cleared by the host.

After reading the command, the host reads the parameters and performs whatever setup is required. It then writes the requested data to the Data register until all the data has been transferred. The host then clears the GO bit in the Status register to indicate the command has been completed and places the command completion code in the Status register.

Error Report To Host

Command code: 45H

Parameters:
Error status in the Parameter register
41H–Continuous carrier detected (Not this adapter)
42H–Continuous carrier detected (This adapter)
43H–No carrier detected
50H—FEH–Internal software error

This command is issued by the adapter when it detects an irrecoverable error occurring after initialization. The host performs a report of the error to the operator and waits for manual intervention such as reset of the host, or a host diagnostic program that will reset the adapter and test for a specific failure symptom.

Modem Interface Section

The Sytek Serial Interface Controller SIC connects the RF modem to the Intel 82586 Local Communications Controller (LCC). The functions performed are as follows:

- Generate a 2 MHz transmit clock (TXC) for the 82586.
- Encode the transmit data (TXD) from the 82586 to the required non-return-to-zero-inverted (NRZI) format specified for the modem.
- Decode the NRZI data received from the modem to the format needed by the 82586, receive data (RXD).
- Recover the receive clock from the received data and drive the receive clock (RXC) function of the 82586.
- Perform the collision detect function while transmitting, and drive the CDT pin of the 82586.
- Drive the carrier sense (CRS) pin of the 82586.
- Detects network failures and reports the failure to the 80188 through an interrupt.
- Place the SIC in a loopback/diagnostic mode under control of the 80188 CPU.

Communications Controller Section

This section uses the Intel 82586 and the Sytek Serial Interface Controller (SIC). The 82586 manages the process of transmitting and receiving packets. On the microprocessor side, this controller operates as a bus master. This means that both the 80188 microprocessor and the 82586 can access the IBM PC Network Adapter's local memory.

There are two major control units in the 82586, the command unit and the receive unit. The two units are controlled and monitored by the microprocessor by a shared memory structure called the system control block.

- The command unit executes commands given by the microprocessor and manages packet transmissions.

- The receive unit handles all activities related to packet reception such as; buffer management, packet and address recognition, and CRC checking.

The other two memory structures used by the 82586 are the command block list and the receive packet area. The adapter memory holds the list of commands to be executed by the 82586, and all received packets. Pointers to these two structures are stored in the system control block along with the contents of the status register, the value of certain counters, and control commands for the 82586.

CSMA/CD Technique

A protocol that is widely used for broadband local networks is the carrier sense with multiple access and collision detection (CSMA/CD) method. This method is supported in hardware on the adapter. The RF modem section detects the presence of a carrier. The Sytek Serial Interface Controller (SIC) detects collisions. The 82586 Local Communications Controller (LCC) supports the higher levels of the protocol after proper configuration for slot time, back-off algorithm, retries, address filtering, data encapsulation, error detection, and other parameters.

A method of detecting collisions in a broadband network is based on a comparison between the data sent by a node and the data received after a round-trip delay to the headend. This technique cannot guarantee 100% detection of all collisions. A capture effect found in all frequency modulation systems allows the possibility that a particularly strong transmitter can capture the channel and take it away from a weaker transmitter. For power differences greater than 6 dB, the weak node backs off and the strong node assumes it has seized a quiet network. Proper cable system design effectively eliminates these undetected collisions. In any case, if an undetected collision occurs, the Link Access Protocol detects this as a CRC failure and retransmission occurs. The overall statistical behavior of the network is unaffected. When a collision has been detected by the SIC, it asserts the Collision Detected (CDT) input to the LCC.

RF Modem Section

This section describes the RF Modem and its circuitry. The modem consists of a single coax tap modulator—demodulator (modem) with a data transfer rate of 2M bits per second to and from the network. The modem transmits to and receives from the network on separate channels. Each channel has a 6 MHz bandwidth, separated by a frequency offset of 168.25 MHz. The transmit center frequency is 50.75 MHz; the receive center frequency is 219.00 MHz. Both frequences are shared on the broadband network cable by all nodes through the use of the carrier sense multiple access with collision detection (CSMA/CD) technique. These frequencies are aligned with CATV channels T–14 and J.

The following is a block diagram of the RF Modem section:

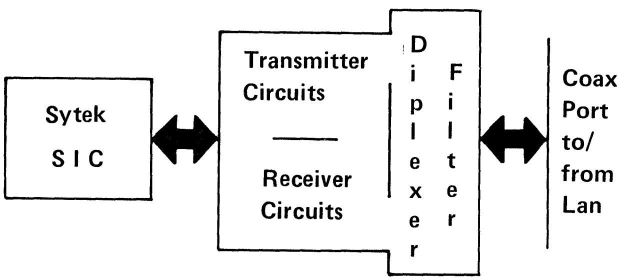


Figure 3-10 RF Modem Block Diagram

Transmitter Description

This section describes the transmitter circuits of the RF Modem. A block diagram of the transmitter section is as follows:

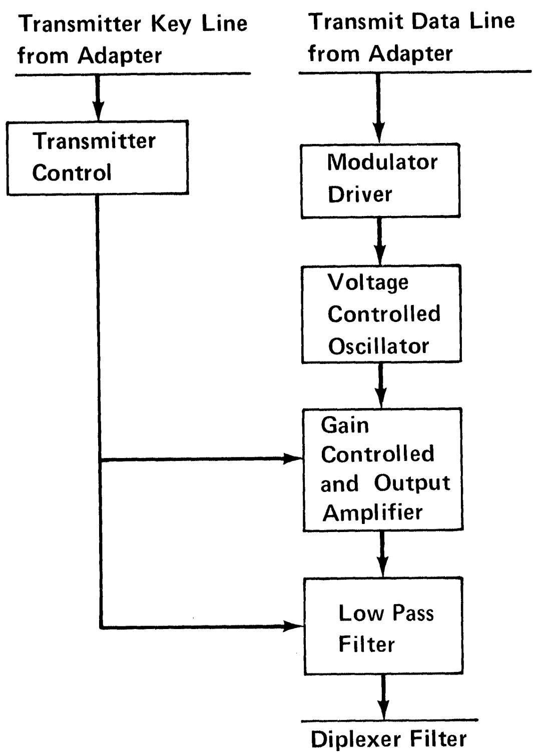


Figure 3-11 Transmitter Block Diagram

- Modulator Driver

  The NRZI-encoded serial TTL data is applied to a driver stage that converts the TTL levels into mark and space voltages. These voltages are applied to the frequency-shift key (FSK) oscillator. The driver stage also functions as a low-pass filter to remove the unwanted harmonic signals from the modulating signal. The modulation adjustment is factory set for a 2 MHz shift.

- Voltage-Controlled Oscillator (VCO)

  The mark and space voltages are used to frequency shift key the VCO. The VCO operates at 50.75 MHz. Also, the oscillator is temperature compensated to provide the necessary frequency stability.

  Within the VCO, are the following circuits:

  - Buffer/Amplifier

    The output of the oscillator is buffered with a two-stage amplifier to minimize oscillator pulling and to provide sufficient level to overcome the loss in the bandpass filter.

  - Bandpass Filter

    The bandpass filter is used to band-limit the FSK signal. It also removes unwanted sidebands and harmonic signals at the amplifier output.

- Gain-Controlled Amplifier

  The gain-controlled amplifier is factory adjusted and can provide up to 17 dB of gain. This stage is used to set the transmitter's output level.

- Output Amplifier

  The output amplifier provides 10 dB of gain and the high-level output needed to drive the cable system.

- Low-Pass Filter

The low-pass filter is used to remove the harmonic signals generated by the amplifiers and the PIN diode switch. It also forms one leg of the band separator, which combines the transmitted and received signals on the coax cable.

Transmitter Characteristics

- Output Impedance

  The output impedance in the transmit channel is 75 ohms nominal.

- Return Loss

  The return loss is greater than or equal to 14 dB in the transmit channel with either power on or off.

- Transmitter Load

  The transmitter can operate continuously into an open or short circuit without damage. Also, it can operate continuously into a cable system where other modem transmitters are enabled at the same time without damage.

- Power Level

  The transmitter output level is set to 56 dBmV ± 1 dB. The transmitter output level variation is within ± 3 dB of its initial setting.

- Frequency Allocation

  The transmitter operates on a center frequency of 50.75 MHz.

- Frequency Stability

  The frequency stability is $\pm 0.6\%$ of the transmit channel frequency.

- Modulation Technique

  The modulation technique used is frequency-shift keying (FSK).

- Frequency Shift

  The frequency shift is 2 MHz ±200 kHz centered about 50.75 MHz.

- Carrier Turn-on Delay

  The transmitter can reach 90% of full output power within 3 to 9 μs of TX Key going low.

- Carrier Turn-off Delay

  The transmitter can reach 10% of full output power within 3 to 9 μs of TX Key going high.

- Envelope Overshoot

  The maximum envelope overshoot is 25% during turn on and off.

- Off Condition Output

  In the off state, the carrier signal level is -20 dBmV or less.

- Spurious Output Levels

  - From 5–10 MHz and from 100–900 MHz, the spurious output levels are -10 dBmV or 60 dB down from the carrier level; whichever is the greater signal level.

  - From 10–100 MHz, the spurious output levels are -21dBmV or 78 dB down from the carrier level; whichever is the greater signal level.

- Spectrum Shape (Bandwidth)

The modulated output spectrum is greater than or equal to 40 dB down at ±3 MHz from the center frequency. The out-of-band power is 40 dB or more below reference carrier level at ±4 MHz from the center frequency.

Receiver Description

This section describes the receiver circuits of the RF Modem. A block diagram of the receiver section is as follows:

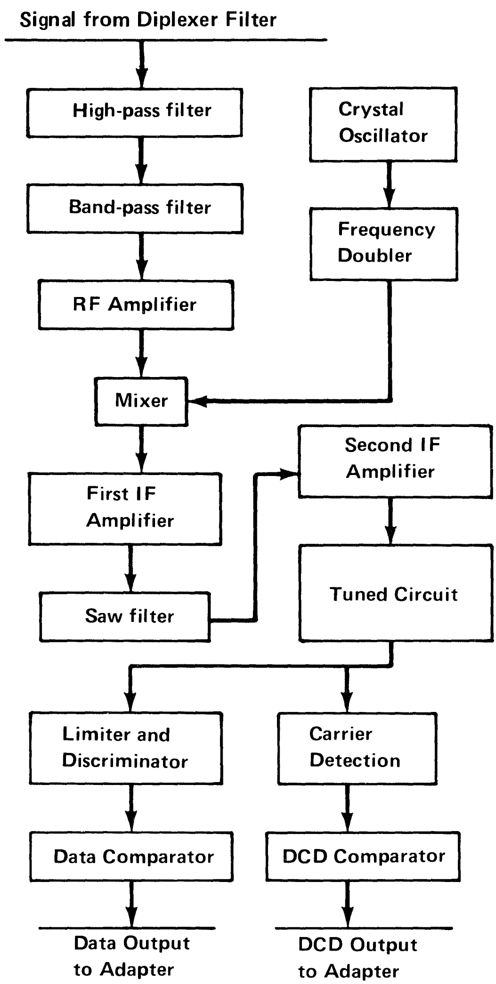


Figure 3-12 Receiver Block Diagram

- High-Pass Filter and One-Half Diplexer Filter

  The incoming signal to the modem is separated from the outgoing signal by the high-pass filter.

- Bandpass Filter

  The bandpass filter passes the 6 MHz bandwidth that is centered on the 219 MHz frequency.

- RF Amplifier

  The RF amplifier increases the signal level to compensate for the loss in the bandpass filter.

- Mixer

  The mixer combines the incoming 219 MHz signal with the 179.5 MHz local oscillator to provide a 39.5 MHz intermediate frequency (IF).

- Crystal Oscillator

  The crystal oscillator starts the local oscillator chain at 89.75 MHz.

- Doubler

  The doubler multiplies the output of the crystal oscillator from 89.75 MHz to 179.5 MHz to provide the local oscillator injection for the mixer.

- First IF Amplifier

  The first IF amplifier provides about 23 dB of gain to compensate for the loss through the saw filter.

- Saw Filter

  The saw filter selects signals only within the desired IF pass-band and attenuates signals outside this band.

- Second IF Amplifier

  The second IF amplifier provides 30 dB of gain needed for the necessary input level for the limiter/demodulator circuit.

- Tuned Circuit

  The tuned circuit provides impedance matching and increases out-of-band attenuation.

- Limiter/Discriminator

  The limiter/discriminator provides amplitude limiting of the input signal and demodulates the FSK signal with a quadrature detector. The limiter also provides a relative signal strength indication to operate the data carrier detect (DCD) comparator.

- Data Comparator

  The data comparator converts the analog signal into a TTL-compatible digital signal.

- DCD Comparator

  The DCD comparator produces a TTL-compatible output signal to indicate the presence of an incoming signal.

Receiver Characteristics

- Input Impedance

  The standard impedance is 75 ohms nominal.

- Return Loss

  The return loss is 14 dB or more in the receive channel with power on. With power off, the return loss is 8 dB or more in the receive channel.

- Frequency Allocation

  The receiver operates on a center frequency of 219.0 MHz.

- Frequency Stability

  The frequency stability after a 10-minute warm-up period is within 0.01% of the receive channel frequency.

- Reception Acceptance Range

  The receiver demodulates incoming signals within ±0.177% of the receive channel center frequency.

- Channel Bandwidth

  The receiver has a 3.6 MHz channel bandwidth.

- Sensitivity

  The normal input level is +8.5 dBmV. Operating range is -7 dBmV to +24 dBmV. The input sensitivity for 20 dB S/N at the demodulator output is less than or equal to -13 dBmV. Minimum quieting at 0 dBmV is 30 dB. The maximum sustained input level without damage to the receiver is +61.25 dBmV.

- Recovered Data

  The jitter on the demodulated data out of the receiver data comparator does not exceed ±150 ns with an input signal of -7 dBmV modulated with a 1 MHz square wave.

- Receive Carrier Detect

  The carrier detect threshold is between -20 and -8 dBmV. The DCD line will go true within 4.0 μs of a -7 dBmV signal being present at the input of the modem. The rise and fall time for the output of the DCD comparator from 10% to 90% of the steady state output is less than or equal to 500 ns.

Adapter Interface Signals

Each input line to the IBM PC Network Adapter presents a maximum of one LS TTL load to the Personal Computer bus. All IBM PC Network Adapter output signals to the Personal Computer bus are driven by tri-state drivers.

Without WAIT states being generated, all Personal Computer processor-generated memory read/write cycles take four time (T) states. All Personal Computer processor-generated I/O read/write and DMA transfers require five T states. See your *IBM Technical Reference* for more information about your computers DMA transfers.

The following Personal Computer interface signals are used by the IBM PC Network Adapter.

Signal	I/O	Description
A0-A19	I	These lines are used to address the BIOS memory and adapter I/O interface registers.
D0-D7	I/O	These lines provide a bidirectional data bus for the Personal Computer processor, Personal Computer memory, and adapter.
ALE	I	Address latch enable, which is used as an indicator of a valid Personal Computer processor address to the adapter.
IOR__	I	This command line instructs the adapter to drive its data onto the Personal Computer data bus.
IOW__	I	This command line instructs the adapter to read the data from the Personal Computer data bus.
MEMR__	I	This command line instructs the BIOS ROM to drive its data onto the Personal Computer data bus.
IRQ2 and IRQ3	O	Interrupt request 2 or 3 is used to signal the Personal Computer processor that the adapter requires attention.

Figure 3-13 (Part 1 of 2). Personal Computer/ Adapter Interface Signals (from Adapter)

Signal	I/O	Description
DRQ3	O	DMA Request 3 is used by the adapter to gain DMA service from the Personal Computer.
DACK3__	I	DMA Acknowledge 3 is used to acknowledge DRQ3 which is requested by the adapter.
I/O CH RDY	O	I/O Channel ready is used to allow the Personal Computer to generate WAIT states to extend the Personal Computer clock cycle up to a maximum of 2.1 μsec.
RESET DRV	I	This line is used to reset or initialize the adapter logic upon power-up or after a low line voltage outage.
T/C	I	Terminal Count: This line provides a pulse that is gated with DACK3__ which may generate a Personal Computer interrupt request whenever the terminal count for the Personal Computer's DMA channel 3 is reached.

Figure 3-13 (Part 2 of 2) Personal Computer/ Adapter Interface Signals (from Adapter)

Power-On Self-Tests (POST)

The IBM PC Network Adapter provides a pass/fail indication to the Personal Computer, as long as the interface is functioning. If the adapter is functioning but the adapter-Personal Computer interface fails, the error is posted at power-on time. The adapter responds to any requests for remote status from other computers on the network indicating that a failure has occurred.

The following is a list of tests, in order, that are performed by the adapter.

1. Microprocessor Self-test

   The 80188 microprocessor performs limited self-test of its functions and certain peripheral circuits that are integrated within the 80188 microprocessor.

   The tests are conditional jump test and register write test. The conditional jump test verifies the proper execution of all conditional jump instructions when the corresponding status flags are in the set/reset conditions. The register write test verifies the entire register set by writing and then chain-copying a certain data pattern (and its complement) to all registers, and then reading and comparing the register's final contents with the original data pattern.

   The peripheral tests are interrupt mask test, spurious interrupt test, and timer 1 test. The interrupt mask test performs the write, read and compare operations to the Interrupt Mask Register (IMR) by using both all ones and all zero patterns. The spurious interrupt test verifies that no spurious interrupts are generated by the hardware when all interrupt masks are off. The timer 1 test verifies the proper operation of timer 1 and timer interrupt. Timer 1 is used as the 10-ms clock in the system.

If the adapter fails the microprocessor self-test, it will execute a halt instruction.

2. ROM Checksum

   Following the microprocessor self-test, a ROM checksum is performed using simple modulo addition expecting an all zeros result. This assumes that a precalculated checksum byte stored in the ROM will provide a zero result.

   The adapter executes a halt instruction, if the results of the addition are not correct.

3. Unit ID PROM Test

   Following the ROM checksum test, the unit ID PROM is tested by verifying that the byte at location 1AH in the PROM has the value of 00H. The unit ID is also checked for having a low bit of zero (because it must be even).

4. RAM Test

   Once the microprocessor, PROM, and ROM have passed their tests, the RAM is tested and also the RAM refresh is tested. The dynamic RAM is refreshed by DMA 1, which is driven by timer 2 periodically. The RAM test verifies the operation of the RAM, DMA 1, timer 2, and timer interrupt.

   If the RAM is not functional, the adapter cannot perform its intended application. An error is reported by using the adapter-Personal Computer interface. Since both digital and analog cable loopbacks require functioning RAM, the adapter does not perform these tests and continues on to the adapter-Personal Computer interface test reporting that the RAM test failed.

5. Host Interface Tests

This test is functionally separated into five sub-tests. The first portion of the Host Interface register test is a stand-alone test that requires no involvement from the host software. The subsequent tests require synchronization and cooperation with the host software. The five sub-tests are as follows:

a. Host interface register test

b. GO interrupt test

c. Data transfer interrupt test

d. Data transfer DMA test

e. Host interface control test

The adapter requires an initial synchronization from the BIOS ROM. Within 500 ms. of the host's clearing the reset bit, the BIOS sets the CC1 and GO bits in the Status register.

a. Host Interface Register Test

   This test verifies the various functions and characteristics of the Data register, the Parameter register, the Adapter Interface register and the Status register. The Host Interface register is not accessible from the adapter. Therefore, the testing of the Host Interface register is done by the BIOS.

   Proper operation of the Data register is verified by writing, reading, and comparing a AAH/55H data pattern in a certain sequence. In addition, the DRF and DRE flags are verified accordingly.

   Proper operation of the Parameter register is tested by writing, reading, and comparing with a AAH/55H/xxH/xxH/xxH/xxH/xxH data pattern where (xx) is a don't care value.

Proper operation of the Adapter Interface register is verified by writing, reading, and comparing with a AAH/51H data pattern. In addition, the CQF flag is tested accordingly.

Proper operation of the Status register is tested by writing a 05H (and then a 02H) to the Status register, reading the Status register and testing for 15H (and then 12H). Note that the write with 02H provides a means to synchronize with the host software.

b. GO Interrupt Test

This test verifies that the GO interrupt is sent by the host and is received by the adapter. This test also verifies that the GO interrupt is sent by the adapter and is received by the host. This test requires cooperation with the host's software.

c. Data Transfer Interrupt Test

This test verifies that the data transfer interrupts (both read and write interrupts) are received by the adapter when the Status register and Adapter Interface register are properly configured. This test also verifies that the data transfer interrupts are received by the host. This test requires cooperation with the host's software. Note that this test verifies the interrupt mechanism, not the integrity of the data being transferred through the interrupts.

d. Data Transfer DMA test

This test verifies that two data bytes are transferred to the adapter by DMA with a DMA interrupt. This test also verifies that the two data bytes are returned to the host with the TC interrupt. The host's software is required to test and validate the data bytes returned from the adapter to ensure the integrity of the data transfers. This is performed by the BIOS after a reset.

e. Host Interface Control Test

This test verifies the various functions that are related to the ownership of the interface. Functions tested include the following:

1) The adapter's inability to acquire control of the interface when the interface is controlled by the host.

2) The adapter's ability to acquire control of the interface after the host relinquishes control of the interface.

3) The host's ability to acquire control of the interface after the adapter relinquishes control of the interface.

4) The host's inability to acquire control of the interface when the interface is controlled by the adapter.

5) Generation of the Host Control interrupt and the Host Relinquish interrupt. This test requires cooperation with the host's software.

6. +12 Volt and -12 Volt Presence Test

This test checks the presence of the Personal Computer's +12 V and -12 V power supplies. When either of the two supplies are below the sense level voltage, a '± 12V not present' condition is provided to the IBM PC Network Adapter microprocessor. When both of the supplies are above the sense level voltage, a '± 12V present' condition is provided to the IBM PC Network Adapter microprocessor.

7. Digital Serial Loopback

   After first initializing the 82586, a test is performed using the loopback points of the SIC integrated circuit. The two circuits tested are the integrated circuit and the adapter interface circuits. The following tests or error states are created and tested in loopback mode.

   - Diagnose Command of the 82586
   - No Error Packet
   - Short Frame
   - CRC Error

   The adapter cannot perform its function if the digital serial interface is not functioning. The analog cable loopback test cannot be performed if the digital serial interface is not working properly. The adapter reports the status of a failure to the Personal Computer.

8. Analog Cable Test

   At this point, all tests have been performed independent of external equipment and support. To perform the analog cable loopback test, a frequency translator is needed to provide frequency translation. The test also assumes that the cable can contain active functional traffic, so the adapter must respect the cable protocol. The adapter tries to send a test packet addressed to itself. Because collisions might occur, this test will try eight times to send a packet and receive it back. If the analog self-test fails, the adapter reports a cable loopback test failure to the Personal Computer. At this point, the adapter may have errors, but it functions normally with respect to the Personal Computer.

A failure at this point could be because of a problem in the RF Modem section or an external failure. The test fails if eight consecutive tries end in collisions or the test packet did not return. The adapter reports the status of a failure to the Personal Computer.

Operational Self-Test

This test is run under normal operation of the adapter and will generate a return code posting the error.

Constant Carrier Detection—The Sytek SIC contains circuitry to detect a constant carrier and to inform the IBM PC Network Adapter processor. The adapter processor causes the 82586 to abort any transmitting packet in progress. The constant carrier status is reported to the Personal Computer by the primary command Error Report.

Configurable Hardware Options

This adapter contains six configurable jumper positions.

- Jumper W1 is the remote program load (RPL) feature. Removing the jumper enables the feature.

- Jumper W2 is a reserved jumper.

- Either jumper W3 or W4 is used to select IRQ. The interrupt must be different from any adapter in your computer. With the jumper in the W3 position, interrupt level 2 is selected. With the jumper in the W4 position, interrupt level 3 is selected.

- One jumper W6 on the center pins of W5 and W7 selects the high I/O base address. Use two jumpers, W5 and W7, to select the low I/O base address. See Figure 3-2 on page 3-26 for the addresses of the registers.

- One jumper W8 is used to disable or enable the BIOS ROM. When the jumper is installed, the BIOS ROM is enabled.

The state of jumpers W1 and W2 is reported back to the Personal Computer. See network control block ADAPTER STATUS in Chapter 2 for more information.

The following figure illustrates the jumper positions on the adapter.

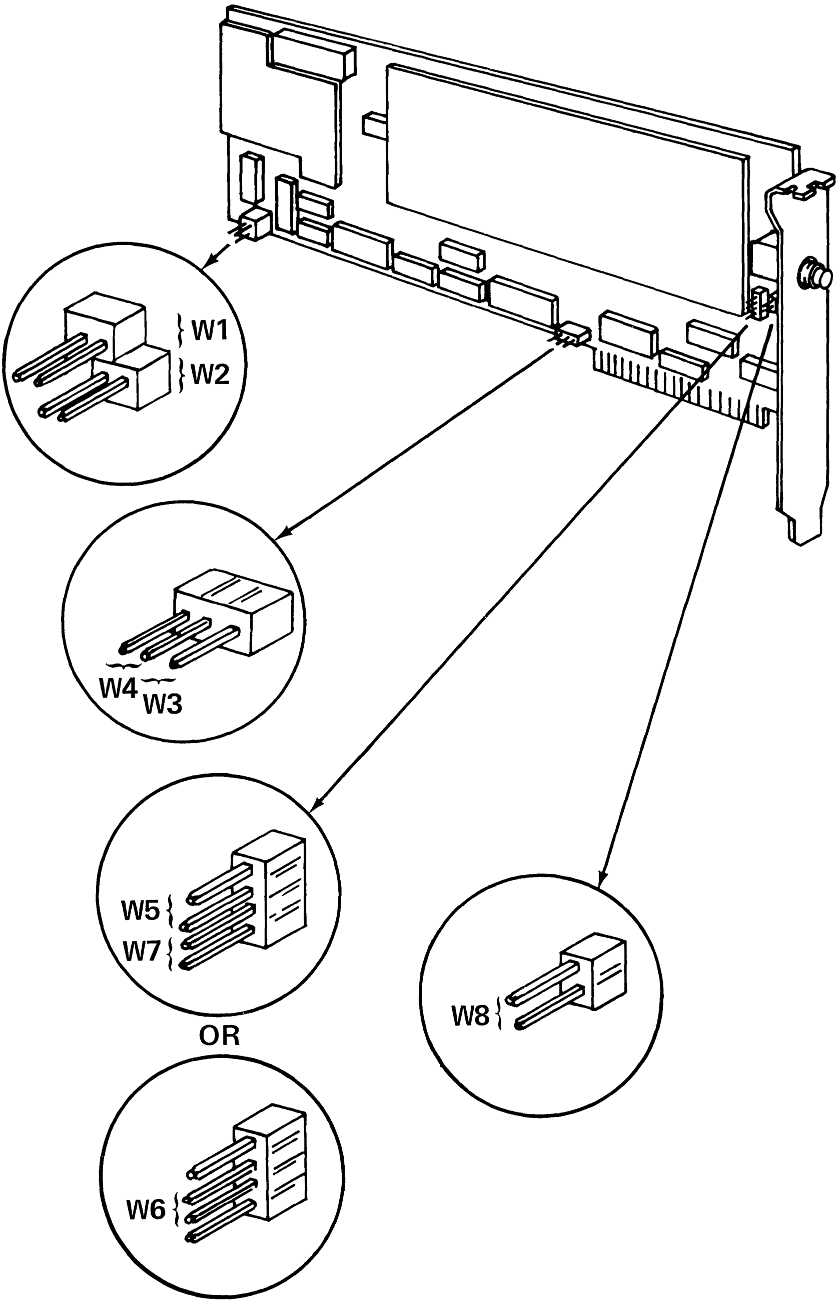


Figure 3-14 Adapter Jumper Positions

Traffic And Error Statistics

The adapter keeps and reports on demand some of the following statistics:

- Duration of reporting period
- Quantity of CRC errors received
- Quantity of alignment errors received
- Quantity of collisions encountered
- Quantity of aborted transmissions
- Quantity of successfully transmitted packets
- Quantity of successfully received packets
- Number of times the receiver exhausted its resources

The reporting period for the IBM PC Network Adapter is from last reset as expressed in minutes. No provision is made for resetting the statistics other than a Personal Computer reset of the adapter or power-up of the Personal Computer. See network control block ADAPTER STATUS in Chapter 2 for more information.

Specifications

This section summarizes basic specifications of the hardware of the IBM PC Network Adapter.

Electrical Power Requirements

The specifications of the power requirements are as follows.

Voltage	Tolerance	Ripple	Total Current Used
+12.0V	±5%	100 mV pp	0.36 A
+5.0V	±5%	100 mV pp	1.40 A
-12.0V	±10%	100 mV pp	0.03 A

Environmental Specifications

Temperature The operating temperature range is from 10 to 35°C, (50 to 91°F) ambient. The storage temperature range is from -40 to 60°C, (-40 to 140°F).

Humidity The operating humidity range is from 8% to 80% non-condensing. The storage humidity range is from 5% to 100% non-condensing.

Altitude The operating altitude is -305 to 2135 meters (-1000 to 7,000 feet).

IBM Translator Unit

This section describes the specifications of the translator unit for the IBM PC Network

The translator unit provides the basic frequency translation and amplification required in a broadband network. A single translator unit can serve a network comprising many local area network adapters and their attached devices.

Device Description

This section briefly describes the translator unit and discusses its functions as shown on a block diagram.

This translator unit is implemented on a printed circuit board that fits inside an enclosure designed to meet FCC Class B. The enclosure also includes the required power supply circuits. The translator unit is designed for high reliability, and has no "field" adjustments.

Functional Description

The function of this unit is to translate a channel with an input center frequency of 50.75 MHz from the network into a channel with an output center frequency of 219 MHz, with the required spectral purity and signal level. The entire 6 MHz channel is translated from the lower band to the upper band.

The following is a block diagram of the IBM Translator Unit.

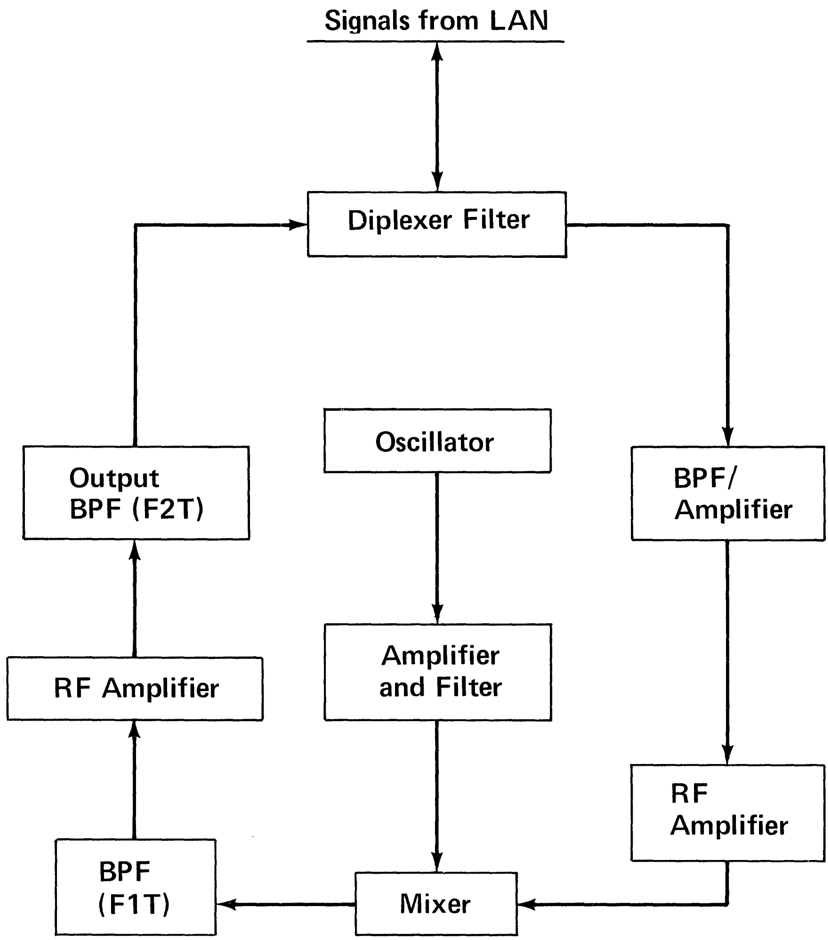


Figure 3-15 Translator Unit Block Diagram

The block diagram can be divided into four main parts discussed in the following sections:

- Input/Output circuits
- Reception circuits
- Local oscillator circuits
- Transmission circuits

Input/Output Circuits

Diplexer Filter A conventional low-loss band separator with a stop-band attenuation of 25 dB. Its function is to prevent the transmitted signals from entering the reception path, and to limit the amount of unwanted signals entering the translator.

Reception Circuits

Bandpass Filter F1R Has a center frequency of 50.75 MHz and a bandwidth of approximately 6 MHz. This device filters out most out-of-band signals, lowering the input intermodulation requirements of the amplifier that follows it.

RF Amplifier A 25 dB low-noise amplifier providing the required signal level to the mixer, for best intermodulation and low loss performance. Its output is matched to the mixer's 50 ohm impedance.

Mixer — A high-performance double-balanced mixer. Its local oscillator tap requires a +7 dBm signal level for optimal operation.

Local Oscillator Circuits

Crystal Oscillator — A common emitter oscillator whose frequency is controlled by a high-stability crystal. This oscillator provides the required frequency stability for the unit.

Amplifier and Filter — Provides a +7 dBm signal to the local oscillator's mixer tap with the required spectral purity for having the lowest spurious level from the mixer.

Transmission Circuits

A typical output signal level of +50.25 dBmV is provided to the trunk, after a translator gain of 36 dB (typical).

Bandpass Filter F1T — Removes unwanted out-of-band products from the mixer's output, and matches the amplifier's input impedance to the mixer's 50 ohm impedance.

RF Amplifier — Amplifies the signal to the required output level, providing

25 dB gain. It is based on a low-distortion solid state design.

Bandpass Filter F2T

Removes unwanted products generated in the amplifier and provides the ultimate attenuation of all unwanted out-of-band signals reaching it.

The IBM PC Network Cable System

This section describes the cable system components used in the IBM PC Network. The connection hardware and kits used in the IBM PC Network are compatible with broadband cable TV components. Signal levels are predesigned to provide the necessary tolerance for each IBM PC Network Adapter. A fully configured network, using IBM components, can support 72 nodes with a maximum radius of 1000 feet.

Cable System Components

The cable system consists of six components:

- The Translators Unit's connection hardware
- Base Expander
- Short Distance Kit
- Medium Distance Kit
- Long Distance Kit
- IBM coaxial cable in either 25, 50, 100, or 200 feet increments

Translator Unit's connection hardware

These components allow attachment of up to eight computers to the Translator Unit. A directional coupler is provided within the components to allow connection for the IBM Base Expander.

Base Expander

When this component is attached to the connection hardware, it allows connection for up to eight Short, Medium, or Long Distance Kits. Signal levels at the taps on the Base Expander are not compatible with the Adapters. A Test Tool is provided for diagnostic purposes.

Short Distance Kit

This kit attaches to any of the eight taps on the Base Expander. The kit allows you to connect up to eight computers.

Medium Distance Kit

This kit attaches to any of the eight taps on the Base Expander through an additional 400 feet of cable. The kit provides connection for up to eight computers.

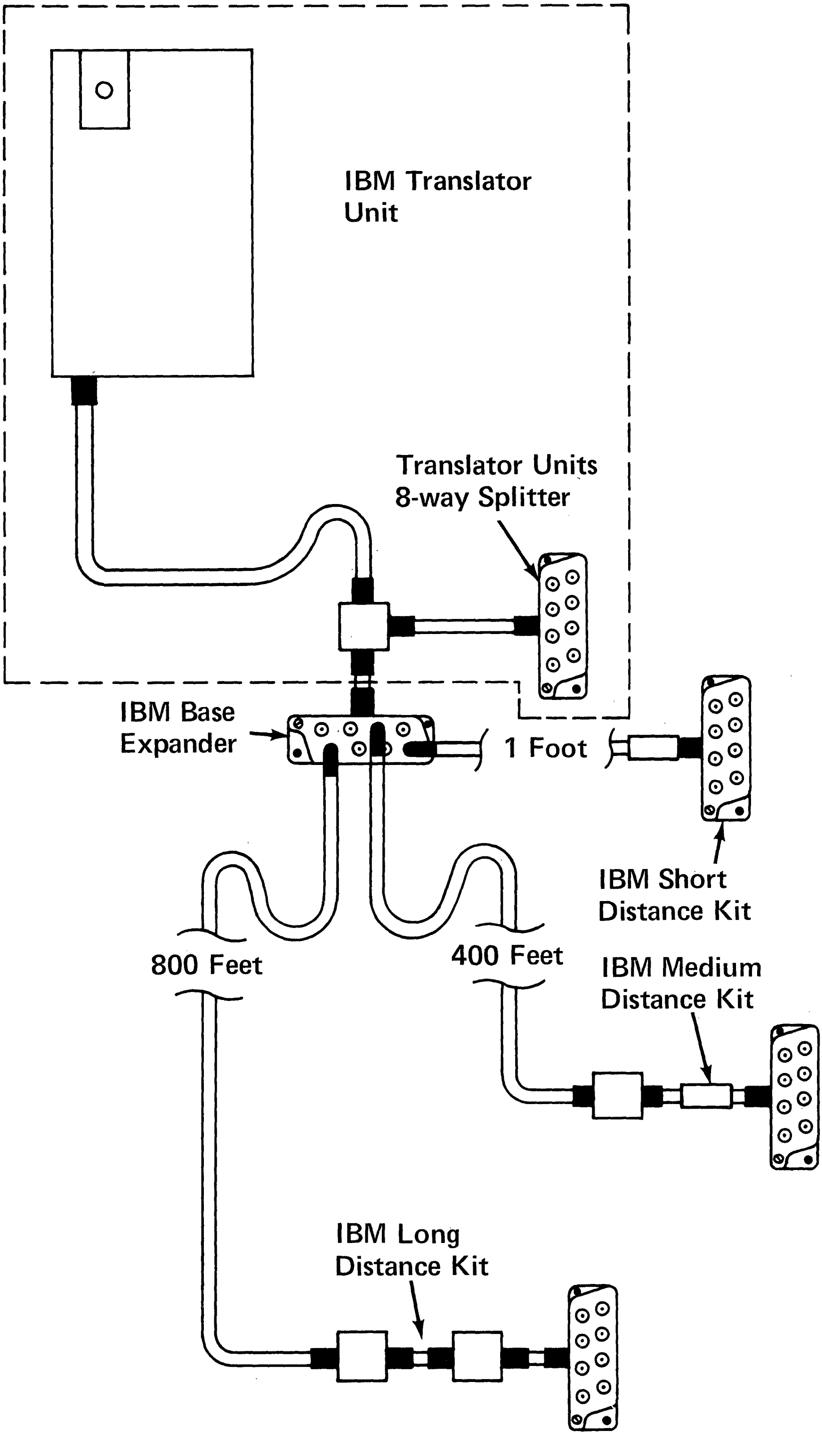
Long Distance Kit

This kit attaches to any of the eight taps on the Base Expander Kit through an additional 800 feet of cable. The kit provides connection for up to eight computers.

IBM Coaxial Cable

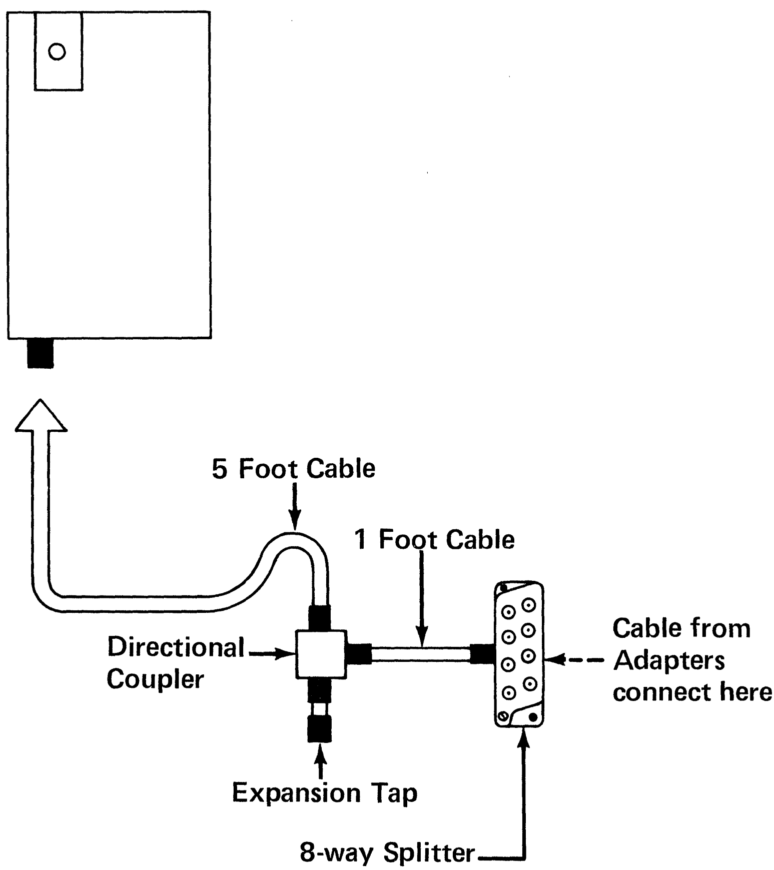
The cable is standard RG-11 type coaxial cable providing different lengths: 25, 50, 100, and 200 feet. These cable increments can be combined to provide the 400 and 800 foot lengths required by the Medium or Long Distance Kits. For the 400 feet length, you must use either four 100 foot lengths or two 200 foot lengths. For the 800 foot length, you must use four 200 foot lengths. In addition, up to 200 feet of cable can be installed between the kit and each computer on the network. For the 200 foot lengths, you must not use eight 25 foot cables.

The following figure illustrates how the previously described components are connected together.


IBM Translator Unit
Translator Units 8-way Splitter
IBM Base Expander
1 Foot
IBM Short Distance Kit
800 Feet
400 Feet
IBM Medium Distance Kit
IBM Long Distance Kit

Connection Hardware

The connection hardware consists of a 5 foot black RG-6 cable, a directional coupler, a one foot beige RG-6 cable, and an 8-way splitter. The taps on the 8-way splitter provide the signal levels compatible with the adapters. The expansion tap on the directional coupler provides an unattenuated signal for attachment of a Base Expander. If a tap on the 8-way splitter is to be used, the terminator must be removed. If a tap is to be discontinued, the terminator must be replaced.



Electrical Specifications

–Impedance, (any node)	75 ohms nominal
–Attenuation, (forward path) *	39.9 dB ±1.5
–Attenuation, (reverse path) *	39.9 dB ±1.5
–Insertion loss, (forward path) **	0.7 dB maximum
–Insertion loss, (reverse path) **	0.5 dB maximum
–Isolation, node to node	18 dB minimum (forward and reverse paths)
–Return loss, (any node)	14 dB minimum (forward and reverse paths)

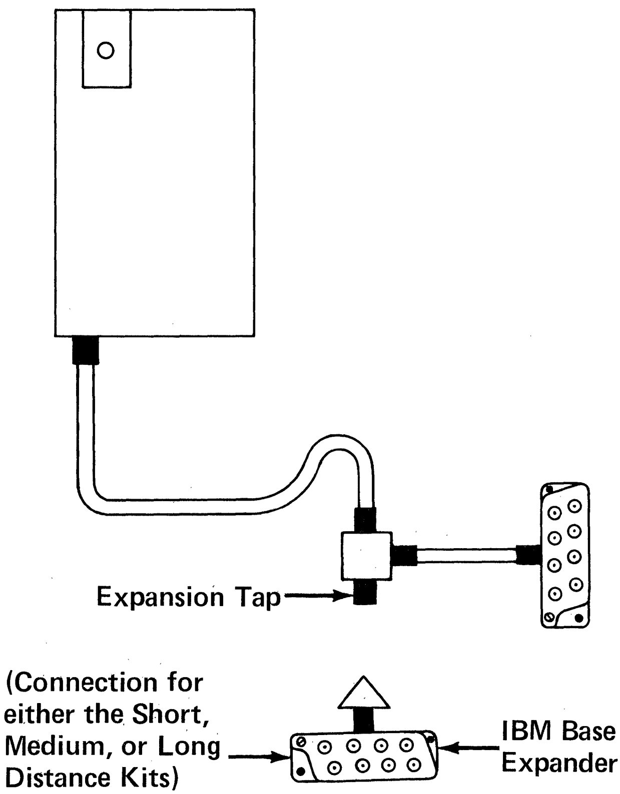
Note: Forward path is 219 MHz. Reverse path is 50.75 MHz.

* Translator Unit tap to any tap on the 8-way splitter.

** Translator Unit tap to the expansion tap.

Base Expander

This component consists of an 8-way splitter and a male adapter. The taps on the 8-way splitter provide signal levels compatible with either the Short, Medium, or Long Distance Kits. The signal levels at the taps of the splitter have not been attenuated enough to allow attachment of an adapter. A 30 dB attenuator is provided as a test connector for diagnostic purposes. The test tool allows connection for an adapter to the Base Expander.



Note:

For the Medium and Long Distance kits, 400 or 800 feet of cable must be installed between the kit and the Base Expander.

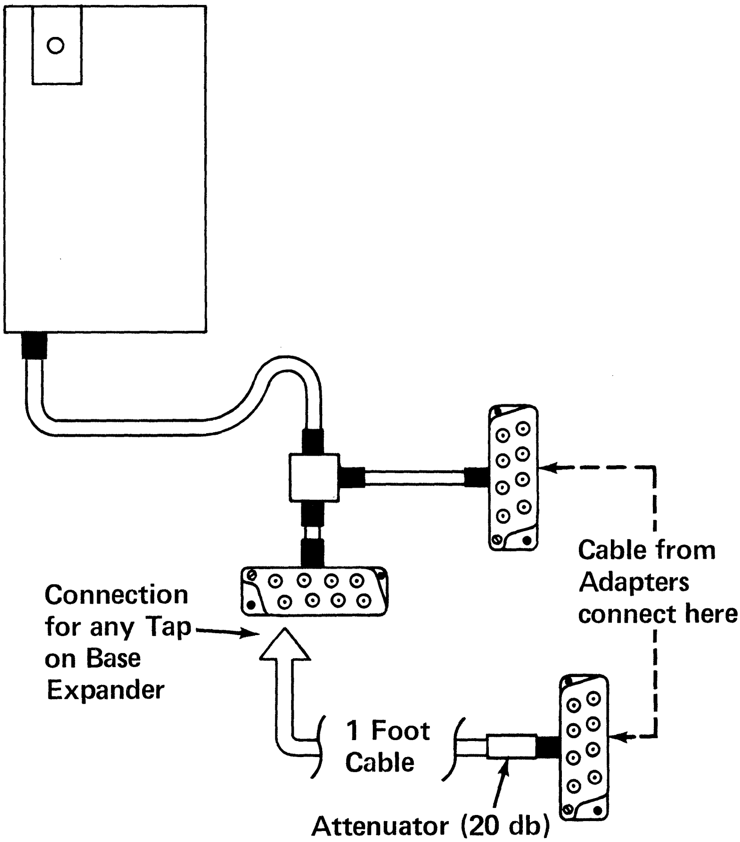
Electrical Specifications

–Impedance, (any node)	75 ohms nominal
–Attenuation, (forward path)	9.5 dB ±0.5
–Attenuation, (reverse path)	9.5 dB ±0.5
–Isolation, (node to node)	18 dB minimum (Forward and reverse paths)
–Return loss, (any node)	14 dB minimum (Forward and reverse paths)

Note: Forward path is 219 MHz. Reverse path is 50.75 MHz.

Short Distance Kit

This kit consists of a 1 foot beige RG-6 cable, a 20 dB attenuator, and an 8-way splitter. The taps on the 8-way splitter provide signal levels compatible with the adapters.



Note:

Up to 200 feet of cable may be installed between the Adapter and the 8-way splitter.

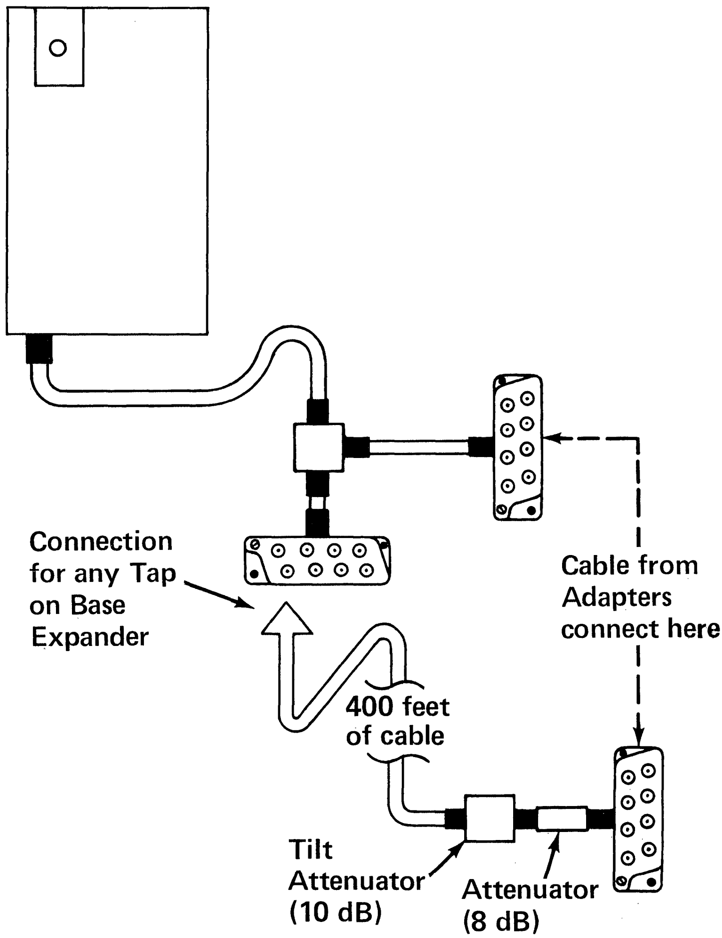
Electrical Specifications

–Impedance, (any node)	75 ohms nominal
–Attenuation, (forward path)	29.8 dB ±1.0
–Attenuation, (reverse path)	29.6 dB ±1.0
–Isolation, (node to node)	18 dB minimum (forward and reverse paths)
–Return loss, (any node)	14 dB minimum (forward and reverse paths)

Note: Forward path is 219 MHz. Reverse path is 50.75 MHz.

Medium Distance Kit

This kit consists of a 10 dB tilt attenuator, an 8 dB attenuator, a 1 foot beige RG-6 cable, and an 8-way splitter. The taps on the 8-way splitter provide signal levels compatible with the adapters. The 10 dB tilt attenuator compensates for the attenuation versus frequency characteristics of the 400 foot length of cable between the Medium Distance Kit and the Base Expander.



Note:

Up to 200 feet of cable may be installed between the Adapter and the 8-way splitter.

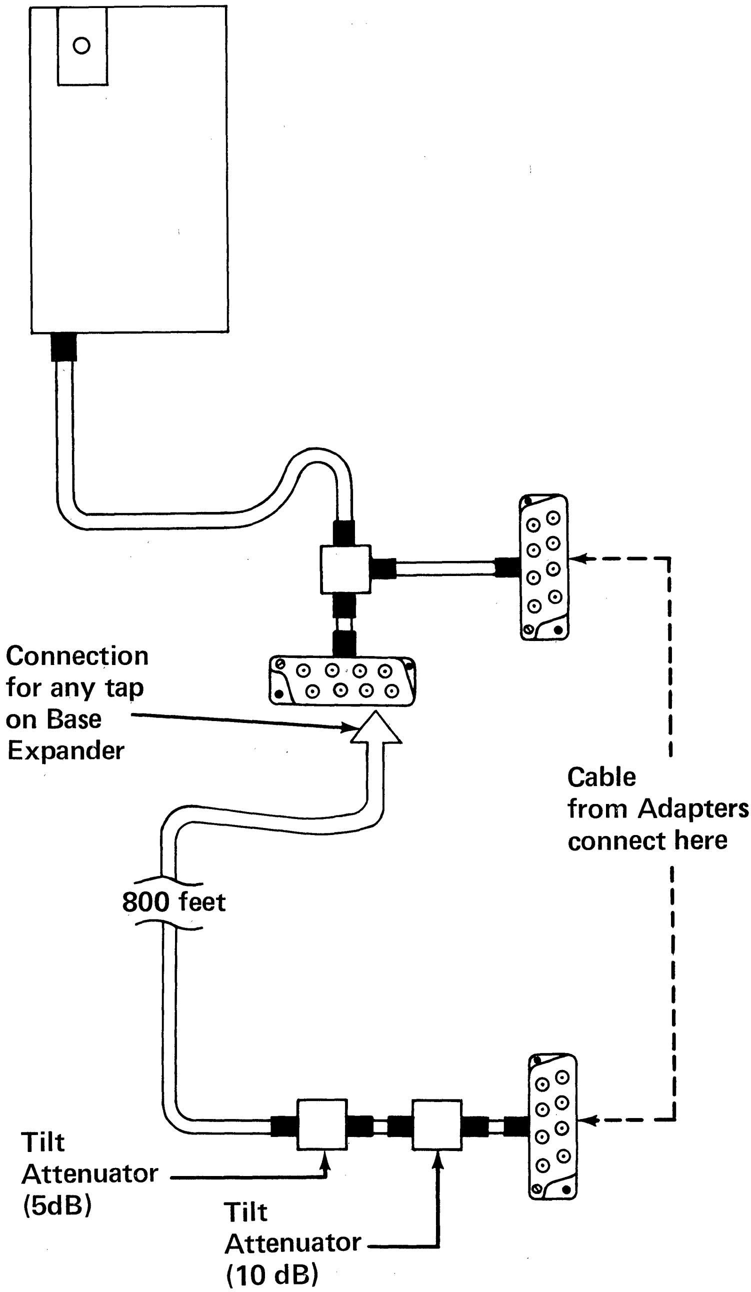
Electrical Specifications

–Impedance, (any node)	75 ohms nominal
–Attenuation, (forward path)	19.7 dB ±1.1
–Attenuation, (reverse path)	26.6 dB ±1.8
–Isolation, (node to node)	18 dB minimum (forward and reverse paths)
–Return loss, (any node)	14 dB minimum (forward and reverse paths)

Note: Forward path is 219 MHz. Reverse path is 50.75 MHz.

Long Distance Kit

This kit consists of a 5 dB tilt attenuator, a male adapter, a 10 dB tilt attenuator, a 1 foot beige RG-6 cable and an 8-way splitter. The taps on the 8-way splitter provide signal levels compatible with the adapters. The 5 and 10 dB tilt attenuators compensate for the attenuation versus frequency characteristics of the 800 foot length of cable between the Long Distance Kit and the Base Expander.



Note:
Up to 200 feet of cable may be installed between the Adapter and the 8-way splitter.

Electrical Specifications

–Impedance, (any node)	75 ohms nominal
–Attenuation, (forward path)	12.7 dB ±0.8
–Attenuation, (reverse path)	23.0 dB ±1.8
–Isolation, (node to node)	18 dB minimum (forward and reverse paths)
–Return loss, (any node)	14 dB minimum (forward and reverse paths)

Note: Forward path is 219 MHz. Reverse path is 50.75 MHz.

IBM Coaxial Cable

All cable lengths are RG-11 coaxial cable with male connectors on both ends of the cable. One end has a female-to-female adapter and a 75 ohm terminator attached to the male connector. Cables can be combined by removing the 75 ohm terminator and joining the male end of one cable to the female adapter of the other. When building the 400 and 800 foot lengths of cable required by the Short and Medium Distance Kits, use 200 foot increments of cable to limit the number of cable connections required. When building cable lengths for installation between an Adapter and an 8-way splitter, no more than 3 cables should be combined.

Cable Characteristics

The IBM Cable has the following characteristics:

- RG-11
- Copper-covered steel-center conductor
- Gas-expanded polyethylene dielectric
- Inner shield of aluminum-polypropylene-aluminum laminated tape bonded to the dielectric
- #34 AWG bare aluminum braid wire
- Non-bonded aluminum-polypropylene-aluminum tape
- #34 AWG bare aluminum braid wire
- Jacket of polyvinylchloride
- Nominal outside diameter 0.405 inch (10.29 mm).

Electrical Specifications

–Impedance	75 ohms nominal
–Attenuation (at 20° C per 100 ft.)	
5 MHz	0.29 dB
55 MHz	0.96 dB
83 MHz	1.18 dB
187 MHz	1.75 dB
211 MHz	1.90 dB
250 MHz	2.05 dB
300 MHz	2.25 dB
–Return Loss (5–300 MHz)	30 dB minimum

Cable Network Specifications

- Channel Assignments

Return Channel T-14	50.75 MHz
Forward Channel J	219 MHz

- RF Connector

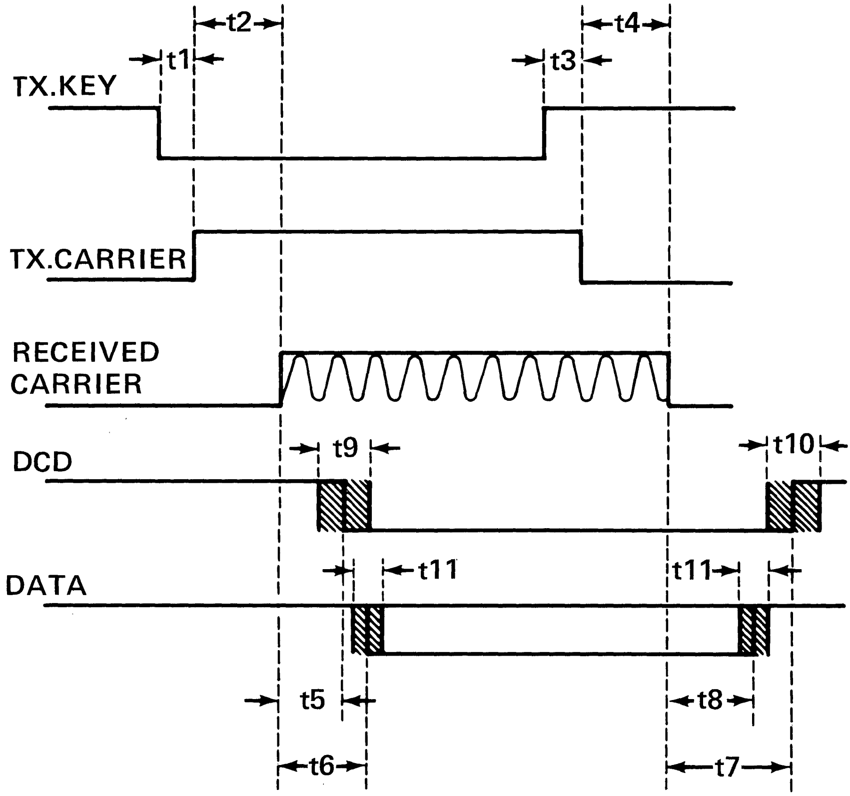
  The RF connector is a type F connector. The RF connector is grounded on the adapter and is grounded on the Translator Unit.

- RF Input/Output

  The RF input/output is transformer coupled.

- RF Modem Signal Timing

  Figure 3-16 shows typical modem signal timing.



t1 = 1.5 μ sec
t2,t4 = Channel Propogation Delay
t3 = 1 μ sec
t5 = 1.3 μ sec
t6,t8 = 1.3 μ sec

t7 = 3.5 μ sec
t9 = ± 1 μ sec
t10 = ± 1.75 μ sec
t11 = ± 150 μ sec

Figure 3-16 RF Modem Typical Timing Diagram

- Error Rate Versus S/N Ratio

  The minimum input for a bit error rate of 1 in 10^8 is -7 dBmV with an input S/N ratio of at least 33 dB.

  If these minimum conditions are met, then the IBM PC Network adapter has a bit error rate of less than 1 error in 10^{13} bits after CRC detection and retry.

- Complete Cable Network overview

Figure 3-17 shows the complete network signal losses and attenuation values. See Appendix B for the complete network specifications.

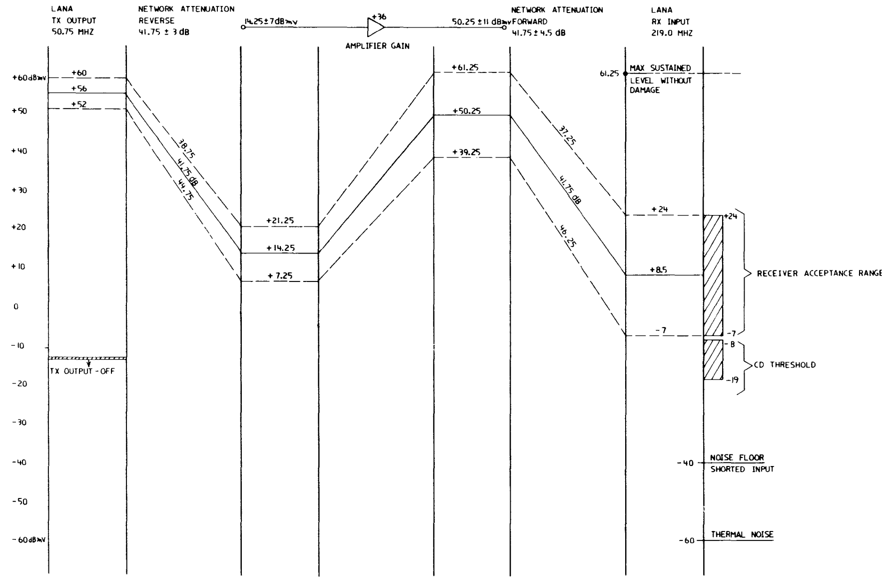


Figure 3-17 Complete Cable Network

Electrical Specifications

The following specifications apply to any version of the cable network, from a minimum 8-node configuration to a maximum configuration of 72 nodes.

–Impedance, (any node)	75 ohms nominal
–Attenuation, (forward path) *	41.75 dB ±4.5
–Attenuation, (reverse path) *	41.75 dB ±3.0
–Isolation, (node to node)	18 dB minimum (forward and reverse paths)
–Return loss, (any node)	14 dB minimum (forward and reverse paths)

Note: Forward path is 219 MHz. Reverse path is 50.75 MHz.

Note: (*) Includes all cable between the Translator Unit and the Adapter.

Notes:

Notes:

Chapter 4. Network Design

Contents

Introduction

This chapter discusses how to design a Local Area Network using the IBM Translator Unit, and IBM PC Network Adapters within each of the Personal Computers. Most of this information deals with the use of either the Short, Medium, or Long Distance Kits and how to combine components in a facility to connect your Personal Computers. The components in the IBM PC Network cable system are designed to easily connect up to 72 nodes. The Translator Unit can only be used on a passive data type of network. If the Short, Medium, or Long Distance Kits do not meet your network requirements, then the information here will assist you in designing your own network. If your requirements are for other simultaneous network services, then you must use another commercially available frequency translator with the proper filters.

The basic principles used in network design are described. Following the procedures described here will help you to configure a network using either the Short, Medium, or Long Distance Kits. If you need to expand beyond the capabilities of the kits, the same principles apply to larger networks. You can design larger networks that will work properly if the proper signal level is delivered from the Translator Unit to each node, and to the Translator Unit from each node.

When the necessary signal level cannot be delivered to a node, you can add amplifiers to extend the range of the network but not with the IBM Translator Unit. However, the design of such networks is beyond the scope of this book. Professional broadband or CATV network design engineers should be consulted to ensure the successful design of such networks. It is recommended that any network extending beyond the range possible with the cable kit be designed, or at least the plans reviewed, by an experienced network/CATV design professional. Such a consultation can help to identify possible problems with installation, design and use, both now and in the future.

Network design involves many steps. The design task using the IBM PC Network is much simpler than if you were starting from scratch. The medium, topology, access method, and frequencies are all defined. If you choose to design your own network, you need to consider the following approach to the design:

- Reviewing your needs
- Surveying the physical layout
- Designing the network
- Installation of the network
- Certification of the layout

Reviewing Your Needs

The first step in designing your IBM PC Network is to decide how the network will be used, who will use it, and where they will use it.

The IBM PC Network is used to transfer data among different Personal Computers within a local area. In addition, with the proper frequency translator, a network like this can also be used to carry voice and video signals. Keep these additional applications in mind when deciding where to place the outlets and run the cable.

If you have an existing base of Personal Computers and you want to connect these to the network, part of your survey is already done. You know who will use the network and where some of the outlets need to be. If you do not have any Personal Computers currently installed, you need to determine where they will be located. Estimate where additional nodes for future expansion should be placed, and plan your layout accordingly.

Surveying the Physical Layout

The following procedures can guide you in the surveying task.

1. Obtain a scaled drawing of the entire facility. Architect's blueprints are best, but if you're only interested in wiring a suite of offices, a small sketch that you draw yourself will suffice.

2. Mark the locations that you know will be network outlets. These locations could be areas that already contain Personal Computers or will have them in the future. Mark these locations on the drawing as close as possible to where the actual connection outlet will be required.

3. Add to this drawing one cable outlet for every desk, or one that could be shared by a cluster of desks. Planning for this kind of expansion might seem excessive now, but it could save extensive rewiring later.

4. Note hallways and other areas suitable for routing the main cable from the Translator Unit to the outlets. Mark possible paths on the drawing for selection of the routes.

5. Decide on a location for the frequency translator. In a larger network, place the frequency translator in a central location to keep cable branches short. Also, mark the location of the power outlet for the translator.

Expanding Beyond the Cable Kit

When your network requires more taps or a longer cable run than is possible to achieve with the cable kit, you must verify that the signal level delivered to each node on the network, including the frequency translator, is correct. If you cannot verify this yourself, you can obtain help from broadband/CATV design engineers or from cable installation consultants that have the capability.

Physical Layout

The purpose of the building survey is to plan distances for the network. Check the materials and construction of the facility's walls, ceilings, and floors wherever you install the cable. From this survey, a more detailed plan for where to place and how to secure the cable should evolve. Many options are possible, above a false ceiling, in a conduit raceway, or cable clamped to the wall. If this is a bigger job than you can or want to handle, call in a contractor. If you don't know exactly what is required by local building codes, some advice from a professional consultant is well worth the cost.

The drawings for the building can be helpful to a contractor because they reveal materials used, they show what is behind walls, ceilings, and floors, and they help identify cable pathways. There might be existing cable conduit or trays for installing such wiring. The drawings should show these in detail.

When installation starts, make any changes to the original drawing as they occur. This is very helpful when you are using the drawing for problem determination.

The cost of the installation depends on many factors. A contractor experienced in laying cable knows the right questions to ask, and might suggest some alternative approaches. The complexity of the installation is also affected by your requirements, such as:

- Local Fire and Electrical codes
- How the cable will be routed
- Whether outlet boxes and plates will be used

Component Description

The following describes some of the components used in broadband networks. Some general specifications are also described. When designing a network, you need to identify these specifications in order to select the correct components.

Passive components are used to distribute the signal power to the necessary outlets. Each component has its own function as follows:

Splitters

Splitters divide or combine power. The power division causes an insertion loss of approximately 10 log n(dB) where "n" equals the number of power splits. The splitter has internal losses caused by impedance mismatches and resistive losses. Isolation prevents any power passing between the lines that have been split. For a two-way splitter, signals and power are symmetrically divided into two separate lines. When a two-way splitter is used, the line that was split will have a 3 dB level reduction and a 0.5 dB internal loss. The total insertion loss is approximately 3.5 dB.

Directional Taps

The directional tap removes a small amount of power from the line input, causing an insertion loss to the line output. A directional tap is a 3 connector device consisting of a line input, line output, and a tap off port. The directional tap removes a small amount of power from the line input and directs it to the tap-off port. The difference in amplitudes between the line input and the tap off port is referred to as the tap attenuation value. Efficient removal of required signal levels leaves the majority of line power intact, capable of suppling many more taps.

The insertion loss in the tap occurs between the line input and the line output. The tap attenuation occurs between the line input and the directional tap port. The isolation occurs between the directional tap port and the line output port.

Tilt Compensators

Coaxial cables have attenuation that varies with frequency. The higher the frequency, the higher the attenuation. This effect is known as tilt. Tilt compensators have attenuation that varies with frequency. The higher the frequency, the lower the attenuation.

These devices equalize cable tilt so that the attenuation at both the high and low frequencies are the same. The tilt compensator has the inverse tilt relationship as the cable it is equalizing. Tilt compensators have symmetrical insertion loss and can be used in either direction.

Tilt compensators have different tilt specifications over different frequency bandwidths. Selecting a compensator that has a different bandwidth specification can provide the value of fixed equalization that is desired if the value of cable tilt that you are looking for is not a standard value.

Terminator

This device is used to prevent reflections of power back into the cable system.

Attenuators

Attenuators provide a constant attenuation over a wide range of frequencies. This attenuation is symmetrical from either end. It does not matter in which direction the attenuator is connected.

Computing Signal Levels and Network Attenuation

This section discusses signal levels in an IBM PC Network. The design of the network includes providing a signal path between the frequency translator and user devices, to ensure adequate signal strength at each node. Each layout design drawn on paper can be checked for providing proper signal amplitude by taking the transmitter output level and subtracting the required receiver signal level. The design must achieve nominal attenuation values that satisfy the level difference requirements between the transmitter and the receiver.

Signal Level Margins

This section discusses signal level margins allowed at each node. The cause of a signal loss in the network and how to account for it is also described.

The specified values are as follows:

Adapter input range:	-7 to 24 dBmV—operating range 8.5 dBmV—nominal 61.25 dBmV—maximum input without damage.
Adapter output:	56 dBmV $\pm$4 dB 56 dBmV—nominal
Frequency Translator input range:	7.25 dBmV to 21.25 dBmV 14.25 dBmV—nominal 60 dBmV—maximum input without damage.
Frequency Translator	

gain:	36 dB ±4 dB 36 dB —nominal
Network Attenuation:	41.75 ±3 dB @ 50.75 MHz 41.75 ±4.5 dB @ 219 MHz

Network Attenuation

Passive loss is the attenuation caused by all the passive components in the network. This loss is constant across the entire frequency spectrum on the network. *Cable loss* is the attenuation caused by the coaxial cable. Cable loss increases with frequency and is called *cable tilt*. The IBM PC Network Medium and Long Distance Kits have built-in tilt compensation. Different tilt compensation is required for different types and lengths of coaxial cable.

The cable attenuation for the IBM PC Network in the forward path (219 MHz) is 1.95 dB per 100 feet and in the reverse path (50.75 MHz) is 0.95 dB per 100 feet.

A 100 foot length of RG-11 U cable has 1.95 dB of attenuation at 219 MHz, and 0.95 dB of attenuation at 50.75 MHz. If you connected an equalizer that had 0.95 dB of attenuation at 219 MHz, and 1.95 dB at 50.75 MHz to the length of coax, the resultant equalized flat loss would be 2.9 dB from 50.75 MHz through 219 MHz. Flat loss in an equalized cable is equivalent to passive loss.

Network balancing consists of adjusting the passive loss in the path to each node to obtain signals within the acceptance range at the PC Network Adapter receiver and at the frequency translator receiver. The loss in a path can be adjusted by changing component values, cable length, or the distribution structure.

- Splitters divide power for symmetrical separation.

- Equalized cable loss transports the signal with a minimum of tilt.

- Attenuators decrease signal level.

- Directional taps remove a small amount of signal power from a signal line and produce a tap signal level lower than the supplying line. Taps are rated by an attenuation value.

- Fine tuning of level (changing level by less than 3 dB) can be accomplished by either changing the length of cable or by using precision attenuators.

Design Procedure

This section provides a checklist of steps to follow when designing a network.

1. Survey the facility to determine the required installation effort, and to identify cable pathways and outlet locations.

2. Determine the physical layout of the cable system.

3. Mark the cable path on the drawing from the frequency translator through the facility to each outlet.

4. Compute or estimate the length of each cable segment and calculate the cable loss of each segment.

5. Calculate the signal level delivered to each outlet from the frequency translator. Calculate the signal level delivered to the frequency translator from the farthest node on each branch.

After completing this design procedure, a design consultant can check over your facility and your design.

A design review before installation by a qualified broadband/CATV engineer can identify any possible problems you might face during installation and use.

Choosing a Topology

The first step in any design of a network is to define the cable routing topology. If all of the nodes are located in clusters, a star topology can be the most appropriate. If the nodes are adjacent to a long run, similar to offices on each side of a long hallway, a bus topology is most appropriate. If uniform outlet distance for a large area is required, a tree or multiple bus topology should be used.

Star Topology

The star topology is the simplest network to design and install because the attenuation path is a single line between the Translator Unit and the node. This topology is made up of mostly cable series attenuation and has a straight-forward control on errors.

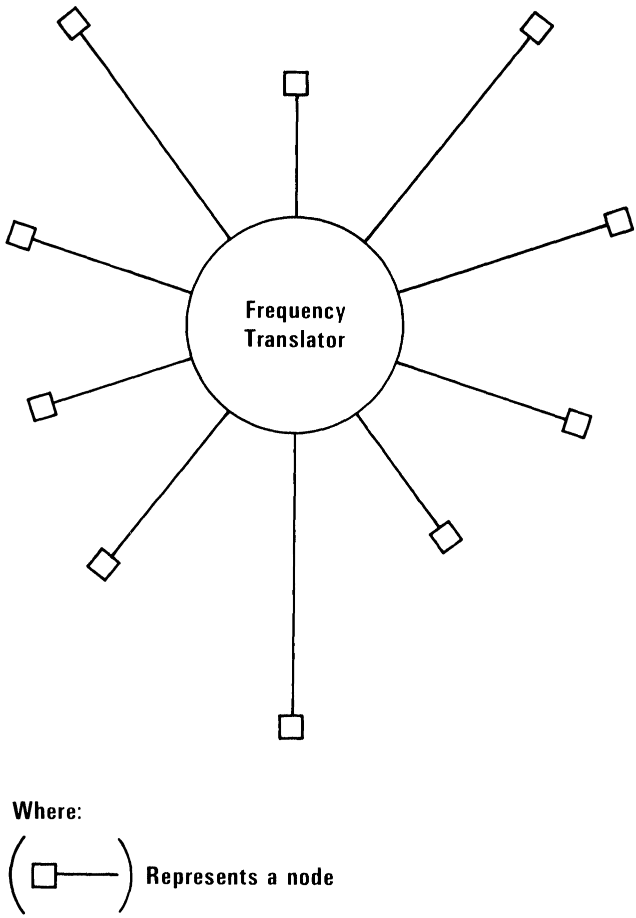


Figure 4-1 Sample Star Layout

The component tolerances add to provide one simple uncertainty. This aspect of the topology simplifies the transparent network design requirement. The star topology's main trade off is that it requires large amounts of cable to implement a large-scale network. For a small-scale network, this can be the most cost-effective solution.

```
Example:

        Problem--Design a network that supports
                 two nodes that are two hundred
                 feet apart.

        Solution--Locate the Translator Unit in
                  the center of the two nodes
                  and route a hundred feet of
                  RG-11/U coaxial cable to each
                  node.  A two-way splitter
                  provides the outputs with a
                  minimum of error.  Place an
                  attenuator on each end of the
                  splitter and connect the
                  cable to the attenuators.
```

The design example is the simplest solution to the problem. The design does not allow for expansion. This network would require redesign when you decide to connect additional nodes. If the two-way splitter has an insertion loss of 3.5 dB to each output, and the coaxial cable has an attenuation of 0.95 dB per 100 feet at 50.75MHz (return path) and an attenuation of 1.95 dB per 100 feet at 219MHz (forward path), the attenuation value can be calculated. The IBM PC Network devices require a network attenuation of 41.75 $\pm$3 dB in the return path and an attenuation of 41.75 $\pm$4.5 dB in the forward path.

If the attenuation of the splitter, cable and the attenuators are within the network specifications, the design is acceptable. A quick check of the component errors reveals that the splitter has a $\pm$0.25 dB tolerance and the tilt of the coaxial cable is 1.0 dB. The total possible component error is less than the required network tolerance. The design will work if the nominal attenuation of the design example does not introduce additional error.

Solving for the attenuator value, we find that the ideal attenuator has 37.30 dB in the return path and 36.30 dB in the forward path. Since attenuators are passive devices the least error would result if the average value of 36.80 dB was used. This would present a problem to the designer since the value of 36.80 is not a standard value. To make sure that the selected standard value attenuator works, perform an error analysis.

As the standard values of attenuators are 3, 6, 10, and 20 dB, cascading certain values produce a 36 ± 1 dB attenuator. If this value is added to the attenuation of the splitter and the cable, the design meets the path attenuation by 0.45 dB. You can verify this by adding the splitter, cable and attenuators, attenuation, and tolerances together and comparing the result to the network specification for return path loss.

Bus Topology

When the outlet design suggests a bus topology, the material cost may be lower than the star topology. The bus topology requires only one main feeder cable that distributes the proper signal levels to the outlets by providing the proper attenuation.

The bus topology has a slightly different design approach than a star approach. The attenuation path to an outlet of a bus network can include the insertion loss of several devices in cascade. The insertion loss of the passive device may have a small dependency on frequency. When the passive device is placed in cascade, the dependency may accumulate into a large path attenuation error. It is very important to select parts that have insertion loss information at several frequencies. This information reduces the design error in the network.

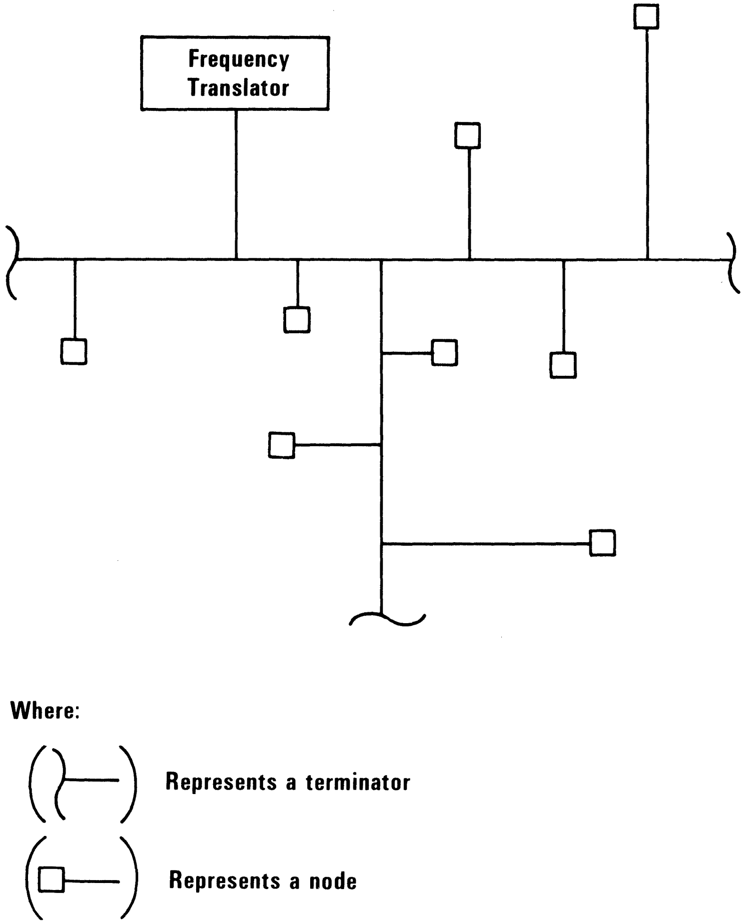


Figure 4-2 Sample Bus Layout

Passive components have tolerance specifications for insertion loss, frequency response, and tap attenuation. The insertion tolerance can accumulate when placed in cascade. When designing a network, it is important to control these tolerances. If the network is to be implemented from paper to practice and the network is not going to be tested for path attenuation at all nodes, limit the number of cascaded insertions. If the design is tested in both paths, it can have cumulative insertion tolerances that exceed the allowable path attenuation error. The network can require field selection of components and slight reconfiguration in the field to bring the network within specifications.

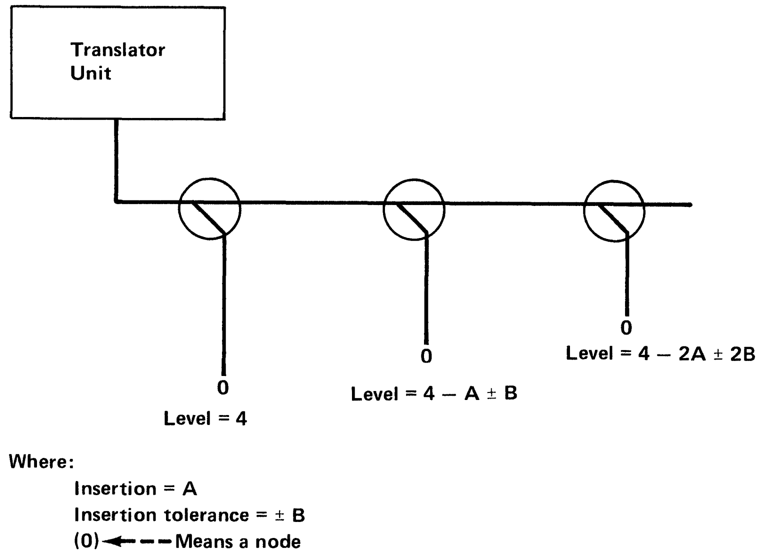


Figure 4-3 Cumulative Insertion Error on a Bus Structure

> **Note:** Neglect the cable and assume that all taps have the same nominal tap attenuation value.

In the figure, all of the taps have the same insertion attenuation of "A", with an insertion tolerance of "B". The taps following the first tap have a higher attenuation value, due to the cascaded insertions. Also note that the figure neglects the cable attenuation. If the insertion loss "A" is made smaller, the difference between the taps will be smaller. In the cascade shown, the center tap has the average attenuation difference between the first and the last taps. This assumption is good for tap-to-cable insertion attenuation ratios as small as 0.5 : 1. If the average tap is designed to have the nominal path attenuation, then a symmetrical error occurs in the first and the last taps.

The previous discussion made the assumption that the insertion attenuation "A" was made as small as possible. High tap isolation values have the smallest insertion losses. Smaller insertion losses introduce less cascade error to the next tap of the same isolation value. High tap isolation values can only be used when large signal levels are fed into the bus. The trade-off of this error-control method is poor power utilization. Many more nodes can be accommodated when the tap isolation range is between 20–30 dB, without incurring the large insertion errors of the low isolation directional couplers.

The transition between tap attenuation values can also be studied to minimize cascaded error. At the last value of tap insertion "A", the attenuation from the nominal tap is two insertions and two cable length insertions from the input of the next cascade tap attenuation value. The advantage of the insertion loss being incurred after a tap can help the next 3 dB lower tap attenuation value to be closely centered about the nominal path attenuation. If the second group of three taps has the same insertion attenuation as the first group of three taps, the nominal value would fall in the middle. If the cable attenuation and the tap insertion added up to 1 dB, this method reduces errors. Use this method when it is not required to have taps located in exact locations. Therefore, if the taps can be made location independent, the error accumulated in a bus structure can be reduced below the tap selection increments of 3 dB. This method, for example, recovers 1 dB of error that would have been used in permitting taps to be located in any position along the bus.

If the cable length between the taps is held relatively constant for a given tap insertion value, the bus can be modeled as a replicating unit that has one cable insertion and one tap insertion. This concept allows the replicating units to be cascaded like lamp extension cords. The first cascade of three values uses one tap attenuation value, while the next cascade of three taps uses the next three dB lower tap attenuation value. If the insertion loss of the tap increases, the tap spacing can be decreased. If the bus requires tilt compensation, the structure can be repeated.

If two groups of three cascaded taps are implemented, the nominal levels appear on the second tap of each group of three taps. When the cable attenuation non-linearities are considered, it may be better to design the nominal of the first group of three to have a higher return attenuation, and the nominal of the second group of three to have a lower forward attenuation. If these nominal errors are symmetrical around the nominal network attenuation, the lowest error approximation can be achieved. Verification of these concepts can be time consuming.

Another method of reducing the insertion attenuation effect on cascaded output levels is to increase the tap from a one-way to a two-, four-, or eight-way. This change does not increase cascaded insertion error. These devices use a single insertion tap connected to a power splitter, providing more outputs with fewer insertions.

Drop-line cable errors from any feeder cable must be taken into account when designing the bus distribution. It is best to make all drop cables the same length. Minimum drop cable tolerances allow an increase in the amount of insertion tolerance permitted.

If feeder cable tilt error becomes too large to accommodate, tilt compensation is required. Periodic spacing results in the least amount of nominal attenuation error to the outlets. Normally an uncompensated bus has too little attenuation in the return path and too much attenuation in the forward path . When tilt compensators are used in a distribution bus, it is possible to over compensate the bus. This requires that the return path be checked for over attenuation and the forward path be checked for under attenuation. This is in addition to the two conditions mentioned for uncompensated bus topologies.

If tilt compensation is required in a cable layout, it is easier to compensate a branch than it is a feeder line. Branches are used to distribute signal power to the feeders. This is analogous to a highway, providing access to individual neighborhoods. The highway is the branch, the neighborhood street is the feeder, the driveway is the drop lines, and the garage is the node (outlet) of the network.

A tree topology allows the functional separation of responsibilities. The branch divides the signal power among the feeder cables, and the feeder cables distribute that power to the outlets. An example of a tree topology is a two-way splitter that branches into two bus feeder lines. The branch can resemble a small feeder bus, except that its outputs branch into other bus topology feeder lines. The branch is a logical extension of the bus design discussed previously.

Extended Coverage

The limiting factors in extending the range of the network are the cable attenuation in the forward path, equalization limitations of the fixed equalizers, and tolerances of the component values. The difference in cable attenuation does not cause a problem when using the IBM kits, because the longest cable run possible is 1000 feet from translator to adapter.

Rigid aluminum low-loss cable or amplifiers would have to be used. Both of these alternatives are beyond the scope of this book.

When designing cable networks, include expansion into the design. The simplest way to expand the network is to design unused splitters or taps into the branches.

Future Needs

This section contains some notes on planning for future expansion of your network, including extending the coverage to add new nodes and connection to other networks.

Adding Outlets to the System

The easiest way to expand an existing network is by connecting new Personal Computers to existing nodes that were designed into the network and left idle. It is best to design your system with expansion in mind. One of the most common techniques used to design for expansion is to leave idle taps on every multi-tap in the system. When four taps on every eight-port tap are left idle, the network has a 100% expansion capability.

In some areas covered by the network, you should leave room for more expansion if multiple services are to be added in the future. Incorporating video or voice into a network can lead to a need for two or three taps in each office requiring such services. For most small office applications, planning for one tap for each desk in the facility plus some extra taps, should satisfy most requirements.

Adding Branches to the System

For networks that require major expansion, outlets might not provide enough connectivity. For such networks, splitters can be inserted into the signal path where a major branch might be required in the future. One leg of the splitter could be terminated for that future need. The other leg of the splitter feeds the existing network. Computing signal levels properly for this portion of the network allows you to add entire branches with many branches and nodes of their own. Then a new portion can be designed, installed, tested separately, and attached to the main network at the reserved expansion leg of the splitter. This is the type of implementation used in the IBM PC Network.

Test Equipment

This section lists the basic items of test equipment that can be used to check out the passive network functions. The two items covered here are the RF sweep generator and the RF voltmeter.

RF Sweep Generator

This type of generator produces signals of the proper frequency and amplitude for transport over the network. Transmitting a fixed-frequency and fixed-amplitude carrier signal over the cable allows you to check nominal attenuation levels in the forward and reverse paths. When the sweep generator is used in the sweep mode, it can locate either compressed cables or loose connections that are causing frequency response irregularities.

The generator's frequency range may be set to either the forward or reverse bandpass, depending on its location either at the translator or at the outlets. Its amplitude is also set depending on whether it feeds the network, or whether it feeds the outlets.

Set the sweep generator to sweep the forward frequencies at the translator unit location. At this location, the generator supplies signals that can be used to test each outlet on the network. The generator can normally be set to supply the same level as the translator, so that the level at the outlets are the same as the network device. The generator can also be connected to any outlet in the network. In this case it should be set to the reverse frequency band. It should transmit at the same level as the devices connected to the network.

RF Sweep Receiver

The device that monitors the sweep generator output is the sweep receiver. This device is normally controlled automatically (range, frequency) by the sweep generator. Investigate any discontinuities or excessive losses at any frequencies in the band to ensure that they are not causing degraded network performance.

This type of receiver allows reception of sweep generator signals and analysis of network path attenuation for a predetermined range of frequencies.

RF Voltmeter

The *RF voltmeter* or *field strength meter (FSM)* is a popular test instrument. It measures the amplitude of an RF signal at a specific frequency. This device is easy to operate and can be used for measuring signal levels, verifying signals, and troubleshooting. The output meter is calibrated in dBmV. Connection to the network is made through an RF cable directly to an outlet or tap port. The frequency is selected by setting the appropriate tuning dials or by entering numbers on a keypad, and the signal level is read on an analog or digital meter. Some RF voltmeters also have built-in dc voltmeters.

The following list covers typical specifications for an RF voltmeter.

Amplitude	-40 dBmV to +60 dBmV
Frequency	4 to 460 MHz
Temperature	0 to 120 degrees F
Accuracy:	
Frequency	±100 kHz to ±1 MHz
Amplitude	±0.5 dB at 68 degrees F, ±1 dB over full frequency

	range
DC	Voltage ±10%.
IF Bandwidth	280 kHz to 600 kHz
Power	Battery
Calibration	Built-in

More Specialized Equipment

An *RF radiation monitor* measures RF energy radiated by equipment or components of the network. This instrument pinpoints areas where the system radiates RF energy, most often through a poor connection, a corroded connector, or a damaged cable. When radiation occurs, you can assume that the system is more susceptible to signal ingress. Signal ingress could hinder the proper operation of the network.

The *cable reflectometer* is used in locating cable faults caused by physical breaks, bends, or kinks in a given span of cable. This instrument can indicate the location of the fault to within a few feet. Cable system troubleshooting time is minimized by the combined use of a reflectometer and accurate, scaled drawings of the cable layout.

Checking a Network

The following uses the IBM PC Network as an example on how to check a network. RF signal levels can be checked at each outlet to ensure that the distribution system is working as desired. This checkout can be done with a sweep generator and a sweep receiver.

Where to Check

If a known working Personal Computer with adapter does not work when it is connected to a suspected outlet, check the outlet. Also check the outlet if the Personal Computer obviously works only marginally (that is, much lower throughput or slower response from the network when sending data to other nodes, compared to the performance when it is connected elsewhere.)

When to Check

- When first installing the network, check signal levels at all or selected outlets. Ensure that every branch can carry RF signals.
- Check the network whenever a problem is suspected, such as when a noticeable performance degradation occurs, and the computer checks out OK otherwise.

How to Check

Method 1: Forward Path Test

This puts the generator at the translator unit location and can check that each adapter receives the proper signal level.

1. Remove the network's RF cable from the translator and connect a 9 dB directional coupler (RMS CA-1090-M or equivalent) as shown in Figure 4-4 on page 4-26.
2. Set the generator to sweep frequencies from 150 to 300 MHz. Set the generator for an amplitude of 60 dBmV.

3. If required, set the notch filters on the sweep transmitter to sweep around channels T-14 and J. This will ensure non-interference with other network channels.

4. Connect the sweep receiver to outlets and monitor the sweep. The network attenuation is equal to the signal generator level minus the sweep receiver level. Forward network attenuation specification is 41.75 ±4.5 dB. Remember to test for this forward attenuation on a flat portion of the spectrum near channel J (219 ± 3 MHz). Also, remember to add additional network attenuation of 9 dB from the test tap.

   Note: The network attenuation used in this example corresponds to the attenuation used in the IBM Translator Unit and the cable kits

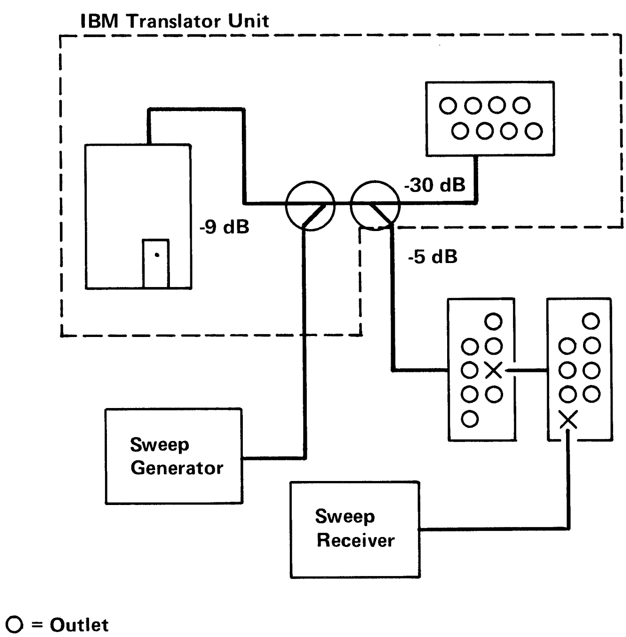


Figure 4-4 Forward Path Attenuation Test

Method 2: Return Path Test

This puts the generator at an outlet in the network, and tests the IBM Translator Unit on both the forward and reverse paths of the cable system. You can check that each adapter receives the proper signal level.

1. Connect the sweep receiver at the -9 dB sweep coupler.

2. Remember to notch out channels T-14 or a degradation in performance can result for the duration of the test.

   Note: Performing this test on an active network may have adverse affects on performance and could cause system errors in some Personal Computer's. Consider performing this test when the network usage is low.

3. Program the return parameters into the sweep generator. Set the sweep to cover a 5 to 116 MHz frequency range at a 60 dBmV level. The return attenuation should be measured near a flat portion of the frequency spectrum near channel T-14. The network attenuation is 41.75 dB ± 3 dB for the return path. Remember to take the additional 9 dB of attenuation into account when measuring network attenuation.

4. When the test is complete, remove the -9 dB coupler.

   Note: The network attenuation used in this example corresponds to the attenuation used in the IBM Translator Unit and the cable kits

Method 3: Walk-through Test

This procedure is useful for isolating a trouble spot.

1. Set up the sweep generator for a continuous wave (C W) single frequency at the Translator Unit location. Couple the generator through the 9 dB coupler as described in the forward path sweep test. The frequency should be in the forward path near channel J. An example frequency would be 225 MHz at an amplitude of 60 dBmV.

2. Set the RF voltmeter to its highest amplitude scale and set the frequency to the same frequency setting as the C W frequency set on the sweep generator.

   Note: When the sweep transmitter is in C W mode, only one frequency is transmitted over the network. A presence of this frequency can quickly determine an open, short, or a poor connection.

3. Take the RF voltmeter and a copy of your system drawing to the splitter in question, and check which nodes are working. This is accomplished by lowering the scale on the meter until a reading can be obtained. If you find a node without a signal, check at least one or two surrounding nodes.

4. Test the signal into the splitter.

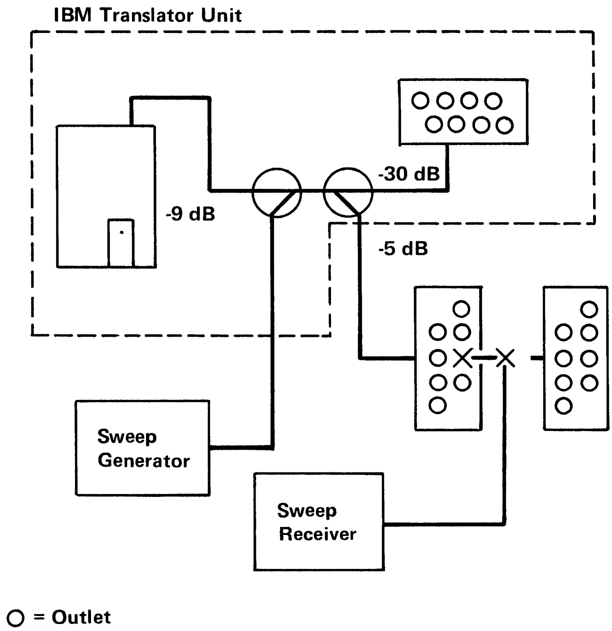


Figure 4-5 Splitter Test

5. If no signal is present, test for a signal into the cable going to the splitter. If a signal is present, the last part tested is not working properly.

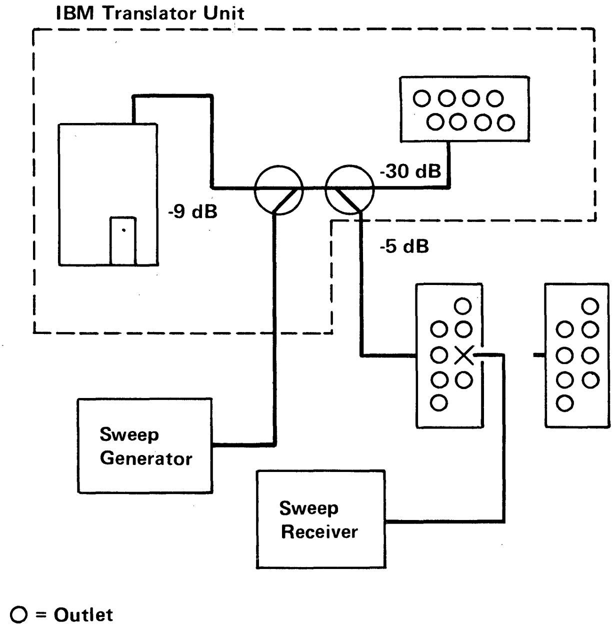


Figure 4-6 Cable Test

6. If no signal is present, test for a signal into the cable going to the Base Expander.

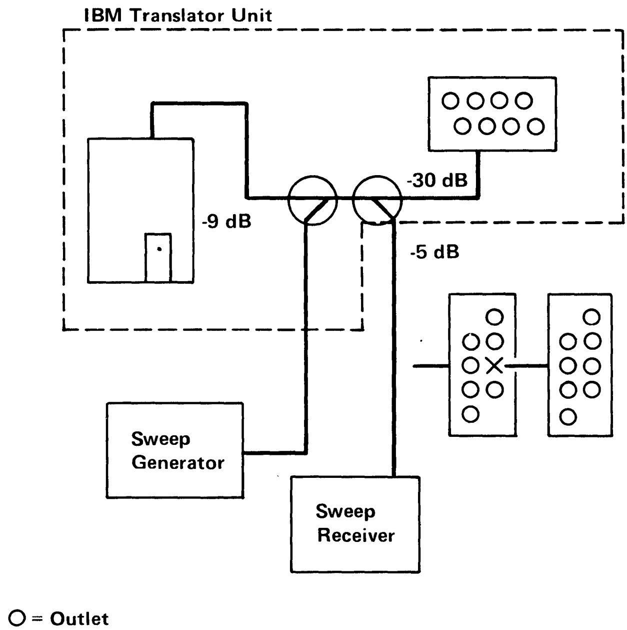


Figure 4-7 Base Expander Test

7. If no signal is present, test for a signal from the directional coupler.

What to Look for when Testing

Look for widely-varying signal levels between nearby outlets, signal levels outside the specified range, and extreme variations in signal levels from expected values (a map of the system with expected levels recorded when the system was working properly is desirable).

Also, check the following for any other problems:

- Cut cables
- Loose or corroded connectors
- Shorted cables
- Unterminated trunk lines

What Can Be Changed if a Problem Is Found

Change cables after isolating the problem.

Change connectors or passive components.

Change the adapter (most RF problems with the adapter would be found by the self-test).

Change the translator if it is not working.

Notes:

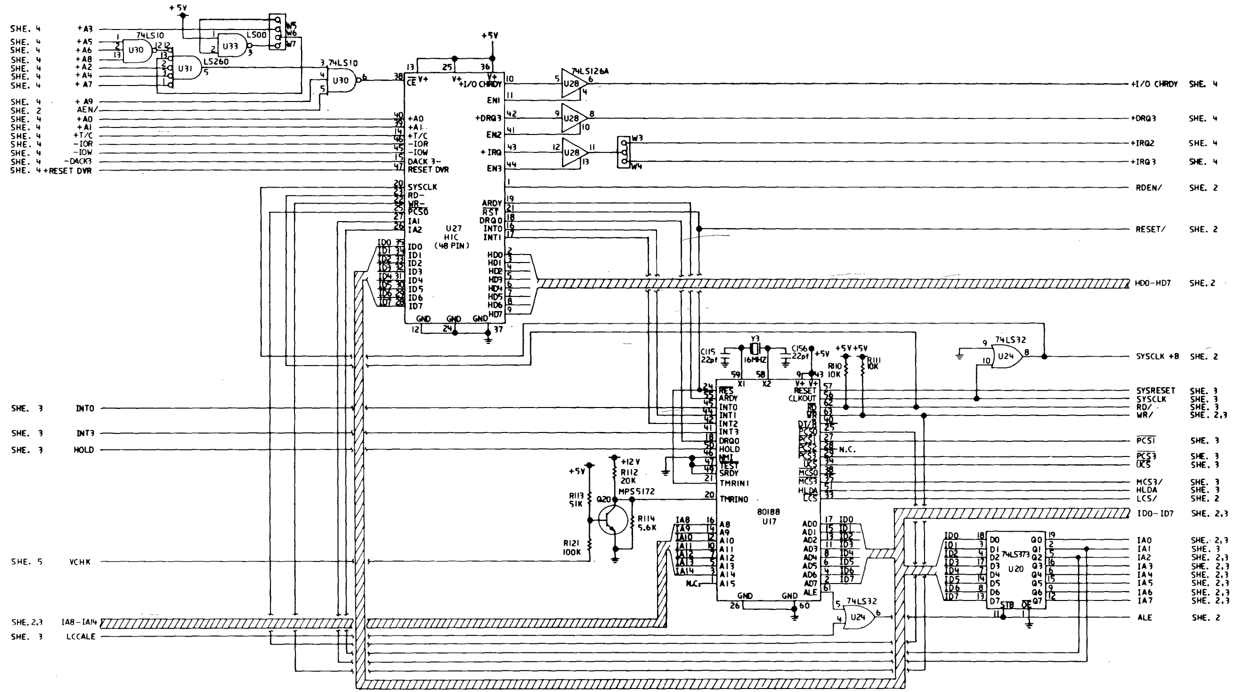
Notes:

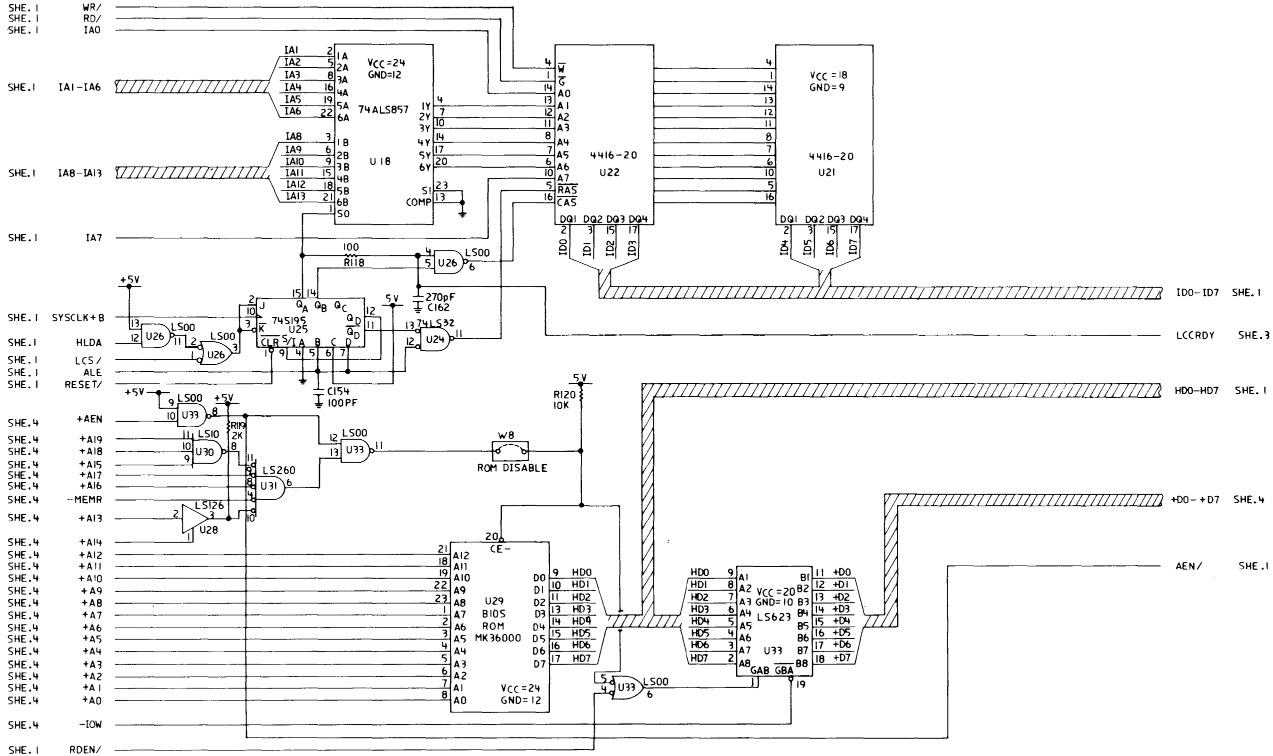
Appendixes

Contents

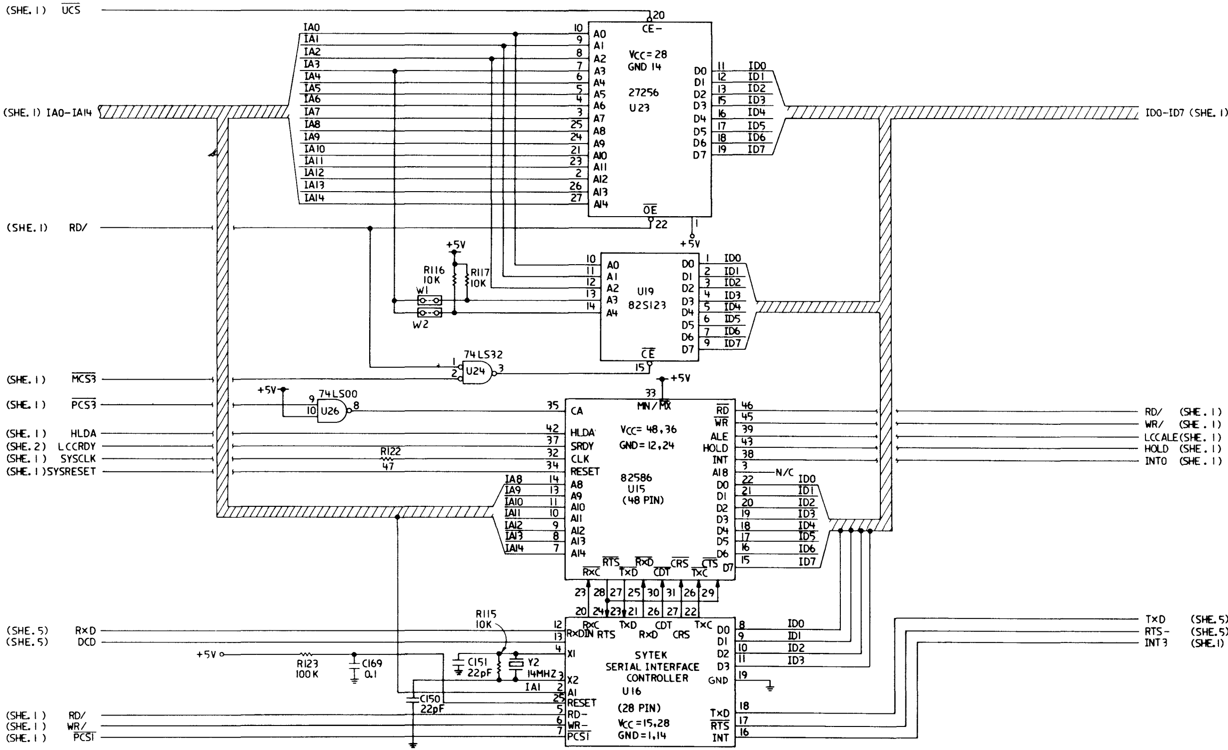
Appendix A. IBM PC Network Schematics

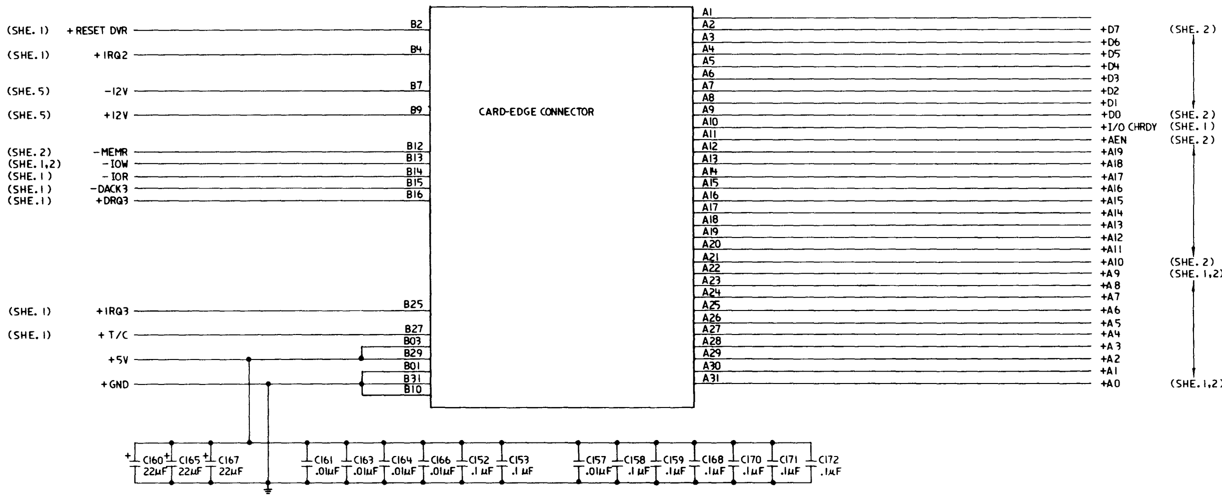
IBM PC Network Adapter Schematics

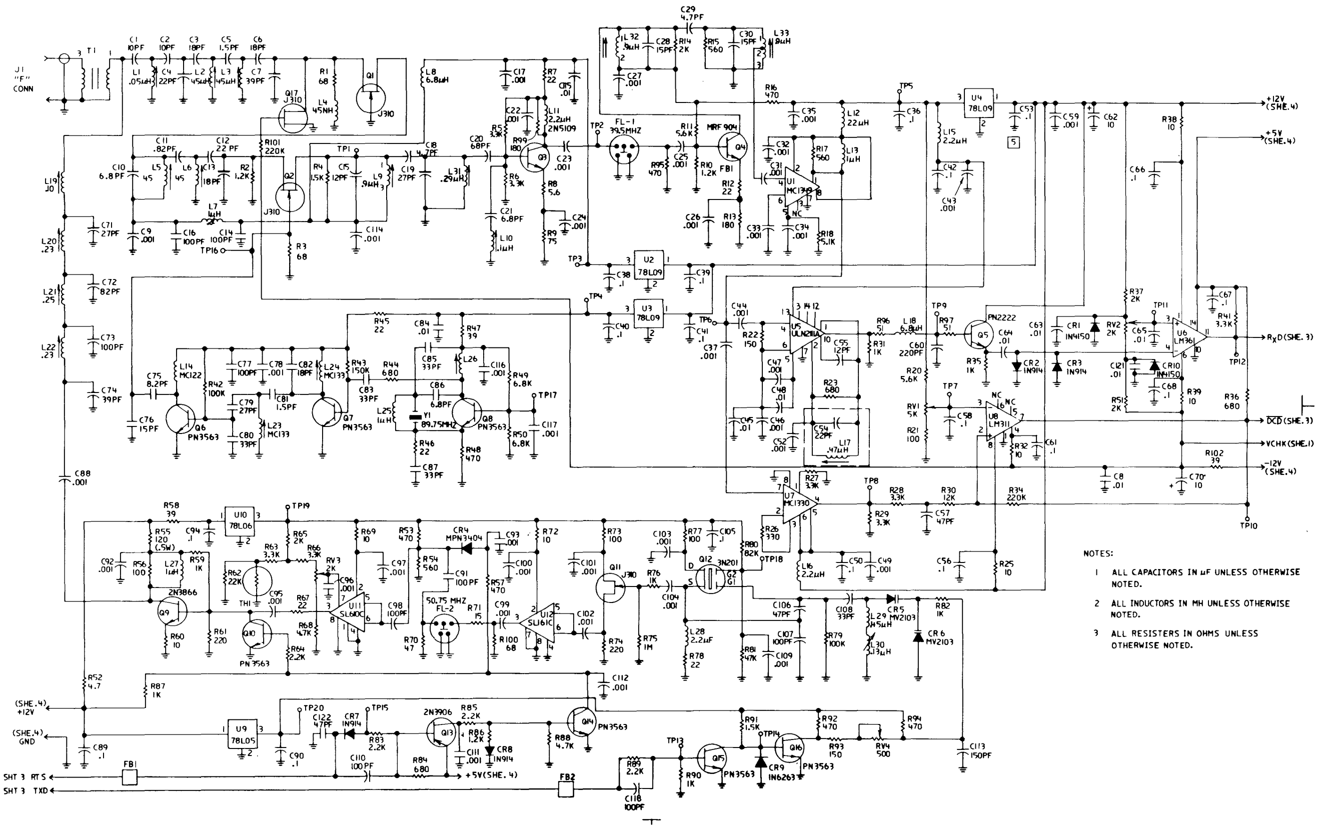

SHE. 4 +A3
SHE. 4 +A5
SHE. 4 +A6
SHE. 4 +A8
SHE. 4 +A2
SHE. 4 +A4
SHE. 4 +A7
SHE. 4 +A9
SHE. 2 AEN/
SHE. 4 +A0
SHE. 4 +A1
SHE. 4 +T/C
SHE. 4 -IOR
SHE. 4 -IOW
SHE. 4 -DACK3
SHE. 4 +RESET DVR
74LS10
U30
U31
LS260
LS00
U33
W5
W6
W7
+5V
U27
HIC
(48 PIN)
74LS126A
U28
W3
W4
+I/O CHRDY SHE. 4
+DRQ3 SHE. 4
+IRQ2 SHE. 4
+IRQ3 SHE. 4
RDEN/ SHE. 2
RESET/ SHE. 2
HD0-HD7 SHE. 2
74LS32
U24
SYSCLK +8 SHE. 2
SYSRESET SHE. 3
SYSCLK SHE. 3
RD/ SHE. 3
WR/ SHE. 2,3
PCS1 SHE. 3
PCS3 SHE. 3
UCS SHE. 3
MCS3/ SHE. 3
HLDA SHE. 3
LCS/ SHE. 2
ID0-ID7 SHE. 2,3
SHE. 3 INT0
SHE. 3 INT3
SHE. 3 HOLD
SHE. 5 VCHK
SHE. 2,3 IA8-IA14
SHE. 3 LCCALE
Y3
16MHZ
C115 22pf
C156 22pf
R110 10K
R111 10K
R112 20K
R113 51K
R114 5.6K
R121 100K
Q20
MPS5172
+12V
80188
U17
N.C.
74LS373
U20
IA0 SHE. 2,3
IA1 SHE. 3
IA2 SHE. 2,3
IA3 SHE. 2,3
IA4 SHE. 2,3
IA5 SHE. 2,3
IA6 SHE. 2,3
IA7 SHE. 2,3
ALE SHE. 2



Schematics A-5


(SHE. 1) UCS
(SHE. 1) IA0–IA14
(SHE. 1) RD/
(SHE. 1) MCS3
(SHE. 1) PCS3
(SHE. 1) HLDA
(SHE. 2) LCCRDY
(SHE. 1) SYSCLK
(SHE. 1) SYSRESET
(SHE. 5) RxD
(SHE. 5) DCD
(SHE. 1) RD/
(SHE. 1) WR/
(SHE. 1) PCS1
ID0–ID7 (SHE. 1)
RD/ (SHE. 1)
WR/ (SHE. 1)
LCCALE (SHE. 1)
HOLD (SHE. 1)
INT0 (SHE. 1)
TxD (SHE. 5)
RTS- (SHE. 5)
INT3 (SHE. 1)
VCC = 28
GND 14
27256
U23
CE-
OE
+5V
U19
82S123
CE
R116 10K
R117 10K
W1
W2
74LS32
U24
74LS00
U26
CA
MN/MX
HLDA
SRDY
CLK
RESET
VCC = 48, 36
GND = 12, 24
82586
U15
(48 PIN)
R122 47
N/C
RD
WR
ALE
HOLD
INT
A18
RxC
RTS
TxD
RxD
CDT
CRS
TxC
CTS
R115 10K
+5V
R123 100K
C169 0.1
C151 22pF
Y2 14MHZ
C150 22pF
SYTEK
SERIAL INTERFACE
CONTROLLER
U16
(28 PIN)
VCC = 15, 28
GND = 1, 14
RxDIN
X1
X2
A1
RD-
WR-
PCS1
GND
TxD
RTS
INT


CARD-EDGE CONNECTOR
(SHE. 1) +RESET DVR B2
(SHE. 1) +IRQ2 B4
(SHE. 5) -12V B7
(SHE. 5) +12V B9
(SHE. 2) -MEMR B12
(SHE. 1,2) -IOW B13
(SHE. 1) -IOR B14
(SHE. 1) -DACK3 B15
(SHE. 1) +DRQ3 B16
(SHE. 1) +IRQ3 B25
(SHE. 1) +T/C B27
+5V B03 B29
+GND B01 B31 B10
A1
A2 +D7 (SHE. 2)
A3 +D6
A4 +D5
A5 +D4
A6 +D3
A7 +D2
A8 +D1
A9 +D0 (SHE. 2)
A10 +I/O CHRDY (SHE. 1)
A11 +AEN (SHE. 2)
A12 +A19
A13 +A18
A14 +A17
A15 +A16
A16 +A15
A17 +A14
A18 +A13
A19 +A12
A20 +A11
A21 +A10 (SHE. 2)
A22 +A9 (SHE. 1,2)
A23 +A8
A24 +A7
A25 +A6
A26 +A5
A27 +A4
A28 +A3
A29 +A2
A30 +A1
A31 +A0 (SHE. 1,2)
C160 22µF
C165 22µF
C167 22µF
C161 .01µF
C163 .01µF
C164 .01µF
C166 .01µF
C152 .1 µF
C153 .1 µF
C157 .01µF
C158 .1 µF
C159 .1µF
C168 .1µF
C170 .1µF
C171 .1µF
C172 .1µF



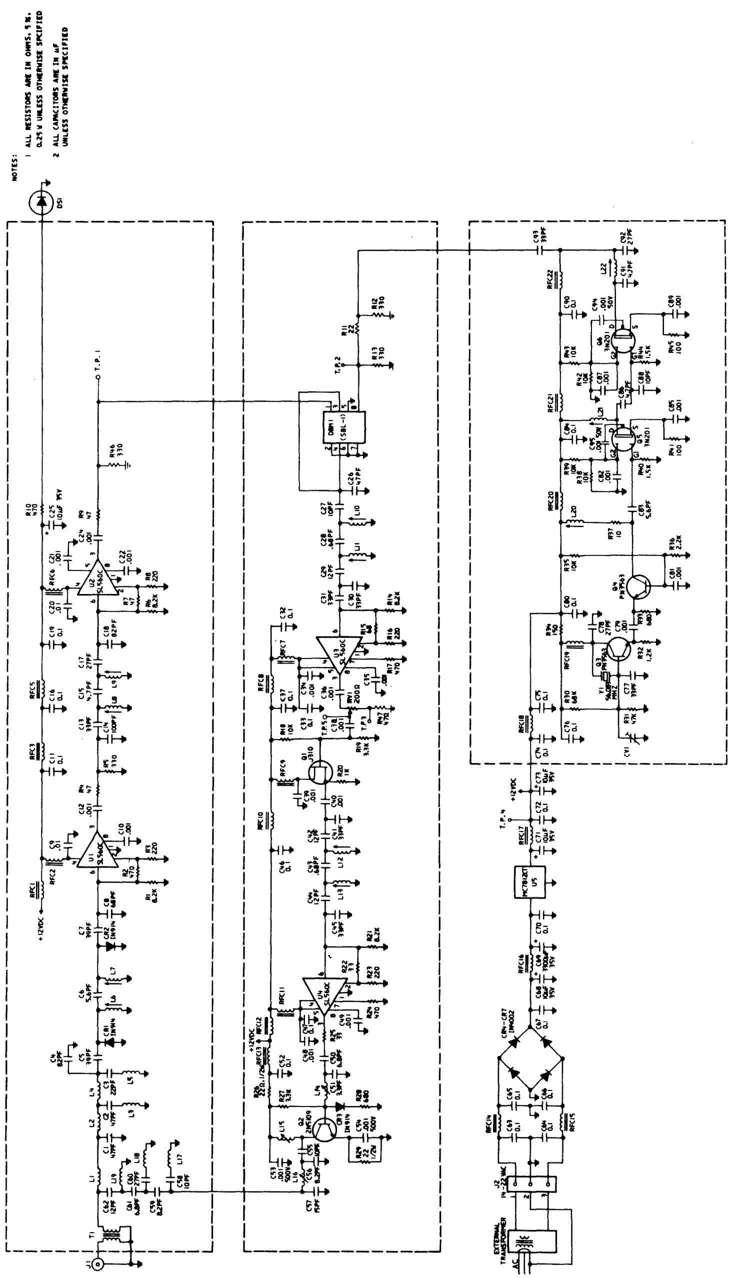
NOTES:

1 ALL CAPACITORS IN µF UNLESS OTHERWISE NOTED.

2 ALL INDUCTORS IN MH UNLESS OTHERWISE NOTED.

3 ALL RESISTERS IN OHMS UNLESS OTHERWISE NOTED.

IBM PC Network Translator Unit Schematics


NOTES:
1 ALL RESISTORS ARE IN OHMS, 5%, 0.25 W UNLESS OTHERWISE SPCIFIED
2 ALL CAPACITORS ARE IN µF UNLESS OTHERWISE SPECIFIED

Appendix B. IBM PC Network Specifications

This appendix is a list of specifications for the IBM PC Network hardware:

Environmental Specifications

The following lists the environmental specifications of the IBM PC Network adapter and the IBM Translator Unit.

Temperature The operating temperature range is from 10 to 35°C, (50 to 91°F) ambient. The storage temperature range is from -40 to 60°C, (-40 to 140°F).

Humidity The operating humidity range is from 8% to 80% non-condensing. The storage humidity range is from 5% to 100% non-condensing.

Altitude The operating altitude is from -305 to 2135 meters, (-1000 to 7,000 feet).

IBM PC Network Adapter Specifications

This section summarizes the basic specifications of the adapter.

Electrical Power Requirements

The specifications for the power requirements of the adapter from the three power supplies are as follows:

Voltage	Tolerance	Ripple	Total Current Used
+12.0V	±5%	100 mV pp	0.36 A
+5.0V	±5%	100 mV pp	1.40 A
-12.0V	±10%	100 mV pp	0.03 A

Adapter RF Modem Specifications

The following is a list of the specifications for the RF modem on the adapter. See "RF Modem Section" on page 3-50 for a complete list of specifications for the RF modem section.

• Input/output impedance	75 ohms typical
• Return loss:	
- With power ON (T-14, J)	≥ 14 dB
- With power OFF (T-14)	≥ 14 dB
- With power OFF (J)	≥ 8 dB
• Transmit channel	T-14 (50.75 MHz)
• Modulation technique	Frequency shift keying (FSK)
• Frequency shift	±2 MHz ±200 kHz centered about 50.75 MHz.
• Transmit output level (ON)	56 dBmV ±4 dB
• Transmit output level (OFF)	≤ -20 dBmV
• Receive channel	J (219.0 MHz)
• Receive bandwidth	3.6 MHz (nominal)
• Receiver Input level	8.5 dBmV
• Input level range	-7 dBmV to 24 dBmV
• Maximum input level	61.25 dBmV

• Carrier detect threshold	-20 to -8 dBmV
• Maximum number of adapters supported	1000
• Maximum distance supported	Up to 5 kilometers with the proper network design.

Network Cable Characteristics

The IBM PC Network hardware communicates on a 75-ohm coaxial cable network. This section defines the characteristics of that network.

• Input/output impedance	75 ohms typical
• Tap reflection coefficient (from 10 to 350 MHz)	$\geq$14 dB
• Transport echo delayed more than 25 ns.	-30 dB maximum
• Port ac-dc power characteristic	Blocked
• Transport channel (6 MHz) flatness (peak to valley)	2 dB
• Transport phase delay in channels T-14 and J, round trip:	$\pm$25 ns $\leq$100 μs
• Transport second order intermodulation distortion	-56 dBC max (Note 1)
• Transport third order distortion maximum (Note 1)	-56 dBC composite
• Forward carrier-to-noise ratio (Note 2)	53 dB minimum
• Reverse carrier-to-noise ratio	53 dB minimum

- Transport carrier-to-hum ratio 40 dB minimum

- Transport Transfer Characteristics (Includes cable and tap):

The output level variation over nominal system loss in the return path must be within ±3 dB of the input level. The forward path must be within ±4.5 dB of the input level. Both specifications apply to the corresponding system channel.

- Forward noise floor in channel J with all transmitters OFF at worst case mode: ≤ -24 dBmV (Note 2)

Note: 1) Referred to available forward tap level.

Note: 2) For a network having a 47.5 dB nominal network attenuation from the adapter transmitter to the adapter receiver and having 1000 adapters on the network.

IBM Translator Unit Specifications

This section defines the electrical specifications of the Translator Unit.

Electrical Specifications

• Input/output impedance	75 ohm typical
• Trunk feed levels (typical)	
- Forward	+50.25 dBmV
- Return	+14.25 dBmV
• Trunk feed return loss	14 dB minimum
• Reflection coefficient bandwidth	4 MHz minimum
• Channel Allocation	
- Input center frequency	50.75 MHz
- Output center frequency	219 MHz
- Channel bandwidth	6 MHz
• Input (receive) characteristics	
- Minimum input level	7.25 dBmV
- Maximum input level	21.25 dBmV
- Selectivity	
3dB bandwidth	≥6 MHz
30dB bandwidth	≤24 MHz
- The maximum sustained input level without damage	60 dBmV

- Output characteristics

- Maximum carrier output level	61.25 dBmV

- Translator oscillator frequency characteristics

- Long term stability (until the end of life over environmental range)	±80 ppm
- Center frequency accuracy	±11.7 kHz
- Spurious level at 6 MHz from carrier	-40 dBC

- Translation characteristics

- Nominal loop gain	36 dB
- Loop gain setting accuracy	±1dB
- Gain variation until end of life and over environmental range	±3 dB
- Pass-band bandwidth	±3 MHz
- In band level ripple (peak to valley)	1.5 dB
- Group delay	±25 ns max.
- Phase delay	200 ns

- Carrier to noise ratio (at typical carrier output level, with the specified signal source at the input)	47 dB min.
- Noise figure (for rated loop gain)	9 dB
• Maximum adapters supported by the Translator Unit:	256
- Pass-band spurious level (referred to maximum carrier output, with the specified signal source at the input)	-40 dBC max
- Carrier to hum ratio	40 dB min
• Power supply requirements	
- Line voltage variation	14.5–23.4 Vac RMS
- Power consumption max	11 watts

Physical and Mechanical Characteristics

• RF Connectors: (The F connector is grounded through the AC transformer)	type F (female)
• Physical dimensions	256.5 x 158.8 x 44.5 mm 10.1 x 6.3 x 1.8 in. (L x W x H)

Appendix C. IBM PC Network Protocols

This appendix provides information about the IBM PC Network protocols.[1] The following list describes the formats of the packets that are carried by the network in response to command requests.

> **Note:** The information provided here is for reference only. Specific details may vary slightly from the actual implementation.

Purpose of the Protocols

Personal Computers can exchange data with each other using the communication services provided by the IBM PC Network Adapters. These adapters accomplish communication functions through the implementation of a common set of protocols.

- Protocols are rules to govern communication over the network. These rules include such functions as data transmission, reception, and acknowledgment over the cable.
- Protocols are needed because multiple computers share a common cable medium. The computers must communicate with each other over the cable medium reliably without interference from one another. The protocols aid in reliable communications, provide addressing for all computers on the network, and control the flow of data traveling to other computers on the network.
- These common protocol functions are available to all computers on the network.

[1] Copyright of Sytek, Inc. and the IBM Corp.

To be compatible with the network, all computers must follow the PC Network protocol procedures and must understand the protocol formats.

Hierarchical Implementation of Protocols

An implementation of the PC Network protocols can be modularly structured in a hierarchical fashion. Compatibility between nodes does not require that the internal implementation of the adapter protocols within each nodes be identically structured. It is only required that they follow the same procedures and protocol formats.

Within the adapter, the software implementation of the protocols are hierarchically organized according to the specific protocol function.

- The adapter's protocol implementation comprises five separate layers of protocol modules.

- Software is arranged in a hierarchy of protocol layers. Each layer communicates with one layer above and one layer below it. Each layer provides specific services to the other layers.

- Layered structure allows easy development and change of the protocol software, and clear straightforward communication with other devices on the cable.

- The lowest layer is the hardware that transmits and receives signals over the cable. It converts the digital data to analog signals for transmission over the cable, and receives analog signals and converts them to digital data for use by the adapter.

- The highest layer communicates with the software resident in the host computer. The layer conveys the necessary data to the user application program.

- Data transmission process is as follows:

- The top layer of the adapter's protocol software receives data to be sent over the network from the host computer's interface software.

- Each layer of the protocol software adds control information to the data it receives from the next higher layer, and passes this data packet down to the next lower layer.

- The packet that is transmitted over the cable has the original data, and all of the specific control information to be interpreted by the receiving adapter's protocol software.

• Data Reception Process:

  - The bottom layer of the adapter's protocol implementation receives a signal from the network's cable.

  - This layer converts the incoming analog signal into digital data, and passes this data to the next higher layer of the protocol.

  - Each layer of the protocol software in the adapter reads and interprets control information inserted by the corresponding layer of the transmitting unit. Each layer then passes the contents of its data packet up to the next higher layer.

  - The top layer delivers the original data to a layer of communication software in the host computer. This data in turn will correspond to the user which sent the data from the originating computer.

The following is a list of the IBM PC Network adapter protocols under the appropriate layers:

• Link Layer

Link Access Protocol (LAP)[2]— provides basic CSMA/CD, packet framing, addressing, and error detection services. LAP is responsible for the exchange of data frames between two nodes. LAP is used to provide service for the Packet Transfer Protocol (PTP)[2].

- Network Layer

  Packet Transfer Protocol (PTP)[2]—provides routing, address discovery, and unacknowledged packet transfer services. PTP is used by the Reliable Stream Protocol (RSP)[2] and Datagram Transport Protocol (DTP)[2].

- Transport Layer

  Reliable Stream Protocol (RSP)[2]—provides error-free virtual connection services to other users through end-to-end acknowledgements and retransmissions. RSP provides transport layer services to the Session Management Protocol (SMP)[2].

  Datagram Transport Protocol (DTP)[2]—provides unacknowledged datagram services between session layer entities, including the User Datagram Protocol (UDP)[2] and the Diagnostic and Monitoring Protocol (DMP)[2].

- Session Layer

  Session Management Protocol (SMP)[2]—provides support for user sessions between nodes. SMP allows users to establish connection to a named process (names) and is responsible for interacting with the Name Management Protocol (NMP)[2] within the local node to determine the address of the named process. Once the destination node

---

2 Copyright of Sytek, Inc.

address is determined, the initiating SMP can communicate with the SMP within the destination node to provide session level services to both users.

User Datagram Protocol (UDP)[2]—provides support for user datagrams between nodes. UDP allows users to send datagrams to a named process (alias) and is responsible for interacting with the NMP within the local node to determine the address of the named process. Once the destination node address is determined, the initiating UDP can exchange datagrams with the UDP within the destination node.

Name Management Protocol (NMP)[1]—provides the binding of alias names and network addresses within the entire local network. NMP provides all name management services, including the translation of remote names to network addresses, to both SMP and UDP.

Diagnostic and Monitoring Protocol (DMP)[1]—provides protocol mechanisms that allow the collection of diagnostic and status information, and provides support for other network management functions.

SMP, UDP, NMP, and DMP services are accessible to the Host Interface Process.

The relationship between the various protocol services is shown in the following illustration.

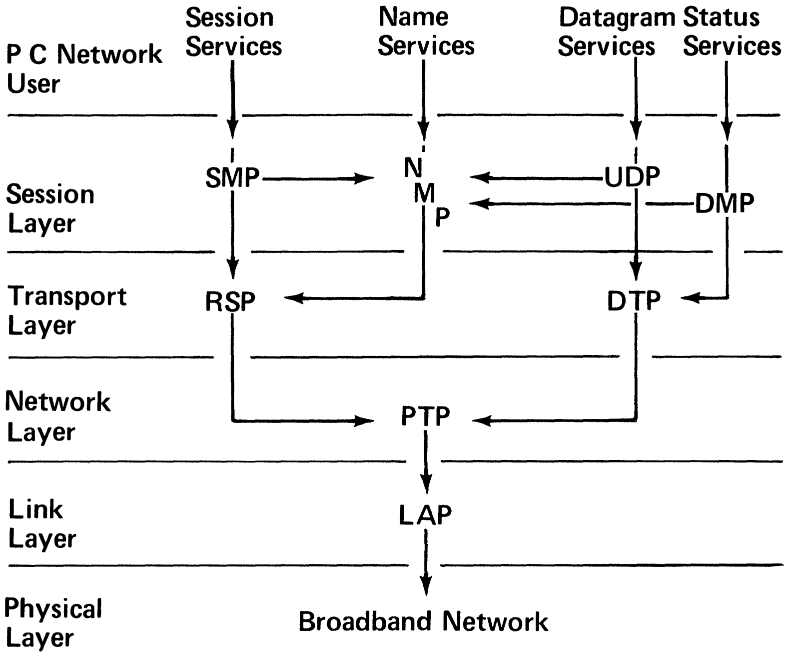


Figure C-1 Relationship Between the Various Protocol Services

General timing of packet exchanges

The following three sections describe the packet exchange interactions for session establishment, data transfer, and session termination.

Session Establishment

The session initiator sends an Open Request to the responder who returns an Open Ack. The initiator then sends a Session request that is followed by a Session Accept or Session Reject.

Session Establish

Initiator		**Responder**
----->	Open Request	----->
<-----	Open Ack	<-----
----->	Session Request	----->
<-----	Session Accept or Reject	<-----

Data Transfer

Both sides of a session exchange Data and Ack / Nack packets during a session's lifetime. Data packets have increasing sequence numbers, and are retransmitted if not acknowledged. Ack / Nack packets may request retransmission of specific packets. Data packets always have the same sequence number when retransmitted.

Data Transfer

Initiator		**Responder**
----->	Data	----->
<-----	Ack / Nack	<-----
----->	Data	----->
----->	Data (Retransmitted)	----->
<-----	Ack / Nack	<-----

Session Termination

Either side of a session can initiate termination. This is accomplished through an exchange of specific Ack / Nack, Close, and Closed messages.

Session Termination

Initiator		**Responder**
<-----	Close	<-----
----->	Close	----->
<-----	Closed	<-----

Packet Processing

This section describes the processing performed by the IBM PC Network adapter. These actions are described by pseudo code listings of processing initiated by the host commands, by processing that is initiated by timer expirations, and by processing initiated by received packets.

The following basic internal mechanisms of the adapter are required for the processing of the adapter protocols:

- The name table in the local adapter is the table that contains the names that are currently being supported by the local node. Names are added to the table upon successful completion of an ADD NAME or ADD GROUP NAME command. Names are removed upon successful completion of a DELETE NAME command. The user can refer to a current entry within the local name table by using the name number (NAME__NUM).

- Several types of packets within the adapter protocols must be retransmitted if a timely response is not received. Each such transmitted packet will have an associated timer and a counter. The timer starts when a packet is transmitted. When expiration time for a timer is reached, the packet will be retransmitted provided that the maximum number of transmissions has not been reached. The maximum number of transmissions is a system parameter known as "maxRetransmit".

- Each node on the network is assigned a unique identifier known by its permanent node name. In the following protocol descriptions, the permanent node name of the local node is simply referred to as the "local node name".

- Packets can be directed specifically to a node by using the permanent node name. The packets can also be directed to a group of nodes by using a group address. Group addresses are derived from a 16-byte name by using the function (f) defined as follows:

```
f(name) = 0000.(N1 XOR N2 . . . N5 XOR N6).FF

Where:
     N1 . . . N5 are the first through the fifth
                 3 byte fields which make up the
                 name, and N6 is the last byte of
                 the name concatenated with two
                 bytes of zeros.

         XOR     represents a logical exclusive
                 or operation.

         The period (.) represents the
         concatenation operation.
```

Group addresses are also derived from the permanent node name by using the function (g) as follows:

```
g(permanent node name) = 0000.(ID3 ID2 ID1).FF

Where:

     ID3 . . . ID1 are the three low-order bytes
                   of the permanent node name.

         The period (.) represents the
         concatenation operation.
```

The adapter processing can be described in an event driven manner where the three types of events are as follows:

1. Receipt of a command block from the host.

Such events are processed by using command specific procedures which are identified below as "PROCEDURE command__type (command__block)". Command__type is the name of the specific type of command, and the command__block includes all the parameters associated with the command as it is passed from the host to the adapter.

Also note that "return . . . to user" indicates the return of a completion of the processing associated with that command. Although certain fields within a returned command block can be inspected and used to provide the necessary information to the user, it does not indicate that there will be no further adapter activity associated with the processing of that command.

2. Expiration of the timers.

Such timer activity is always associated with a specific prior packet transmission and the action taken by the adapter in response to such timer activity depends on the original associated packet type.

3. Reception of a packet.

The processing performed by the adapter depends on the type of packet received as well as the values of specific control fields within the packet.

The next section describes the adapter processing for each of the above mentioned types of events.

Pseudo Code for NCB Commands

RESET

```
PROCEDURE Reset (command__block);
      {set configuration parameters}
BEGIN
IF either of the two configuration
   parameters are equal to zero THEN;
   set configuration parameters to default values
   {6 sessions, and 12 commands}
ELSE
   BEGIN
   set configuration parameters in configuration table;
   return command__completed status to user;
   END
END;
```

STATUS

```
PROCEDURE Status (command__block);
   {get LANA configuration parameters and status}
BEGIN
IF buffer length is illegal THEN;
   return illegal__buffer__length status to user;
ELSE
   IF name in conflict THEN;
      return name conflict status to user;
   ELSE IF name is local name or name is * THEN;
      BEGIN
      get configuration parameters and status of local LANA;
      return configuration parameters and status;
      return actual length of configuration
         parameter and status;
      IF size of user buffer is smaller than the configuration
         parameter and status THEN;
         return message incomplete status to user;
      ELSE
         return command__completed status to user;
      END
   ELSE
      BEGIN
      send status__request packet to remote node;
      wait for response from remote node;
         {remote node send status__response packet}
      IF status__response packet is received within the timeout
         interval THEN;
         BEGIN
         extract configuration parameters and
            status of remote LANA;
         return configuration parameters and status;
         return actual length of configuration
            parameter and status;
         IF size of user buffer is smaller than the configuration
            parameter and status THEN;
            return message incomplete status to user;
         ELSE
            return command__completed status to user;
```

```
        END
    ELSE
        return command__timed__out status to user;
    END
END;
```

ADD NAME

```
PROCEDURE Add__Name (command__block);
    {request local registration of name}
BEGIN
IF name begins with * or null THEN;
  return illegal__name status to user;
ELSE
  BEGIN
  search for name in local name table;
  IF name is found THEN;
    BEGIN
      IF name in conflict THEN;
        return name conflict status to user;
      ELSE
      return existing name__number to user;
      return duplicate__name status to user;
    END
  ELSE
    IF local name table is full THEN;
      return name__table__full status to user;
    ELSE
      BEGIN
      REPEAT
        BEGIN
        broadcast name__claim packet to network;
        wait for response from remote node in network;
        {all remote nodes search for name in their
        local name tables and send name__claim__response
        packet if found}
        END
      UNTIL name__claim__response packet is received
        or number of times to broadcast is reached;
      IF name__claim__response packet is not
      received THEN;
        BEGIN
        enter name to local name table;
        return name__number to user;
        return command__completed status to user;
        END
```

```
      ELSE
        return name__in__use status to user;
      END
  END
END;
```

ADD GROUP NAME

```
PROCEDURE Add__Group__Name (command__block);
    {request local registration of name}
BEGIN
IF name begins with * or null THEN;
  return illegal__name status to user;
ELSE
  BEGIN
  search for name in local name table;
  IF name is found THEN;
    BEGIN
      IF name in conflict THEN;
        return name conflict status to user;
      ELSE
      return existing name__number to user;
      return duplicate__name status to user;
    END
  ELSE
    IF local name table is full THEN;
      return name__table__full status to user;
    ELSE
      BEGIN
      REPEAT
        BEGIN
        broadcast add__group__name__claim
          packet to network;
        wait for response from remote node in network;
        {all remote nodes search for name in their
        local name tables and send name__claim__response
        packet if found}
        END
      UNTIL name__claim__response packet is received or
        number of times to broadcast is reached;
      IF name__claim__response packet is not
      received THEN;
        BEGIN
        enter name to local name table;
        return name__no to user;
        return command__completed status to user;
```

```
        END
      ELSE
        return name__in__use status to user;
      END
  END
END;
```

DELETE NAME

```
PROCEDURE Delete__Name (command__block);
    {request local de-registration of name}
BEGIN
IF name begins with * or null THEN;
  return illegal__name status to user;
ELSE
  BEGIN
  search for name in local name table;
  IF name is found THEN;
    BEGIN
    check for pending non active session commands;
    IF non active session command is found THEN;
      terminate the non active session command;
    check the session count in table entry;
    IF session count is zero THEN;
      BEGIN
      remove name from local name table;
      return command__complete status to user;
      END
    ELSE
      BEGIN
      change status of name to de-registered;
        {name will be removed from local name
        table when session count reaches zero}
      return command__completed__name__has__
        active__session status to user;
      END
    END
  ELSE
    return illegal__name status to user;
  END
END;
```

CALL

```
PROCEDURE Call (command__block);
   {open user session with remote name using name supplied}
BEGIN
IF local resource is not available THEN;
   return no__resource__available status to user;
ELSE
   BEGIN
   search for name in local name table;
   IF name is not found THEN;
      return illegal__name status to user;
   ELSE
   IF name is marked "conflict detected" THEN;
      return name__conflict__detected status to user;
   ELSE
      IF local session table is full THEN;
         return local__session__table__full status to user;
      ELSE
         BEGIN
         search for remote name in local name table;
            BEGIN
            REPEAT
               BEGIN
               IF name is not found THEN
                  broadcast name__query packet to network;
               ELSE loop back name query packet;
                  wait for response from remote node or
                  timeout;
                  {all remote nodes search for name
                  in their local name tables and send
                  name__query__response packet if found}
               END
            UNTIL name__query__response packet is received
               or number of times to broadcast is
               reached;
            IF name__query__response packet is not
               received THEN;
               return unknown__remote__name status to user;
            END
```

```
      BEGIN
      send session__request packet to destination
         node;
         {destination node search for pending
         LISTEN command, if found, return
         session__accept packet, else, return
         session__reject packet}
      IF network error THEN;
         return "session aborted" to user;
      IF response from destination is received
         within the timeout interval THEN;
         CASE response OF
            session__accept packet:
               BEGIN
               set session__established indicator
                  in session table;
               return local__session__number
                  to user;
               return command__completed status
                  to user;
               END
               {session established}
            session__reject packet:
               return session__open__rejected status
                  to user;
         END {case}
      ELSE
         return command__timed__out status to user;
      END
   END
 END
END;
```

LISTEN

```
PROCEDURE Listen (command__block);
   {open user session with remote name, using
   name supplied}
BEGIN
IF local resource is not available THEN;
   return no__resource__available status to user;
ELSE
   BEGIN
   search for name in local name table;
   IF name is not found THEN;
      return illegal__name status to user;
   ELSE
      IF name is marked "conflict detected" THEN
         return "name conflict detected" status to user;
      ELSE
         IF local session table is full THEN;
            return local__session__table__full status to user;
         ELSE
            BEGIN
            REPEAT
               BEGIN
               wait for session__request packet
                  or name__conflict__detection;
               IF LISTEN specific is specified THEN;
                  check source of session__request packet;
                  IF source of session__request pack is
                     same as remote name THEN;
                     set session__request__completed indicator;
                     {LISTEN specific satisfied}
                  ELSE
                     reset session__request__completed
                        indicator;
                     {LISTEN not satisfied, continue
                     to wait}
               ELSE
                  IF LISTEN any specified THEN
                  set session__request__completed indicator;
                  {LISTEN ANY satisfied}
```

```
      END
    UNTIL session__request__completed indicator is set
      or name__conflict__detected;
    IF name__conflict__detected for outstanding
      LISTEN THEN
    ELSE IF network error THEN;
      return "session aborted" status to user;
       ELSE
         send session__accept packet to source;
         wait for first packet on session;
         set session__established
           indicator in session table;
         return source of session__request packet to user;
         return local__session__number to user;
         return command__completed status to user;
         {session established}
     END
  END
END;
```

HANG UP

```
PROCEDURE Hang Up (command__block);
   {close user session indicated by
   local__session__number}
BEGIN
IF local session number is illegal THEN;
   return illegal__session__number status to user;
ELSE
   IF session is already closed and session__closed status has
      not been reported THEN;
      return session__closed status to user;
   ELSE
      IF session is already aborted and session__aborted
         status has not been reported THEN;
         return session__aborted status to user;
      ELSE
         BEGIN
         REPEAT
            IF a RECEIVE command is pending THEN;
               terminate RECEIVE command with a
               session__closed status to user;
         UNTIL all pending RECEIVE commands
            are terminated;
         REPEAT
            IF a SEND or CHAIN SEND command
               is pending THEN;
               wait until the SEND, CHAIN SEND, or
                  HANG UP has completed or
                  timed out;
            IF the SEND command was timed out THEN
               abort the session;
         UNTIL all pending SEND and CHAIN SEND
            commands are completed or timed out;
         send close packet to destination node;
         wait for close packet from destination node or
            close timeout;
            {destination node close the session
            and send close packet}
         IF close packet is received before the close
```

```
        timeout interval THEN;
        BEGIN
        close the session;
        return command__completed status to user;
        END
    ELSE
        BEGIN
        abort the session;
        return session__aborted status to user;
        END
    IF RECEIVE to name command is pending THEN;
        terminate RECEIVE to name command with
        session__closed or session__aborted status;
    ELSE
        IF a RECEIVE ANY or to any name command
        is pending THEN
            Terminate the RECEIVE ANY command
            with session closed or session
            aborted status;
    END
END;
```

SEND

```
PROCEDURE Send (command__block);
   {send data through user session as indicated by
   local__session__number}
BEGIN
IF local session number is illegal THEN;
   return illegal__local__session__number
   status to user;
ELSE
   IF session is closed and session__closed status has not
      been reported THEN;
      return session__closed status to user;
   ELSE
      IF session is aborted and session__aborted status has
         not been reported THEN;
         return session__aborted status to user;
      ELSE
         BEGIN
         send session data packet(s) to destination node;
         wait for ack packet(s) from destination node
            or for timeout;
         IF session data is sent successfully within the
            timeout interval for session send THEN;
            return command__completed status to user;
         ELSE
            BEGIN
            abort the session;
            return command__timed__out status to user;
            END
         END
END;
```

CHAIN SEND

```
PROCEDURE Chain Send (command__block);
   {send data through user session as indicated by
   local__session__number}
BEGIN
IF local session number is illegal THEN;
   return illegal__local__session__number status to user;
ELSE
   IF session is closed and session__closed status
      has not been reported THEN;
      return session__closed status to user;
   ELSE
      IF session is aborted and session__aborted
         status has not been reported THEN;
         return session__aborted status to user;
      ELSE
         BEGIN
         send session data packet(s) to destination node;
         wait for ack packet(s) from destination node
            or for timeout;
         IF session data is sent successfully within the
            timeout interval for session send THEN;
            return command__completed status to user;
         ELSE
            BEGIN
            abort the session;
            return command__timed__out status to user;
            END
         END
END;
```

RECEIVE

```
PROCEDURE Receive (command__block);
   {receive data through user session as indicated by
   local__session__number}
BEGIN
IF local session number is illegal THEN;
   return illegal__local__session__number
      status to user;
ELSE
   IF session is closed and session__closed status has not
      been reported THEN;
      return session__closed status to user;
   ELSE
      IF session is aborted and session__aborted status has
         not been reported THEN;
         return session__aborted status to user;
      ELSE
         BEGIN
         wait for session message (data packet(s))
            from source node;
         IF session data is received within the timeout
            interval for session receive THEN;
            BEGIN
            send ack packets(s) to source node;
            transfer session data to user buffer
               of appropriate length;
            return actual length of transfer to user;
            IF size of user buffer is smaller than
               received session data THEN;
               return message incomplete status to user;
            ELSE
               return command__completed status to user;
            END
         ELSE
            return command__timed__out status to user;
            {session data received}
         END
END;
```

RECEIVE ANY

```
PROCEDURE Receive__Any (command__block);
  {receive any data sent to the
  specified name__number}
BEGIN
IF name number is illegal THEN;
  return illegal__name__number
    status to user;
ELSE
  IF session is closed and session__closed
    status has not been reported THEN;
    return session__closed status to user;
  ELSE
    IF session is aborted and session__aborted
      status has not been reported THEN;
      return session__aborted status to user;
    ELSE
      BEGIN
      REPEAT
        BEGIN
        wait for a session message (data packet(s));
        IF RECEIVE specific is specified THEN;
          BEGIN
          check recipient name in session message;
          IF recipient name in session message is
            same as local name THEN;
            set receive__completed indicator;
            {receive to specific name satisfied}
          ELSE
            reset receive__completed indicator;
            {receive to specific not satisfied,
            continue to wait}
          END
        ELSE
          set receive__completed indicator;
          {receive to any name satisfied}
```

```
    END
  UNTIL receive__completed indicator is set;
  IF "conflict detected" error THEN;
    return name__conflict__detected status to user;
  ELSE
    send ack packet(s) to sending node;
    transfer session data to user buffer
      of appropriate length;
    return actual length of transfer to user;
    return recipient name number to user;
    return local__session__number to user;
    IF size of user buffer is smaller than
      received session data THEN;
      return message incomplete status to user;
    ELSE
      return command__completed status to user;
      {session data received}
  END
END;
```

SESSION STATUS

```
PROCEDURE Session__Status (command__block);
   {obtain status of session indicated by name}
BEGIN
IF buffer length is illegal THEN;
   return illegal__buffer__length status to user;
ELSE
   BEGIN
      IF name in conflict THEN
         return "name conflict detected" status to user;
      IF name does not start with * THEN;
         search for name in local name table;
         return number of pending sessions;
         return number of pending datagram receives;
         IF name or * is found THEN;
            BEGIN
            get session status in session table
               for each session on the name or;
               for all session if * ;
               return session status to user;
               return actual length of session status;
            IF size of user buffer is smaller than
               size of session status data THEN;
               return message incomplete status to user;
            ELSE
               return command__completed status to user;
            END
      ELSE
         return illegal__name status to user;
      END
END;
```

SEND DATAGRAM

```
PROCEDURE Send Datagram (command__block);
  {send datagram to remote node with the specified
  name registration}
BEGIN
IF buffer length is illegal THEN;
  return illegal__buffer__length status to user;
ELSE
  BEGIN
  search for name corresponding to name number
    in local name table;
  IF name in conflict THEN;
    return name conflict status to user;
  ELSE
    IF name is not found THEN;
      return illegal__name__number status to user;
    ELSE
      BEGIN
      send datagram to destination node;
      return command__completed status to user;
      END
  END
END;
```

SEND BROADCAST DATAGRAM

```
PROCEDURE Send__Broadcast__Datagram (command__
      block);
   {send broadcast datagram to all nodes on the network}
BEGIN
IF buffer length is illegal THEN;
   return illegal__buffer__length status to user;
ELSE
   BEGIN
   search for name in local name table;
   IF name in conflict THEN;
      return name__conflict status to user;
   ELSE
      IF name is not found THEN;
         return illegal__name__number status to user;
      ELSE
         BEGIN
         broadcast datagram to all nodes on the network;
         return command__completed status to user;
         END
   END
END;
```

RECEIVE DATAGRAM

```
PROCEDURE Receive Datagram (command__block);
   {receive datagram from any node on the network}
BEGIN
search for name corresponding to name number
   in local name table;
IF name is not found THEN;
   return illegal__name__number status to user;
ELSE
   BEGIN
   REPEAT
      BEGIN
      wait for arrival of datagram;
      IF datagram receive specific is specified THEN;
         BEGIN
         check recipient name in datagram message;
         IF recipient name in datagram message is same
            as name specified by local name
            number THEN;
            set receive__completed indicator;
            {datagram receive to
            specific name satisfied}
         ELSE
            reset receive__completed indicator;
            {datagram receive not satisfied,
            continue to wait}
         END
      ELSE
         set receive__completed indicator;
         {datagram receive to any name satisfied}
      END
   UNTIL receive__completed indicator is set;
   IF "conflict detected" error THEN;
      return name__conflict__detected status to user;
   ELSE
      transfer datagram data to user buffer
         of appropriate length;
      return actual length of transfer to user;
      return local name__number to user;
```

```
    return sender's name to user;
    IF size of user buffer is smaller than
      received datagram THEN;
      return message incomplete status to user;
      {data received, unable to
      transfer entire message}
    ELSE
      return command__completed status to user;
      {datagram received}
  END
END;
```

RECEIVE BROADCAST DATAGRAM

```
PROCEDURE Receive Broadcast Datagram (command__
      block);
  {receive broadcast datagram from
  any node in the network}
BEGIN
search for name corresponding to name number in
  local name table;
IF name is not found THEN;
  return illegal__name__number status to user;
ELSE
  wait for arrival of broadcast datagram;
  IF "conflict detected" error THEN;
    return name__conflict__detected status to user;
  ELSE
    transfer datagram to user buffer
      of appropriate length;
    return actual length of transfer to user;
    return sender's name to user;
    IF size of user buffer is smaller than received
      datagram THEN;
      return message incomplete status to user;
      {broadcast datagram received, unable to return
      entire message}
    ELSE
      return command__completed status to user;
      {broadcast datagram received}
END;
```

Timer Expiration Processing

This section describes the pseudo code processing that is performed when a timer expires.

```
PROCEDURE packet associated timer__expiration
BEGIN
CASE outstanding packet type OF

  name query:
    IF retransmit counter is less than
      maxRetransmit THEN;
      increment retransmit counter;
      reset and restart timer;
      send another name__query packet;
    ELSE
      IF non-accept query response
        received THEN
        return non-accept query
          status to user;
      ELSE
        return unknown__remote__name
          status to user;
        send name__query__cancel packet;

  name claim:
    IF retransmit counter is less than
      maxRetransmit THEN
      increment retransmit counter;
      reset and restart timer;
      send another name__claim packet;
    ELSE
      return name__number and command
        completed status to user;

  nonexclusive name claim;
    IF retransmit counter is less than
      maxRetransmit THEN
      increment retransmit counter;
      reset and restart timer;
```

```
        send another nonexclusive name
           claim packet;
     ELSE
        return name__number and command
           completed status to user;

  session request:
     IF retransmit counter is less than
        maxRetransmit THEN
        increment retransmit counter;
        reset and restart timer;
        send another session__request packet;
     ELSE
        return command__timed__out status to user;

  session accept:
     IF retransmit counter is less than
        maxRetransmit THEN
        increment retransmit counter;
        reset and restart timer;
        send another session__accept packet;
     ELSE
        send abort packet;
        return session__aborted to user;

  positive name query response:
     release connection resources;
     IF name not registered more than once THEN
        return name conflict status to user;

  data:
     IF retransmit counter is less than
        maxRetransmit THEN;
        increment retransmit counter;
        reset and restart timer;
        IF retransmit queue is empty THEN
           format and send ack packet with
           POLL set;
        ELSE
           send last packet in the retransmit
           queue with POLL set;
     ELSE
```

```
        abort the session;
        return session aborted, no
           response status to user;

  close:
     IF retransmit counter is less than
        maxRetransmit THEN
        increment retransmit counter;
        reset and restart timer;
        send another close packet;
     ELSE
        abort the session;
        return session aborted, no
           response status to user;

  status request:
     IF retransmit counter is less than
        maxRetransmit THEN
        increment retransmit counter;
        send another status__request packet;
     ELSE
        return no__response status to user;

END CASE
```

```
PROCEDURE packet associated timer__expiration
BEGIN
CASE outstanding packet type OF

   session__send:
      send abort packet;
      return session__send with command
         timed out status;

   session__send__multiple:
      send abort packet;
      return session__send__multiple
         with command timed out status;

   session__receive:
      return session__receive with
         command timed out status;

END
```

Packet Reception Processing

This section describes the receipt of packets by the protocol software. The conditions for this type of processing are as follows:

- The packet has been received.
- The packet has gone through CRC checking.
- The address of this node has matched the address in the packet.

```
PROCEDURE packet__received
BEGIN
CASE packet type OF

  name query:
    IF DNODEID does not match local node name or any
    currently enabled group address THEN;
      ignore packet;
    ELSE
      IF packet arrived with same SNODEID
        and DID THEN;
        ignore this packet;
      ELSE
      IF DNAME is stored in local name table THEN
        IF resources exist for a new session THEN
          send positive__name__query__response
          packet to sender and start timer;
        ELSE
          send negative__name__query response
            packet to sender;
      ELSE
        ignore packet and send no response;

  name claim:
    IF DNODEID does not match local node name or any
    currently enabled group address THEN
      ignore packet;
```

```
    ELSE
      IF packet arrived with same SNODEID
      and DID  THEN
        ignore this packet;
      ELSE
      IF DNAME conflicts with a name
        in local name table THEN
        send name__claim__response packet to sender;
      ELSE
        ignore packet and send no response;

nonexclusive name claim:
    IF DNODEID does not match local node name or any
    currently enabled group address THEN
      ignore packet;
    ELSE
      IF packet arrived with same SNODEID
      and DID  THEN
        ignore this packet;
      ELSE
      IF DNAME conflicts with a name exclusively
        registered in local name table THEN
        send name__claim__response packet to sender;
      ELSE
        ignore packet and send no response;

positive name query response:
    match this packet with the originating active open;
    cancel timer associated with previous name query;
    identify the session established by the SNCID and the
    CONNID values;
    send session request packet;
    start timer for session request packet;

negative name query response:
    match this packet with the originating active open;
    cancel timer associated with previous name query;
    return command__completed, session
      open rejected to user

name query cancel:
    match this packet with any half-established connection
    (SNODEID and SCONNID);
```

```
    IF match is found THEN
      release associated connection resources;
      issue tear down to lower layer;
    ELSE
      ignore this packet;

  name claim response:
    remove name from local name table;
    cancel timer associated with previous name claim;
    return name__in__use status to user;

  session request:
    cancel positive name query response timer;
    IF passive open specific is outstanding with remote name
    equal to SNAME and local name equal to DNAME THEN
      send session__accept packet over session;
      start timer for session__accept;
      choose local__session__no for this session;
      return command__completed status to user;
    ELSE
    IF passive open non-specific is outstanding with local name
    equal to DNAME THEN
      send session__accept packet over session;
      start timer for session__accept;
      choose local__session__number
        for this session;
      set remote name = SNAME;
      return command__completed status to user;
    ELSE
      send session__rejected packet over session;

  session accept:
    cancel timer associated with previous session request;
    choose local__session__number for this session;
    return local__session__number to user;
    return command__completed status to user;
    {session established}

  session reject:
    cancel timer associated with previous session request;
    return session__open__rejected status to user;
```

```
session data:
   remove all acknowledged packets from retransmit queue;
   note remote entity's current window value (WIN);
   cancel all associated timers;
   IF SEQ is not the next expected value THEN
      IF nack packet has not been sent THEN
         send nack packet;
      ELSE
         ignore packet;
   ELSE
      {SEQ is next expected}
      IF POLL is indicated THEN
         send ack;
         transfer session data to user buffer
         specified in session__receive
         command;
      IF size of user buffer is smaller than
         received session data THEN
            return message incomplete status to user;
      ELSE
         {EOM is indicated}
         return actual amount received to user;
         return command__completed status to user
   REPEAT for each outstanding session__send and
        session__send__multiple command;
      IF all data has been acked THEN
        return command with success status;
   UNTIL all session__send and session__send__multiple
        commands have been examined;
      IF no session__send or session__send__multiple
        commands are pending AND a session__close is
        pending THEN
        send a close packet;

ack:
   remove all acknowledged packets from the retransmit
   queue;
   note remote entity's current window value (WIN);
   restart timers on unacknowledged packets;
   cancel timers on acknowledged packets
   REPEAT for each outstanding session__send and
        session__send__multiple command;
      IF all data has been acked THEN
```

```
      return command with success status;
  UNTIL all session__send and session__send__multiple
      commands have been examined;
    IF no session__send or session__send__multiple
      commands are pending AND a session__close is
      pending THEN
      send a close packet;

nack:
  remove all acknowledged packets from the retransmit
  queue;
  note remote entity's current window value (WIN);
  restart timers on unacknowledged packets;
  cancel timers on acknowledged packets;
  retransmit last unacknowledged (last) packet(s);
  REPEAT for each outstanding session__send and
      session__send__multiple command;
    IF all data has been acked THEN
      return command with success status;
  UNTIL all session__send and session__send__multiple
      commands have been examined;
    IF no session__send or session__send__multiple
      commands are pending AND a session__close is
      pending THEN
      send a close packet;

close:
  IF local user initiated the close THEN
    send closed packet;
    close the session;
    return command__completed status to user
  ELSE
    {close packet received from close initiator}
    initialize retry counter;
    initialize retransmission timer;
    send close packet to close initiator

closed:
  notify user that connection is closed;

user datagram:
IF DNAME is not *broadcast THEN
  BEGIN {non-broadcast datagram received}
```

```
IF previous packet arrived with
same SNODEID and DID THEN
  ignore this packet
IF no pending datagram receive command exists THEN
  ignore packet
REPEAT
   {check if received datagram matches a
   receive specific}
   choose next pending datagram receive specific
   IF DNAME = local name THEN
      BEGIN
      {datagram receive to specific name
      satisfied}
      transfer datagram data to user buffer
         of appropriate length;
      return actual length of transfer to user;
      return local name__number to user;
      return SNAME to user;
      IF size of user buffer is smaller than
      received datagram THEN
         BEGIN
         return message incomplete status to user;
         {datagram received, unable to transfer
         entire message}
         END
      ELSE
         return command__completed status to user
         {datagram received}
      END
UNTIL all datagram receive specific commands checked
{datagram receive to any name satisfied}
transfer datagram data to user buffer;
return actual length of transfer to user;
return local name__number to user;
return SNAME to user;
IF size of user buffer is smaller than
received datagram THEN
   BEGIN
      return message incomplete status to user;
      {datagram received, unable to
      transfer entire message}
   END
ELSE
```

```
        return command_completed status to user
        {datagram received}
  END  {non-datagram received}
  ELSE  {broadcast datagram received}
    BEGIN
    IF broadcast datagram is sent by the
    same local name THEN
      ignore datagram
      {broadcast datagram receive no satisfied,
      continue to wait}
    ELSE
      IF previous packet arrived with
      same SNODEID and DID THEN
        ignore this packet
      IF no broadcast datagram receive
      command outstanding THEN
        ignore packet
      ELSE
        choose next pending broadcast datagram receive
        transfer broadcast datagram to user buffer;
        return actual length of transfer to user;
        return sender's name to user;
        IF size of user buffer is smaller than received
        datagram THEN
          return message incomplete status to user;
          {broadcast datagram received, unable to
          return entire message}
        ELSE
          return SNAME to user;
          return command_completed status to user
          {broadcast datagram received}
    END  {broadcast datagram received}

  status request:
    IF DNODEID does not match local node name or any
    currently enabled group address THEN
      ignore packet
    ELSE
      IF previous packet recently arrived with same
      SNODEID and DID THEN
        ignore this packet
      ELSE
        search for name in local name table;
```

```
        IF name is found THEN
          send status response packet;
        ELSE
          ignore this packet;

  status response:
    IF DNODEID does not match local node name or any
    currently enabled group address THEN
      ignore packet
    ELSE
      IF previous packet arrived with same SNODEID
      and DID THEN
        ignore this packet
      ELSE
        cancel timer associated with previous status
        request;
        return configuration parameters and status;
        return actual length of configuration parameter
        and status;
        IF size of user buffer is smaller than the
        configuration parameter and status THEN
          return message incomplete status to user;
        ELSE
          return command__completed status to user
END CASE;
```

Field Definitions

This section describes the fields found within the protocol packet types.

Field	Meaning
ACK	This field is an 8-bit field that includes the sequence number + 1 modulo 256 of the last correctly received packet.
ALGNERRS	This is a 16-bit field specifying the number of packets received with alignment errors.
ALIASNAME	This is the 16-byte field containing the name.
ALIASNR	This is an 8-bit specifying the number that is assigned to a given name.
ALIASTAT	The low order 4-bits specify the status of a name. See "* Local name table—16 entries of 18 bytes each" on page 2-35 for the numbers that are returned in this field.
CLREAS	This is an 8-bit reason code indicating the reason for a connection being closed. CLREAS = 00H indicates a normal close.
CMDRESP	This is a 1-bit flag indicating whether the packet contains a command or a response to a command.

COLLISIONS	This is 16-bit field specifying the number of transmitted packets that experienced collisions.
CONNID	This is a 16-bit identifier used to determine which session a packet is assigned to.
CRC	This is a 32-bit field containing the cyclic redundancy check for the packet according to the Autodin-II 32-bit polynomial generator.
CRCERRS	This is a 16-bit field specifying the number of CRC errors being reported.
DADDR	This 48-bit field is used to identify the destination Link level address of the packet. A value of all binary ones indicates a broadcast. All other adapter addresses will have the 16 highest bits set to zero. A value whose least significant bits are hex FF indicates a group address of the form f(name).
DATA	DATA is a variable length field containing user data.
DID	DID is a 16-bit field that is incremented with each retransmission of a name query, name claim, and a get status packet.
DLEN	This 16-bit field contains the length of the data field specified in bytes of the Link level packet.

DNAME This is a 16-bit field that identifies the name of a destination (ASCII characters).

DNCID This is a 16-bit field used with the CONNID to determine the session for each packet or to identify a datagram.

DNODEID This is a 48-bit field indicating the Link level address of an intended destination.

EFD This is an 8-bit end frame delimiter flag (7EH).

EOM This is a 1-bit end of message indicator. The bit marks the end of a user's logical message.

JUMPSTAT1 This is the 1-bit field indicating the status of jumper W1. When set to 1, jumper W1 is in place.

JUMPSTAT2 This is the 1-bit field indicating the status of jumper W2. When set to 1, jumper W2 is in place.

NACKREAS This is the 8-bit reason code indicating the reason why a packet was not successfully delivered. The following is a list of the codes that can be contained in this field.

- 00H–No specific reason
- 18H–Nack response to an unexpected open request
- 20H–Incompatible RSP version
- 22H–Bad service ID
- 24H–Invalid RSP control information
- 26H–No remote resources
- 32H–OK
- 34H–Name claim rejected

- 35H–Name query rejected

NODEIDMASK	This is a 6-byte mask used in a logical AND operation to get the destination node name.
NODEIDMATCH	This is the 6-byte match value to use in the operation: permanent node name AND NODEIDMASK equals NODEIDMATCH.
NRALIAS	This is the 16-bit field specifying the number of names to follow.
PKTSIZE	This is a 16-bit field indicating the maximum packet size that an initiating session node is willing to accept.
PNCID	This is a 16-bit field used with the CONNID to determine the network connection with which a packet is associated at a previous node or to identify a datagram.
PNODEID	This is a 48-bit field indicating the Link level address of the previous node.
POLL	This is a 1-bit field when set to 08H, indicates that a return packet should be generated back to the sender. 00H indicates no poll.
RECVMSGS	This is a 32-bit field specifying the number of packets that have been successfully received for a given period of time.
REPORTPD	This is a 16-bit field specifying the period of time, in minutes, for the statistical data that was gathered.

RVAL	This is an 8-bit value indicating the reason why a requested session was rejected. RVAL = 3 indicates no matching LISTEN command. RVAL = 4 indicates incompatible version.
SADDR	This is a 48-bit field identifying the source Link level address of the packet.
SCONNID	This is a 16-bit identifier used to determine the session number for a packet.
SEQ	SEQ is an 8-bit session packet sequence number. It is incremented with each *new* data transmission.
SHORTFRAMES	This is a 16-bit field specifying the number of short frames being reported.
SNAME	This is a 16-bit field containing the name of a source node. (Specified in ASCII characters)
SNCID	This is a 16-bit field used with the CONNID to determine the network connection with which a packet is associated at a source node or to identify a datagram.
SNODEID	This is 48-bit field indicating the Link level address of a source node.
STD	STD is an 8-bit start delimiter flag (7EH).

TRANSID	This is a 16-bit field indicating the transaction ID for status and status response packets.
WIN	WIN is a 4-bit field that indicates the number of packets beyond the ones already acknowledged that the sender is willing to accept.
XMITABRTS	This is a 16-bit field specifying the number of transmitted packets that were aborted.
XMITMSGS	This is a 32-bit field specifying the number of packets that were successfully transmitted for a given period of time.

Packet Types, Formats and Functions

With the previously defined field semantics, the following describes the protocol packet types and their functions.

Name Query Packet

Name Query Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 5000H	; Fixed value for ; this position of ; packet
N/A	11H	DB 10H	; Fixed value for ; this position of ; packet
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets

Name Query Packet

Field Name	Offset	Length	Comments
CONNID	13H	DW ?	; Field ID for ; packet
N/A	15H	DW 0202H	; Fixed value for ; this position of ; packet
N/A	17H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	19H	DW 0100H	; Fixed value for ; this position of ; packet
N/A	1BH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1DH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1FH	DW 10XXH	; Fixed value for ; this position of ; packet
N/A	21H	DW XX10H	; Fixed value for ; this position of ; packet
DNAME	23H	DB 16 DUP(?)	; ASCII Name for ; destination
SNAME	33H	DB 16 DUP(?)	; ASCII Name for ; source

Name Query Packet

Field Name	Offset	Length	Comments
PNCID	43H	DW ?	; (See definition)
DID	45H	DW ?	; Number that is ; incremented by ; 1 for each ; packet
SNCID	47H	DW ?	; Field ID for ; packet
DNODEID	49H	DW ?	; Low address
	4BH	DW ?	; Mid address
	4DH	DW ?	; Hi address
SNODEID	4FH	DW ?	; Low address
	51H	DW ?	; Mid address
	53H	DW ?	; Hi address
PNODEID	55H	DW ?	; Low address
	57H	DW ?	; Mid address
	59H	DW ?	; Hi address
CRC	5BH	DD ?	; Check byte
EFD	5FH	DB 7EH	; End-of-packet ; byte

DADDR This equals f(remote name).

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID	Choose a new CONNID value on the first transmission or repeat the previous CONNID on retransmissions.
XXXX	The X's represent a don't-care value.
DNAME	This is equal to the remote name.
SNAME	This is equal to the source name.
DID	Use the next DID value.
SNCID	Use the next SNCID value on the first transmission or repeat the previous SNCID value on retransmission.
PNCID	Low-byte of PNCID equals the low-byte of SNCID.
DNODEID	This is equal to f(remote name).
SNODEID	This is equal to the local node name
PNODEID	This is equal to SNODEID
CRC	This is the cyclic redundancy check for the packet.

Name Claim and Name Claim Cancel Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 5000H	; Fixed value for ; this position of ; packet
N/A	11H	DB ?	; 10H = Name Claim ; packet. A0H = ; Name Claim ; Cancel Packet
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets
CONNID	13H	DW ?	; Field ID for ; packet

Name Claim and Name Claim Cancel Packet

Field Name	Offset	Length	Comments
N/A	15H	DW 0202H	; Fixed value for ; this position of ; packet
N/A	17H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	19H	DW 0400H	; Fixed value for ; this position of ; packet
N/A	1BH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1DH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1FH	DW 10XXH	; Fixed value for ; this position of ; packet
N/A	21H	DW 0000H	; Fixed value for ; this position of ; packet
DNAME	23H	DB 16 DUP(?)	; ASCII Name for ; destination
PNCID	33H	DW ?	; Equal to SNCID

Name Claim and Name Claim Cancel Packet

Field Name	Offset	Length	Comments
DID	35H	DW ?	; Number that is ; incremented by ; 1 for each ; packet
SNCID	37H	DW ?	; Field ID for ; packet
DNODEID	39H	DW ?	; Low address
	3BH	DW ?	; Mid address
	3DH	DW ?	; Hi address
SNODEID	3FH	DW ?	; Low address
	41H	DW ?	; Mid address
	43H	DW ?	; Hi address
PNODEID	45H	DW ?	; Low address
	47H	DW ?	; Mid address
	49H	DW ?	; Hi address
CRC	4BH	DD ?	; Check byte
EFD	4FH	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of f(remote name).

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID	Choose a new CONNID value on the first transmission or repeat the previous CONNID on retransmissions.
XXXX	The X's represent a don't-care value.
DNAME	This is equal to the remote name.
PNCID	This is equal to SNCID.
DID	Use the next DID value.
SNCID	Use the next SNCID value on the first transmission or repeat the previous SNCID value on retransmission.
DNODEID	This is equal to f(remote name).
SNODEID	This is equal to the local node name
PNODEID	This is equal to SNODEID
CRC	This is the cyclic redundancy check for the packet.

Nonexclusive Name Claim and

Nonexclusive Name Claim Cancel Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 5000H	; Fixed value for ; this position of ; packet
N/A	11H	DB ?	; 10H=Nonexclusive ; Name Claim ; Packet. A0H=Non- ; exclusive Name ; Claim Cancel ; packet
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets

Nonexclusive Name Claim and

Nonexclusive Name Claim Cancel Packet

Field Name	Offset	Length	Comments
CONNID	13H	DW ?	; Field ID for ; packet
N/A	15H	DW 0202H	; Fixed value for ; this position of ; packet
N/A	17H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	19H	DW 0600H	; Fixed value for ; this position of ; packet
N/A	1BH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1DH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1FH	DW 10XXH	; Fixed value for ; this position of ; packet
N/A	21H	DW 0000H	; Fixed value for ; this position of ; packet
DNAME	23H	DB 16 DUP(?)	; ASCII Name for ; destination

Nonexclusive Name Claim and

Nonexclusive Name Claim Cancel Packet

Field Name	Offset	Length	Comments
PNCID	33H	DW ?	; Equal to SNCID
DID	35H	DW ?	; Number that is ; incremented by ; 1 for each ; packet
SNCID	37H	DW ?	; Field ID for ; packet
DNODEID	39H	DW ?	; Low address
	3BH	DW ?	; Mid address
	3DH	DW ?	; Hi address
SNODEID	3FH	DW ?	; Low address
	41H	DW ?	; Mid address
	43H	DW ?	; Hi address
PNODEID	45H	DW ?	; Low address
	47H	DW ?	; Mid address
	49H	DW ?	; Hi address
CRC	4BH	DD ?	; Check byte
EFD	4FH	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of f(remote name).

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

WIN	This equals the current value indicating the number of packets the sender is willing to accept.
CONNID	Choose a new CONNID value on the first transmission or repeat the previous CONNID on retransmissions.
XXXX	The X's represent a don't-care value.
DNAME	This is equal to the remote name.
PNCID	This is equal to SNCID.
DID	Use the next DID value.
SNCID	Use the next SNCID value on the first transmission or repeat the previous SNCID value on retransmission.
DNODEID	This is equal to f(remote name).
SNODEID	This is equal to the local node name
PNODEID	This is equal to SNODEID
CRC	This is the cyclic redundancy check for the packet.

Positive Name Query Response

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 3000H	; Fixed value for ; this position of ; packet
N/A	11H	DB 20H	; Fixed value for ; this position of ; packet
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW ?	; Field ID for ; packet

Positive Name Query Response

Field Name	Offset	Length	Comments
SCONNID	15H	DW ?	; Connection ; ID of ses- ; sion
N/A	17H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	19H	DW 0101H	; Fixed value for ; this position of ; packet
N/A	1BH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1DH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1FH	DW 10XXH	; Fixed value for ; this position of ; packet
N/A	21H	DW 0010H	; Fixed value for ; this position of ; packet
DNAME	23H	DB 16 DUP(?)	; ASCII Name for ; destination
SNAME	33H	DB 16 DUP(?)	; ASCII Name for ; source

Positive Name Query Response

Field Name	Offset	Length	Comments
DNCID	43H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
SNCID	45H	DW ?	; Field ID for ; packet
SNODEID	47H	DW ?	; Low address
	49H	DW ?	; Mid address
	4BH	DW ?	; Hi address
CRC	4DH	DD ?	; Check byte
EFD	51H	DB 7EH	; End-of-packet ; byte

DADDR This equals the address of the queried node.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID This field is equal to the appropriate connection ID indicating the session and associated packet.

SCONNID	This field is equal to the connection ID indicating the session of the requesting node.
XXXX	The X's represent a don't-care value.
DNAME	This is equal to the remote name.
SNAME	This is equal to the source name.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
SNCID	This field is equal to the session identifier for the source node.
SNODEID	This is equal to the local node name
CRC	This is the cyclic redundancy check for the packet.

Negative Name Query Response

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H 03H 05H	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H 09H 0BH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
N/A	11H	DB 30H	; Fixed value for ; this position of ; packet
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.

Negative Name Query Response

Field Name	Offset	Length	Comments
CONNID	13H	DW ?	; Field ID for ; packet
N/A	15H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
NACK- REAS	17H	DB 00H	; Reason why ; packet nacked.
N/A	18H	DB XXH	; Don't-care value ; for this ; position of ; packet
N/A	19H	DW 0101H	; Fixed value for ; this position of ; packet
N/A	21H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	23H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	25H	DW 10XXH	; Fixed value for ; this position of ; packet
N/A	27H	DW XX10H	; Fixed value for ; this position of ; packet

Negative Name Query Response

Field Name	Offset	Length	Comments
DNAME	29H	DB 16 DUP(?)	; ASCII Name for ; destination
SNAME	39H	DB 16 DUP(?)	; ASCII Name for ; source
DNCID	49H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	4BH	DD ?	; Check byte
EFD	4FH	DB 7EH	; End-of-packet ; byte

DADDR This equals the address of the queried node.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID This field is equal to the appropriate connection ID indicating the session and associated packet.

XXXX The X's represent a don't-care value.

NACKREAS	This is set to indicate the reason why a packet was nacked.
DNAME	This is equal to the remote name.
SNAME	This is equal to the source name.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Name Claim Response

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
N/A	11H	DB 30H	; Fixed value for ; this position of ; packet
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.

Name Claim Response

Field Name	Offset	Length	Comments
CONNID	13H	DW ?	; Field ID for ; packet
N/A	15H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
NACK- REAS	17H	DB ?	; Reason why ; packet nacked.
N/A	18H	DB XXH	; Don't-care value ; for this ; position of ; packet
N/A	19H	DW 0401H	; Fixed value for ; this position of ; packet
N/A	1BH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1DH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1FH	DW 10XXH	; Fixed value for ; this position of ; packet
N/A	21H	DW 0000H	; Fixed value for ; this position of ; packet

Name Claim Response

Field Name	Offset	Length	Comments
DNAME	23H	DB 16 DUP(?)	; ASCII Name for ; destination
DNCID	33H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	35H	DD ?	; Check byte
EFD	39H	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of remote name.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID This field is equal to the appropriate connection ID indicating the session and associated packet.

NACKREAS This is set to indicate the reason why a packet was nacked.

XXXX The X's represent a don't-care value.

DNAME	This is equal to the remote name.
SNAME	This is equal to the source name.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.
CRC	This is the cyclic redundancy check for the packet.

Name Query Cancel Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 5000H	; Fixed value for ; this position of ; packet
N/A	11H	DB A0H	; Fixed value for ; this position of ; packet
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets
CONNID	13H	DW ?	; Field ID for ; packet

Name Query Cancel Packet

Field Name	Offset	Length	Comments
N/A	15H	DW 0202H	; Fixed value for ; this position of ; packet
N/A	17H	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	19H	DW 0100H	; Fixed value for ; this position of ; packet
N/A	1BH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1DH	DW XXXXH	; Don't-care value ; for this ; position of ; packet
N/A	1FH	DW 10XXH	; Fixed value for ; this position of ; packet
N/A	21H	DW XX10H	; Fixed value for ; this position of ; packet
DNAME	23H	DW 16 DUP(?)	; ASCII Name for ; destination
SNAME	33H	DW 16 DUP(?)	; ASCII Name for ; source
PNCID	43H	DW ?	; (See definition)

Name Query Cancel Packet

Field Name	Offset	Length	Comments
DID	45H	DW ?	; Number that is ; incremented by ; 1 for each ; packet
SNCID	47H	DW ?	; Field ID for ; packet
DNODEID	49H	DW ?	; Low address
	4BH	DW ?	; Mid address
	4DH	DW ?	; Hi address
SNODEID	4FH	DW ?	; Low address
	51H	DW ?	; Mid address
	53H	DW ?	; Hi address
PNODEID	55H	DW ?	; Low address
	57H	DW ?	; Mid address
	59H	DW ?	; Hi address
CRC	5BH	DD ?	; Check byte
EFD	5FH	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of remote name.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID	Choose a new CONNID value on the first transmission or repeat the previous CONNID on retransmissions.
XXXX	The X's represent a don't-care value.
DNAME	This is equal to the remote name.
SNAME	This is equal to the source name.
PNCID	Low-order byte of PNCID equals low-order byte of SNCID.
DID	Use the next DID value.
SNCID	Use the next SNCID value on the first transmission or repeat the previous SNCID value on retransmission.
DNODEID	This is equal to f(remote name).
SNODEID	This is equal to the local node name
PNODEID	This is equal to SNODEID
CRC	This is the cyclic redundancy check for the packet.

Session Request

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 0040H	; Fixed value for ; this position of ; packet
POLL	11H	DB 0?H	; (00-07)H ; means no poll ; (08-0F)H ; means to send ; a return packet.

Session Request

Field Name	Offset	Length	Comments
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW ?	; Field ID for ; packet
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet
N/A	17H	DW 0001H	; Fixed value for ; this position of ; packet
PKTSIZE	19H	DW ?	; Packet size that ; can be accepted ; from the remote ; node.
N/A	1BH	DW 0000H	; Fixed value for ; this position of ; packet
N/A	1DH	DW 1010H	; Fixed value for ; this position of ; packet

Session Request

Field Name	Offset	Length	Comments
SNAME	1FH	DB 16 DUP(?)	; ASCII Name for ; source
DNAME	2FH	DB 16 DUP(?)	; ASCII Name for ; destination
DNCID	3FH	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	41H	DD ?	; Check byte
EFD	45H	DB 7EH	; End-of-packet ; byte

DADDR This equals the address of the next node of a network connection. (Possibly the destination node.)

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

POLL When this byte is set to (08-0F)H, the receiver is requested to send a return packet. When this byte is set to (00-07)H, indicates no poll.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID	This field is equal to the appropriate connection ID indicating the session and associated packet.
SEQ	This field is equal to the session packet sequence number for this packet.
ACK	This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.
PKTSIZE	Set this size to the maximum that an initiating session entity will accept.
SNAME	This is equal to the source name.
DNAME	This is equal to the destination name.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Session Accept

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H 03H 05H	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H 09H 0BH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 0040H	; Fixed value for ; this position of ; packet
POLL	11H	DB 0?H	; (00-07)H ; means no poll ; (08-0F)H ; means to send ; a return packet.

Session Accept

Field Name	Offset	Length	Comments
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW DUP(0)	; Field ID for ; packet
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet
N/A	17H	DW 0002H	; Fixed value for ; this position of ; packet
PKTSIZE	19H	DW ?	; Packet size that ; can be accepted ; from the remote ; node.
DNCID	1BH	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	1DH	DD DUP(0)	; Check byte

Session Accept

Field Name	Offset	Length	Comments
EFD	22H	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of the responding node.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

POLL When this byte is set to (08-0F)H, the receiver is requested to send a return packet. When this byte is set to (00-07)H, indicates no poll.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID This field is equal to the appropriate connection ID indicating the session and associated packet.

SEQ This field is equal to the session packet sequence number for this packet.

ACK This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.

PKTSIZE Set this size to the maximum that an initiating session entity will accept.

DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Session Reject

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H 03H 05H	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H 09H 0BH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
POLL	11H	DB 0?H	; (00-07)H ; means no poll ; (08-0F)H ; means to send ; a return packet.

Session Reject

Field Name	Offset	Length	Comments
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW ?	; Field ID for ; packet
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet
N/A	17H	DB 03H	; Fixed value for ; this position of ; packet
RVAL	18H	DB ?	; Value to indicate ; why session was ; rejected.
DNCID	19H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	1BH	DD ?	; Check byte

Session Reject

Field Name	Offset	Length	Comments
EFD	1FH	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of the responding node.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

POLL When this byte is set to (08-0F)H, the receiver is requested to send a return packet. When this byte is set to (00-07)H, indicates no poll.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID This field is equal to the appropriate connection ID indicating the session and associated packet.

SEQ This field is equal to the session packet sequence number for this packet.

ACK This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.

RVAL This field indicates the reason code for the session reject.

DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Ack Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H 03H 05H	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H 09H 0BH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
POLL	11H	DB 4?H	; (40-47)H ; means no poll. ; (48-4F)H ; means to send ; a return packet.

Ack Packet

Field Name	Offset	Length	Comments
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW ?	; Field ID for ; packet
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet.
N/A	17H	DB XXH	; Don't-care value ; for this ; position of ; packet
DNCID	18H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	1AH	DD ?	; Check byte
EFD	1EH	DB 7EH	; End-of-packet ; byte

DADDR	This equals the group address of the responding node.
SADDR	This equals the address of the transmitting node.
DLEN	This equals the length of the Link level packet data field.
POLL	When this byte is set to (48-4F)H, the receiver is requested to send a return packet. When this byte is set to (40-47)H, indicates no poll.
WIN	This equals the current value indicating the number of packets the sender is willing to accept.
CONNID	This field is equal to the appropriate connection ID indicating the session and associated packet.
SEQ	This field is equal to the session packet sequence number for this packet.
ACK	This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Nack Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
POLL	11H	DB 5?H	; (50-57)H ; means no poll. ; (58-5F)H ; means to send ; a return packet.
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW ?	; Field ID for ; packet

Nack Packet

Field Name	Offset	Length	Comments
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet.
NACK-REAS	17H	DB ?	; Reason why ; packet not ; received.
DNCID	18H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify ; datagram
CRC	1AH	DD ?	; Check byte
EFD	1EH	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of the responding node.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

POLL	When this byte is set to (58-5F)H, the receiver is requested to send a return packet. When this byte is set to (50-57)H, indicates no poll.
WIN	This equals the current value indicating the number of packets the sender is willing to accept.
CONNID	This field is equal to the appropriate connection ID indicating the session and associated packet.
SEQ	This field is equal to the session packet sequence number for this packet.
ACK	This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.
NACKREAS	This field is equal to the code indicating the reason for the nacked packet.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Close Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H 03H 05H	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H 09H 0BH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
POLL	11H	DB 6?H	; (60-67)H ; means no poll. ; (68-6F)H ; means to send ; a return packet.

Close Packet

Field Name	Offset	Length	Comments
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW ?	; Field ID for ; packet
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet.
N/A	17H	DB XXH	; Fixed value for ; this position of ; packet
DNCID	18H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify ; datagram
CRC	1AH	DD ?	; Check byte
EFD	1EH	DB 7EH	; End-of-packet ; byte

DADDR	This equals the address of the next node of a network connection. (Possibly the destination node.)
SADDR	This equals the address of the transmitting node.
DLEN	This equals the length of the Link level packet data field.
POLL	When this byte is set to (68-6F)H, the receiver is requested to send a return packet. When this byte is set to (60-67)H, indicates no poll.
WIN	This equals the current value indicating the number of packets the sender is willing to accept.
CONNID	This field is equal to the appropriate connection ID indicating the session and associated packet.
SEQ	This field is equal to the session packet sequence number for this packet.
ACK	This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Closed Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H 03H 05H	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H 09H 0BH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
N/A	11H	DB 70H	; Fixed value for ; this position of ; packet
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value

Closed Packet

Field Name	Offset	Length	Comments
CONNID	13H	DW ?	; Field ID for ; packet
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet.
CLREAS	17H	DB ?	; Reason why ; connection ; closed.
DNCID	18H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	1AH	DD ?	; Check byte
EFD	1EH	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of remote name.

SADDR This equals the address of the transmitting node.

DLEN	This equals the length of the Link level packet data field.
WIN	This equals the current value indicating the number of packets the sender is willing to accept.
CONNID	This field is equal to the appropriate connection ID indicating the session and associated packet.
SEQ	This field is equal to the session packet sequence number for this packet.
ACK	This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.
CLREAS	This field is equal to the code indicating the reason for the connection being closed.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Data Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
POLL	11H	DB 0?H	; (00-07)H ; means no poll. ; (08-0F)H ; means to send ; a return packet.

Data Packet

Field Name	Offset	Length	Comments
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW ?	; Field ID for ; packet
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet.
EOM	17H	DB ?0H	; End-of-mes- ; sage indicator. ; (80-F0)H equals ; end-of-mes- ; sage.
DATA	18H	DB ?? DUP(?)	; Variable length ; field.
DNCID	XXH	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	XXH	DD ?	; Check byte

Data Packet

Field Name	Offset	Length	Comments
EFD	XXH	DB 7EH	; End-of-packet ; byte

DADDR This equals the next node of a network connection. (Possibly the destination node.)

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

POLL When this byte is set to (08-0F)H, the receiver is requested to send a return packet. When this byte is set to (00-07)H, indicates no poll.

WIN This equals the current value indicating the number of packets the sender is willing to accept.

CONNID This field is equal to the appropriate connection ID indicating the session and associated packet.

SEQ This field is equal to the session packet sequence number for this packet.

ACK This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.

EOM When this bit is set to 1, packets contains the end of the user's logical message. When this bit is set to 0, packet does not contain the end of the message.

DATA This is a variable field containing user data.

DNCID This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)

CRC This is the cyclic redundancy check for the packet.

Datagram Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 5100H	; Fixed value for ; this position of ; packet
N/A	11H	DW 0100H	; Fixed value for ; this position of ; packet
N/A	13H	DW 0001H	; Fixed value for ; this position of ; packet
N/A	15H	DW 1010H	; Fixed value for ; this position of ; packet

Datagram Packet

Field Name	Offset	Length	Comments
N/A	17H	DW 0000H	; Fixed value for ; this position of ; packet
SNAME	19H	DB 16 DUP(?)	; Name of source ; node in ASCII
DNAME	29H	DB 16 DUP(?)	; ASCII Name for ; destination
DATA	39H	DB ?? DUP(?)	; Variable length ; field.
PNCID	XXH	DW FFFEH	; Equal to ; SNCID
DID	XXH	DW ?	; Number that is ; incremented by ; 1 for each ; packet
SNCID	XXH	DW FFFEH	; Field ID for ; packet
DNODEID	XXH XXH XXH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
SNODEID	XXH XXH XXH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
PNODEID	XXH XXH XXH	DW ? DW ? DW ?	; Low address ; Mid address ; Hi address
CRC	XXH	DD ?	; Check byte

Datagram Packet

Field Name	Offset	Length	Comments
EFD	XXH	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of remote name.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

SNAME This field is equal to the source name.

DNAME This field is equal to a destination name or a " *broadcast" for a broadcast all.

DATA This is a variable field containing user data.

PNCID This field is equal to the datagram identifier for the previous node.

DID This field is equal to the next DID value.

SNCID This field is equal to the datagram identifier for the source node.

DNODEID This is equal to the destination node name.

SNODEID This is equal to the local node name.

PNODEID This field is equal to the SNODEID field.

CRC	This is the cyclic redundancy check for the packet.

Get Status Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 5100H	; Fixed value for ; this position of ; packet
N/A	11H	DW 0300H	; Fixed value for ; this position of ; packet
N/A	13H	DW 0003H	; Fixed value for ; this position of ; packet
CMDRESP	15H	DB ?2H	; (02-72)H = ; command. ; (82-F2)H = ; response.

Get Status Packet

Field Name	Offset	Length	Comments
N/A	16H	DB XXH	; Fixed value for ; this position of ; packet
N/A	17H	DW 8001H	; Fixed value for ; this position of ; packet
N/A	19H	DW 8001H	; Fixed value for ; this position of ; packet
TRANSID	1BH	DW ?	; ID for status ; and status re- ; sponse packet
N/A	1DH	DB 10H	; Fixed value for ; this position of ; packet
DNAME	1EH	DB 16 DUP(?)	; ASCII Name for ; destination
N/A	2EH	DB 00H	; Fixed value for ; this position of ; packet
N/A	2FH	DW 0000H	; Fixed value for ; this position of ; packet
PNCID	31H	DW FFFEH	; Equal to ; SNCID
DID	33H	DW ?	; Number that is ; incremented by ; 1 for each ; packet

Get Status Packet

Field Name	Offset	Length	Comments
SNCID	35H	DW FFFEH	; Field ID for ; packet
DNODEID	37H	DW ?	; Low address
	39H	DW ?	; Mid address
	3BH	DW ?	; Hi address
SNODEID	3DH	DW ?	; Low address
	3FH	DW ?	; Mid address
	41H	DW ?	; Hi address
PNODEID	43H	DW ?	; Low address
	45H	DW ?	; Mid address
	47H	DW ?	; Hi address
CRC	49H	DD ?	; Check byte
EFD	4DH	DB 7EH	; End-of-packet ; byte

DADDR This equals the group address of remote permanent node name.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

CMDRESP When this bit is set to 1, the packet is a response. When set to 0, the packet is a command.

TRANSID This field contains the transaction identification number.

DNAME	This field is equal to a destination name.
PNCID	This field is equal to the datagram identifier for the previous node.
DID	Use the next DID value.
SNCID	This field is equal to the datagram identifier for the source node.
DNODEID	This is equal to the destination node name.
SNODEID	This is equal to the local node name.
PNODEID	This field is equal to the SNODEID field.
CRC	This is the cyclic redundancy check for the packet.

Status Response Packet

Field Name	Offset	Length
STD	00H	DB 7EH
DADDR	01H	DW ?
	03H	DW ?
	05H	DW ?
SADDR	07H	DW ?
	09H	DW ?
	0BH	DW ?
DLEN	0DH	DW ?
N/A	0FH	DW 5100H
N/A	11H	DW 0300H
N/A	13H	DW 0003H
N/A	15H	DB 82H
N/A	16H	DB 00H
N/A	17H	DW 8001H
N/A	19H	DW 8001H
TRANSID	1BH	DW ?
N/A	1DH	DB 10H
DNAME	1EH	DW 16 DUP(?)
N/A	2EH	DB 00H

Status Response Packet

Field Name	Offset	Length
STATLEN	2FH	DW ?
DNODEID	31H	DW ?
	33H	DW ?
	35H	DW ?
JUMPSTAT	37H	DB ?
SELFTEST	38H	DB 00H
SWVERSION	39H	DW ?
REPORTPD	3BH	DW ?
CRCERRS	3DH	DW ?
ALGNERRS	3FH	DW ?
COLLISIONS	41H	DW ?
XMITABRTS	43H	DW 0000H
XMITMSGS	45H	DD ?
RECVMSGS	49H	DD ?
REXMITCNT	4DH	DW ?
NORESOURCES	4FH	DW ?
N/A	51H	DW XXXXH
N/A	53H	DW XXXXH
N/A	55H	DW XXXXH
N/A	57H	DW XXXXH

Status Response Packet

Field Name	Offset	Length
N/A	59H	DW XXXXH
N/A	5BH	DW XXXXH
N/A	5DH	DW XXXXH
N/A	5FH	DW XXXXH
N/A	61H	DW XXXXH
N/A	63H	DW XXXXH
N/A	65H	DW XXXXH
N/A	67H	DW XXXXH
N/A	69H	DW XXXXH
NRALIAS	6BH	DB ?
ALIASNAME	6CH	DB 16 DUP(?)
ALIASNR	7CH	DB ?
ALIASTAT	7DH	DB ?
o	XXH	o
o	XXH	o
o	XXH	o
ALIASNAME	XXH	DB 16 DUP(?)
ALIASNR	XXH	DB ?
ALIASTAT	XXH	DB ?

Status Response Packet

Field Name	Offset	Length
PNCID	XXH	DW FFFEH
DID	XXH	DW ?
SNCID	XXH	DW FFFEH
DNODEID	XXH	DW ?
	XXH	DW ?
	XXH	DW ?
SNODEID	XXH	DW ?
	XXH	DW ?
	XXH	DW ?
PNODEID	XXH	DW ?
	XXH	DW ?
	XXH	DW ?
CRC	XXH	DD ?
EFD	XXH	DB 7EH

DADDR This equals the group address of remote node (g(requesting node)).

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

TRANSID This is the transaction identification number

DNAME This is equal to the destination name.

STATLEN	This is equal to the offset value of NRALIAS minus 12H.
DNODEID	This is equal to the responding node ID.
JUMPSTAT	Two high order bits, when set to 1, indicates that jumpers W2 and W1 are in place. The highest order bit corresponds to jumper W2.
SELFTEST	This byte is the result code of the selftest of the node.
SWVERSION	These bytes are used to indicate the software version currently used.
REPORTPD	This is indicated in minutes for the time since the last hardware reset.
CRCERRS	Number of received packets with failed CRC checks.
ALGNERRS	Number of received packets out of alignment.
COLLISIONS	Number of collisions encountered on transmission packets.
XMITABRTS	Number of transmitted packets that aborted.
XMITMSGS	Number of packets successfully transmitted by the Link level.
RECVMSGS	Number of successfully received packets.
REXMITCNT	Count of packets retransmitted.

NORESOURCES	Count of receive failures due to lack of resources.
NRALIAS	Number of alias names to follow in the packet. The next three fields, ALIASNAME, ALIASTAT, and ALIASNR, are repeated in sequence with the number given in the NRALIAS field. This fields offset can be calculated by using the STATLEN field offset and adding 12H to it.
ALIASNAME	Alias name for 16 bytes.
ALIASNR	Number assigned to the alias name.
ALIASTAT	This byte indicates the status of the name specified in the ALIASNAME field. See "* Local name table—16 entries of 18 bytes each" on page 2-33 for the values in this field.
PNCID	This field is equal to the datagram identifier for the previous node.
DID	Use the next DID value.
SNCID	This field is equal to the datagram identifier for the source node.
SNODEID	This is equal to the local node name.
PNODEID	This field is equal to the SNODEID field.

CRC	This is the cyclic redundancy check for the packet.

Abort Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address ; This is the ; next node ; address ; (initiator's ; address)
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 4000H	; Fixed value for ; this position of ; packet
POLL	11H	DB 9?H	; (90-97)H ; means no poll. ; (98-9F)H ; means to send ; a return packet.

Abort Packet

Field Name	Offset	Length	Comments
WIN	12H	DB 0?H	; Low-order 4-bits ; define number of ; packets. High- ; order 4-bits are ; fixed value.
CONNID	13H	DW ?	; Field ID for ; packet
SEQ	15H	DB ?	; Session packet ; sequence ; number
ACK	16H	DB ?	; Number that in- ; cludes number ; +1 modulo 256 ; of last ; correctly ; received packet.
N/A	17H	DB XXH	; Don't-care value ; for this ; position of ; packet
DNCID	18H	DW ?	; Number used to ; determine ; the session for ; packet or to ; identify data- ; gram
CRC	1AH	DD ?	; Check byte
EFD	1EH	DB 7EH	; End-of-packet ; byte

DADDR	This equals the address of the next node of a network connection. (Possibly the destination node.)
SADDR	This equals the address of the transmitting node.
DLEN	This equals the length of the Link level packet data field.
POLL	When this byte is set to (98-9F)H, the receiver is requested to send a return packet. When this byte is set to (90-97)H, indicates no poll.
WIN	This equals the current value indicating the number of packets the sender is willing to accept.
CONNID	This field is equal to the appropriate connection ID indicating the session and associated packet.
SEQ	This field is equal to the session packet sequence number for this packet.
ACK	This field is equal to the acknowledgment sequence number. This number indicates the next expected sequence number to be received.
DNCID	This field is equal to the session identifier for the destination node. This field is used with CONNID to specify the destination session.)
CRC	This is the cyclic redundancy check for the packet.

Self-Test Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW F000H	; Fixed value for ; this position of ; packet
CRC	11H	DD ?	; Check byte
EFD	15H	DB 7EH	; End-of-packet ; byte

DADDR This equals the address of the transmitting node.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field.

CRC	This is the cyclic redundancy check for the packet.

Ident Packet

Field Name	Offset	Length	Comments
STD	00H	DB 7EH	; Start delimiter ; flag byte
DADDR	01H	DW ?	; Low address
	03H	DW ?	; Mid address
	05H	DW ?	; Hi address
SADDR	07H	DW ?	; Low address
	09H	DW ?	; Mid address
	0BH	DW ?	; Hi address
DLEN	0DH	DW ?	; Length of Link ; level packet
N/A	0FH	DW 5100H	; Fixed value for ; this position of ; packet
N/A	11H	DW 0300H	; Fixed value for ; this position of ; packet
N/A	13H	DW 0003H	; Fixed value for ; this position of ; packet
CMDRESP	15H	DB ?1H	; 41H = command ; C1H = response
N/A	16H	DB 00H	; Fixed value for ; this position of ; packet

Ident Packet

Field Name	Offset	Length	Comments
N/A	17H	DW 8001H	; Fixed value for ; this position of ; packet
N/A	19H	DW 8001H	; Fixed value for ; this position of ; packet
TRANSID	1BH	DW 0000H	; ID for status ; and status re- ; sponse packet
N/A	1DH	DB 10H	; Fixed value for ; this position of ; packet
DNAME	1EH	DB 16 DUP(?)	; ASCII Name for ; destination
N/A	2EH	DB 00H	; Fixed value for ; this position of ; packet
NODEID- MASK	2FH 31H 33H	DW ? DW ? DW ?	; Mask value ; ;
NODEID- MATCH	35H 37H 39H	DW ? DW ? DW ?	; Match value ; ;
PNCID	3BH	DW FFFEH	; Equal to ; SNCID
DID	3DH	DW ?	; Number that is ; incremented by ; 1 for each ; packet

Ident Packet

Field Name	Offset	Length	Comments
SNCID	3FH	DW FFFEH	; Field ID for ; packet
DNODEID	41H	DW ?	; Low address
	43H	DW ?	; Mid address
	45H	DW ?	; Hi address
SNODEID	47H	DW ?	; Low address
	49H	DW ?	; Mid address
	4BH	DW ?	; Hi address
PNODEID	4DH	DW ?	; Low address
	4FH	DW ?	; Mid address
	51H	DW ?	; Hi address
CRC	53H	DD ?	; Check byte
EFD	57H	DB 7EH	; End-of-packet ; byte

DADDR This is a broadcast address for requests. For a response, this is the group address of the initiators permanent node name.

SADDR This equals the address of the transmitting node.

DLEN This equals the length of the Link level packet data field. node.

CMDRESP When this byte is set to C1H, the packet is a response. When set to 41H, the packet is a command.

TRANSID	This field contains the transaction identification number.
DNAME	This field is equal to a destination name.
NODEIDMASK	Mask value to use against destination node name.
NODEIDMATCH	Match value to request a response.
PNCID	This field is equal to FFFEH.
DID	Use the next DID value.
SNCID	This field is equal to FFFEH.
DNODEID	This is equal to the destination node name.
SNODEID	This is equal to the local node name.
PNODEID	This field is equal to the SNODEID field.
CRC	This is the cyclic redundancy check for the packet.

Protocol Interactions

This section provides pseudo-code descriptions of the interactions of various layers of the communications protocols.

Session to Transport Layer Interactions

This section is a description of the transfer of data between the session and transport layers of the protocol architecture. Once a session has been requested by the host, the session layer software calls the transport layer software to establish a reliable connection between the source and destination computers.

```
PROCEDURE open__RSP__connection
   {sourceServiceID, destServiceID,
   networkAddr};
BEGIN
   initialize retry count = maxRetransmissions;
REPEAT
   decrement retry count;
   set open request timer;
   make Send Establish call to PTP;
   {sends open__request packet to PTP}
   REPEAT
      wait for response from remote node;
   UNTIL open request timer = 0 or
      response received;
      UNTIL retry count = 0 or response received;
   IF openAck received THEN
      {validate that packet was expected}
   IF openAck unexpected THEN ignore it
   ELSE
     store returned destination connection ID;
     return successful open__RSP__connection call
     with proper connection ID to use;
   ELSE IF openNack received THEN
      return failed open__RSP__connection call
```

```
    with reason code;
ELSE
  return failed open__RSP__connection call with
  no response status to user;
  END;
```

Network Layer Interaction

This section describes the interactions between the network layer protocol entities in two adapter cards.

```
PROCEDURE send__PTP__packet (NCID, bufAddr, bufLen)
  {requests that the specified buffer be sent to the
  specified NCID}
BEGIN
  check for parameter error;
  check status of connection
IF NCID is not valid THEN
  return call as failed with invalid NCID code;
  ELSE IF no connection exists THEN
    return call as failed with connection not;
    established code;
    ELSE IF connection requested
    by remote source THEN
      format and send route__completion packet;
      send__LAP__frame (bufAddr, bufLen, destNodeID)
  ELSE
    format and send connection__data packet;
    send__LAP__frame (bufAddr, bufLen, destNodeID)
END;
```

```
PROCEDURE send__LAP__frame (bufAddr,
      bufLen, destNodeID)
  {send a buffer of data as the data field of a frame
  to the indicated destination node name}
BEGIN
  assemble frame
  set destination link level
  address = destNodeID;
  set data field = data buffer;
  generate CRC word for contents of buffer;
  REPEAT
    monitor receiver's carrier sense signal
```

```
    UNTIL no carrier detected for
      interframe__wait__time bits;
      transmit frame;
      monitor channel for at least
      collision__byte__count bytes following preamble;
      IF collision detected or carrier lost THEN
        jam channel;
        increment retransmission counter;
      IF retransmission counter = max THEN
        do not reschedule frame for
        retransmission
      ELSE
        reschedule frame for later
        transmission;
     ELSE
      continue transmission to end of frame;
END;
```

```
PROCEDURE receive LAP frame
  {LAP takes the bits presented to it from the physical
  layer and transfers a valid frame to PTP}
BEGIN
  LAP allocates buffer for next incoming frame;
  LAP receives frame from physical layer;
    check DLAddr;
IF frame is not for this LANA THEN
  ignore frame;
  reallocate buffer
ELSE
  check CRC;
  IF CRC of frame is ≠ to calculated CRC THEN
    ignore frame;
  ELSE
    received__frame (bufAddr, bufLen,
    rcvdChanID,
    reception__type {, groupAddr})
END;
```

```
PROCEDURE receive LAP frame
   {LAP takes the bits presented to it from the physical
   layer and transfers a valid frame to PTP}
BEGIN
   LAP allocates buffer for next incoming frame;
   LAP receives frame from physical layer;
       check DLAddr;
IF frame is not for this LANA THEN
   ignore frame;
   reallocate buffer
ELSE
   check CRC;
   IF CRC of frame is ≠ to calculated CRC THEN
       ignore frame
   ELSE
       received__frame (bufAddr, bufLen,
       rcvdChanID,
       reception__type {, groupAddr})
END;
```

```
PROCEDURE received__frame (bufAddr, bufLen,
        rcvdChanID, receptionType {,
        groupAddr})
   {send data buffer from link to network layer and
   interpret network layer header and trailer. Pass data
   on to transport layer}
BEGIN
   check packet type
IF type is route tear down THEN
   notify network entity to erase route from
   memory;
ELSE IF type is connection data THEN
   set NCID to nextLID field in packet's trailer;
   received__call (PID, bufAddr, bufLen)
ELSE IF type is route completion THEN
   set nextLID = received packet's trailer's
```

```
    prevID field;
    set nextNodeID to received packet's prevNodeID
    field;
    received__call (PID, bufAddr, bufLen)
    {connection established}
ELSE IF type is discovery THEN
    IF destNodeID field does not match adapter's
    permanent node name or any currently enabled group
    address THEN
        ignore packet;
    ELSE
        validate packet using discovery table
    IF packet is a duplicate THEN
        ignore packet
    ELSE IF packet is a datagram type THEN
        received__call (PID=1, bufAddr, bufLen)
    ELSE
        {packet is route establishing type}
        allocate entry in network connection
            table;
        set NCID to index of this entry;
        set entry's nextLID to packet's
            prevLID;
        set entry's nextNodeID to source
            permanent node name;
        received__call (PID=0, bufAddr, bufLen)
END;
```

Packet Reception Procedures

This section describes the receipt of packets by the transport and session layer entities from the network layer entity. At this point, the packet has been received and determined to be for this node.

```
PROCEDURE received__call (PID, bufAddr, bufLen)
  BEGIN
  IF protocolID = 1 {datagram} THEN
CASE packet type OF

user datagram:
    search for sender's name in remote name
    table;
    IF sender's name is not found THEN
      set unknown__remote__name indicator
      {unable to determine sender's name}
    IF datagram receive specific is
      specified THEN
        check recipient alias number in
        datagram message;
    IF recipient alias number in datagram
      message is same as local alias number
      THEN
      set receive__completed indicator
         {datagram receive to specific
         alias satisfied}
    ELSE
      reset receive__completed indicator
        {datagram receive not satisfied,
        continue to wait}
    ELSE
      set receive__completed indicator
        {datagram receive to any alias
        satisfied}
      transfer datagram data to user buffer;
      return actual length of transfer to user
      return local alias__no to user;
   IF unknown__remote__name indicator
      is set THEN
```

```
        return unknown__remote__name status to
        user;
           {datagram received, unable to
           determine sender's name}
     ELSE
        return sender's name to user
        IF size of user buffer is smaller than
        received datagram THEN
        return
        message incomplete status to user;
            {datagram received, unable to transfer
            entire message}
     ELSE
       return command__completed status to user
          {datagram received}

name query:
     IF name is stored in local name table THEN
       send name__query__response packet to
       sender;
     ELSE
       ignore packet;

name query response:
    check total number of query__response
       packets received;
    IF one query__response packet is received
       THEN
       enter information to remote name
       table;
       send datagram to remote node;
       return command__completed status to
       user;
    ELSE
       send name__conflict packets to
       nodes that responded with
       query__response packet;
       return unknown__remote__name status to
       user;

name claim:
    IF name is in local name table THEN
      send name__claim__response packet to
```

```
        sender;
    ELSE
        ignore packet;

name claim response:
    return alias__name__in__use status to user;

status request:
    IF alias name is * THEN
    IF local alias table is empty THEN
        return no__valid__aliases status
        to user;
    ELSE
        REPEAT
        search for alias name in local alias
        table;
        get session status is session table;
        return session status to user
        UNTIL all alias names are found;
            return actual length of session
            status;
        IF size of user buffer is smaller than
        session status THEN
            return
            message incomplete status to user;
        ELSE
            return command__completed status to
            user;
    ELSE
        search for alias name in local alias
        table;
    IF alias name is found THEN
        get session status in session table;
        return session status to user;
        return actual length of session
        status;
    IF size of user buffer is smaller than
        session status THEN
        return
        message incomplete status to user;
    ELSE
        return command__completed status to
        user;
```

```
    ELSE
        return illegal__alias__name status to
        user;

status response:
    IF status__response packet is received
        within the timeout interval THEN
        get configuration parameters and
        status of responding adapter;
        return configuration parameters and
        status;
        return actual length of configuration
        parameter and status;
        IF size of user buffer is smaller than
        the configuration parameter and status
        THEN
            return message incomplete status
            to user;
    ELSE
        return command__completed status to
        user;
    ELSE
        return command__timed__out status to
        user;
END CASE;
```

```
IF protocolID = 0 {session} THEN
   CASE packet type OF

open request:
   IF open__request packet is a duplicate THEN
   return appropriate response again (open
   ack or open nack) to sender
   ELSE
      IF specified service exists and can
      provide resources THEN
         send open ack to sender;
         notify RSP user with
         new__connection__from__remote call
   ELSE
      send open nack with reason;

open ack:
    validate that open ack packet was expected
    (open request was issued with the given
    source connection ID);
    IF open ack was not expected THEN
        ignore packet
    ELSE
        return user's call successfully with
        connection ID;

open nack:
    return error indication to user's open call;
    return error reason;

ack:
    remove the acknowledged packet form the
    retransmit queue;
    update the last acknowledged variable;

nack:
    remove all acknowledged packets form the
    retransmit queue;
    retransmit all unacknowledged packets (up
    to the current set__window__size value);
```

```
close:
    IF close packet received from close
    initiator THEN
       send close packet to close initiator;
    ELSE
       {close packet received form close
       non-initiator}
       acknowledge receipt of close packet;
       send closed packet;
       notify user that connection is closed;

closed:
    notify user that connection is closed;

session request:
    REPEAT
    IF passive open specified is specified THEN
         check source of session__request packet;
         IF source of session__request
         packet is same as remote name THEN
            set session__request__completed
            indicator;
            {passive open specified satisfied}
    ELSE
        reset session__request__completed
        indicator;
        {passive open not satisfied,
        continue to wait}
   ELSE
        set session__request__completed indicator
        {passive open any satisfied}
        UNTIL session__request__completed indicator
            is set;
            send session__accept packet to source;
            set session__established indicator in
            session table;
            return source of session__request packet to
            user;
            return local__session__no to user;
            return command__completed status to user
            {session established}
```

```
session accept:
    set session__established indicator in
    session table;
    return local__session__no to user;
    return command__completed status to user
    {session established}

session reject:
    return session__open__rejected status to user;

session data:
    IF session data is received within the
    timeout interval for session received THEN
       transfer session data to user buffer;
       return actual length of transfer
       to user;
       IF size of user buffer is smaller than
       received session data THEN
          return
          message incomplete status to user;
    ELSE
        return command__completed status to
        user;
        ELSE
        return command__timed__out status to user
        {session data received};
END CASE;
END
```

Appendix D. Adapter BIOS

```
                              TITLE LANA ROS POD


                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                              ;                                                                       ;
                              ;       NETWORK ADAPTER DIAGNOSTICS                                     ;
                              ;                                                                       ;
                              ;       THIS CODE IS ACCESSED AT POWER ON AND DURING SOFT RESETS. THE   ;
                              ;       ADAPTER IS TESTED FOR PROPER OPERATION. IF A ERROR IS DETECTED  ;
                              ;       A MESSAGE IS DISPLAYED TO INDICATE THE TYPE OF FAILURE.         ;
                              ;                                                                       ;
                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                              ;                                                                       ;
                              ;               LANA ADDRESSING         CARD 1          CARD 2          ;
                              ;       (SR)    STATUS REGISTER          360             368            ;
                              ;       (PR)    PARAMETER REGISTER       361             369            ;
                              ;       (DR)    DATA REGISTER            362             36A            ;
                              ;       (HIR)   HOST INTERFACE REG       363             36B            ;
                              ;       BL      CONTAINS DELTA           00              08             ;
                              ;                                                                       ;
                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                              ;               INTERRUP VECTORS                                        ;
                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


 0000                         DATA    SEGMENT AT 0
 0028                                 ORG     0AH*4                   ;INT 2 VECTOR ADDRESS
 0028      01 [               INTR_2          DW 1 DUP (?)            ;CONTAINS OFFSET OF INT 2 HANDLER
                  ????
                        ]

 002A      01 [                               DW  1 DUP (?)           ;CONTAINS CS FOR 2
                  ????
                        ]

 002C                                 ORG     0BH*4                   ;INT 3 VECTOR ADDRESS

 002C      01 [               INTR_3           DW  1 DUP (?)                  ;CONTAINS OFFSET OFF INT 3 HANDLER
                  ????
                        ]

 002E      01 [                               DW   1 DUP (?)                 ;CONTAINS CS FOR INT 3
                  ????
                        ]

 004C                                 ORG     13H*4                   ;HARD FILE
 004C      01 [               HARD_FILE_BIOS  DW   1 DUP (?)                ;
                  ????
                        ]

 004E      01 [                               DW   1 DUP (?)                ;
                  ????
                        ]

 0060                                 ORG     18H*4                   ;
 0060      01 [               INT_18          DW   1 DUP (?)                ;IPL  vector address
                  ????
                        ]

 0062      01 [                               DW   1 DUP (?)                ;
                  ????
                        ]

 0070                                 ORG     1CH*4                   ;INT 1C TIMER TICK
 0070                         TICK_INT        LABEL   WORD            ;

 0170                                 ORG     05CH*4                  ;LANA BIOS SOFT INT
 0170      01 [               LANA_BIOS_INT   DW   1 DUP (?)                  ;CONTAINS OFFSET OFF 5C
                  ????
                        ]

 0172      01 [                               DW   1 DUP (?)                  ;CONTAINS CS
                  ????
                        ]


                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                              ;               RAM WORK AREA                                           ;
                              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

 1000                                 ORG     1000H                   ;
 1000                         RAM_AREA        LABEL   BYTE            ;
 1000      01 [               INT_OCCUR?      DB   1 DUP (?)                ;80=INT
                 ??
                      ]

 1001      01 [               INTS_ACTIVE     DB   1 DUP (?)                ;CONTAINS THE INT LEVELS FOUND ACTIVE
                 ??
                      ]


                                                                      ;08=INT 3  04=INT 2
 1002      01 [               COUNT           DB   1 DUP (?)                ;USE FOR DEBUG
                 ??
                      ]

 1003      01 [               DATA_XFER       DB   1 DUP (?)                ;DMA WRITES HERE
                 ??
                      ]

 1004      01 [                               DB   1 DUP (?)                ;2ND BYTE DMA
                 ??
                      ]

 1005      01 [               FAIL_FLAG       DB   1 DUP (?)                ;80=HARDWARE FAILURE
                 ??
                      ]

 1006      01 [               RF_TEST         DB   1 DUP (?)                ;80=HOT RF TEST IN PROCESS
                 ??
                      ]

 1007      01 [               SAVE_MASK       DB   1 DUP (?)                ;IMR MASK FOR 1ST INT CHIP
                 ??
                      ]

 1008      01 [               SAVE_MASKA      DB   1 DUP (?)                ;IMR MASK FOR 2ND INT CHIP
                 ??
                      ]

 1009      01 [               SAVE_TICK_INT   DW   1 DUP (?)                ;OFFSET OF TICK HANDLER IN BIOS
                  ????
                        ]
```

```
100B      01 [                         DW    1 DUP (?)           ;CODE SEG OF BIOS
               ????
                   ]

100D      01 [           TICKS         DW    1 DUP (?)           ;COUNTER FOR TIMER TICKS
               ????
                   ]

100F      01 [           T_O_FLAG      DB    1 DUP (?)           ;GROSS TIME OUT FLAG=80=FAIL
               ??
                 ]

1010      01 [           PC_ID         DB    1 DUP (?)           ;ID BYTE
               ??
                 ]


                         ;--INT 2 = BIT 2,--INT 3 = BIT 3
1011      01 [           TEMP_INT      DB    1 DUP (?)           ;HOLDS CURRENT ACTIVE INT LEVEL
               ??
                 ]

1012      01 [           LANA_0_INT    DB    1 DUP (?)           ;CONTAINS ACTIVE INT LEVEL FOR LANA 0
               ??
                 ]

1013      01 [           LANA_1_INT    DB    1 DUP (?)           ;CONTAINS ACTIVE INT LEVEL FOR LANA 1
               ??
                 ]

1014      01 [           LANA_1_ACTIVE DB    1 DUP (?)           ;0=LANA NOT ACTIVE
               ??
                 ]

1015      01 [           NO_SYNC       DB    1 DUP (?)           ;80 = NOT IN SYNC FAILURE
               ??
                 ]



                         ;SAVE AREA FOR POST VECTORS

1016      01 [           SAVE_INT2     DW    1 DUP (?)           ;OFFSET INT 2
               ????
                   ]

1018      01 [                         DW    1 DUP (?)           ;SEGMENT
               ????
                   ]

101A      01 [           SAVE_INT3     DW    1 DUP (?)           ;OFFSET INT 3
               ????
                   ]

101C      01 [                         DW    1 DUP (?)           ;SEGMENT
               ????
                   ]


101E .    01 [           DMA_DATA      DB    1 DUP (?)           ;MOV  DMA DATA HERE
               ??
                 ]

101F      01 [                         DB    1 DUP (?)           ;MOV SECOND BYTE HERE
               ??
                 ]


1020                     RAM_AREA_END  LABEL  BYTE          ;
1020                     DATA    ENDS





                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                         ;               EQUATES                                                       ;
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

= 008F          ANALOG_FAIL       EQU    8FH     ;ANALOG FAILURE AS REPORTED BY LANA
= 0000          DMA               EQU    00H     ;BASE ADDRESS OF DMA
= 0003          DMA_CHAN3         EQU    03H     ;SELECTS CHAN 3 DMA
= 0004          DMA_MASK          EQU    04H     ;BIT FOR MASKING OF DMA REQUESTS
= 000A          DMA_SINGLE_MASK   EQU    0AH     ;ADD OF SINGLE MASK
= 0082          DMA_3_PAGE        EQU    82H     ;PAGE REG FOR DMA CHAN 3
= 00FB          ENABLE_INT2       EQU    0FBH    ;USE FOR IMR TO ENABLE INT REQ 2
= 00F7          ENABLE_INT3       EQU    0F7H    ;USE TO ENABBLE INT 3
= 00F3          ENABLE_INT2_3     EQU    0F3H    ;
= 00FD          ENABLE_INT9       EQU    0FDH    ;
= 0020          EOI               EQU    20H     ;ENABLE GENERAL INTERRUPTS
= 008B          FAIL_ERR_REPORT   EQU    8BH     ;CMD 41 DID NOT REPORT CORRECTLY
= 008A          FAIL_GO           EQU    8AH     ;FAILED TO GET A GO BIT
= 008E          HOT_RF            EQU    8EH     ;FLAG FOR POSSIBLE HOT CARRIER
= 0088          HOST_DETECT_HIC   EQU    88H     ;ERROR NUMBER FOR HOST DETECTING HIC
= 0020          INTA00            EQU    020H    ;CHIP 1
= 0021          INTA01            EQU    021H    ;CHIP 1
= 00A1          INTB01            EQU    0A1H    ;
= 0010          INT3_FLAG         EQU    10H     ;INT LEVEL 3 ACTIVE
= 0028          INT2_VECT         EQU    028H    ;PHYSICAL ADD FOR INT VECTOR 2 (A*4)
= 002C          INT3_VECT         EQU    02CH    ;"           "   "   "     "   3 (B*4)
= 004C          INT13_VECT        EQU    04CH    ;"           "   "   "     "   13(13H*4)
= 0170          INT5C_VECT        EQU    170H    ;   "        "   "   "     "   5C(5CH*4)
= 0218          INT86_VECT        EQU    218H    ;   "        "   "   "     "   86(86H*4)
= 04A2          LANA_0_STATS      EQU    4A2H    ;OFFSET FOR BIOS STATUS LANA 0
= 04A3          LANA_1_STATS      EQU    4A3H    ;OFFSET FOR BIOS STATUS LANA 1
= 000C          MASK_IRQ2_3       EQU    0CH     ;MASK FOR DISABLE INT REQ 2,3
= 008C          NO_CRD_PRESENT    EQU    8CH     ;ERROR CODE FOR NO CARD PRESENT
= 0360          STATUS_REG        EQU    360H    ;STATUS REGISTER FIRST CARD
= 0361          PARAMETER_REG     EQU    361H    ;
= 000E          PC_ID_ADD         EQU    0EH     ;OFFSET FOR PC IDENTY
= 00FC          PC3               EQU    0FCH    ;PC3 ID
= 0362          DATA_REG          EQU    362H    ;
= 0363          HOST_INTR_REG     EQU    363H    ;
= 0085          HOST_FAIL         EQU    085H    ;FLAG FOR HOST TEST FAIL
= 0004          RESET             EQU    04H     ;RESET BIT IN HIR
= 0080          RESET_INT_REQ     EQU    80H     ;HC = 1 IN STATUS REG =RESET REQ'S
= FFFF          ROS_CODE          EQU    0FFFFH  ;POINTER INTO ROS
= 0333          RF_COUNT          EQU    333H    ;45 SEC HOT RF DELAY
= 0089          SYNC_FAILED       EQU    89H     ;WE DID NOT GET IN SYNC
= 0222          T_O_CNT           EQU    222H    ;30 SECS (18.2 * 30 = 546 = 222H)
= 00FE          TIMER_INT_ENA     EQU    0FEH    ;ENABLE MASK FOR INT 0 - TIMER
```

```
                                                EXTRN   MAIN:NEAR               ;BIOS HANDLER FOR SOFT INT 5C
                                                EXTRN   HARD_FILE:NEAR          ;INTERCEPT HARD FILE SOFT INT 13
                                                EXTRN   REM_IPL:NEAR            ;BIOS HANDLER FOR IPL

                                                PUBLIC RAS_START

 0000                                   NETWORK         SEGMENT PARA PUBLIC     'CODE'
                                                ASSUME  CS:NETWORK
                                                ASSUME  DS:DATA
                                                ASSUME  SS:NOTHING
                                                ASSUME  ES:NOTHING


                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                        ;                 CODE START                                             ;
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

 0000                                   HOST_INTERFACE  PROC     FAR
 0000  55                                       MOD_ID  DB       055H           ;GENERIC BIOS HEADER
 0001  AA                                               DB       0AAH
 0002  10                                               DB        16D           ;# OF 512 BYTE BLOCKS
 0003                                   START:
 0003  EB 2D                                    JMP     SHORT HI1               ;JMP TO START OF CODE

 0005  36 33 36 30 37 31                        DB      '6360715 (C) COPYRIGHT  IBM CORP. 1984'     ;COPYRIGHT NOTICE
       35 20 28 43 29 20
       43 4F 50 59 52 49
       47 48 54 20 20 49
       42 4D 20 43 4F 52
       50 2E 20 31 39 38
       34
 002A  30 34 2F 31 39 2F                        DB      '04/19/84'
       38 34


 0032                                   RAS_START:
 0032                                   HI1:
 0032  FB                                       STI                             ;ENABLE INTERRUPTS
 0033  33 C0                                    XOR     AX,AX                   ;ESTABLISH SEG REG
 0035  8E D8                                    MOV     DS,AX
 0037  8E C0                                    MOV     ES,AX

 0039  FC                                       CLD
 003A  BF 1000 R                                MOV     DI,OFFSET RAM_AREA      ;GET STARTING RAM LOCATION
 003D  B9 0021                                  MOV     CX,(RAM_AREA_END-RAM_AREA) + 1  ;RAM SIZE
 0040  F3/ AA                                   REP     STOSB           ;CLEAR THE WORK AREA

 0042  26: A2 04A2                              MOV     ES:LANA_0_STATS,AL
 0046  26: A2 04A3                              MOV     ES:LANA_1_STATS,AL

 004A                                   PC_TYPE:
 004A  B8 FFFF                                  MOV     AX,ROS_CODE             ;ROS POINTER FOR DS
 004D  8E D8                                    MOV     DS,AX
 004F  BB 000E                                  MOV     BX,PC_ID_ADD
 0052  8A 07                                    MOV     AL,[BX]                 ;GET PC INFO
 0054  06                                       PUSH    ES
 0055  1F                                       POP     DS                      ;DS=ES
 0056  A2 1010 R                                MOV     PC_ID,AL                ;SAVE IT FOR LATER USE
 0059  E4 21                                    IN      AL,INTA01               ;IMR 1ST INT CHIP
 005B  A2 1007 R                                MOV     SAVE_MASK,AL


                                                ;---INSURE INT 2,3 MASKED OFF

 005E  E8 0708 R                                CALL    MASK_INT_2_3            ;MASK INT 2 & 3
 0061  E8 07FF R                                CALL    SAVE_INT_VECT           ;SAVE 2 & 3
 0064  E8 088E R                                CALL    SET_UP_TIME_TICK        ;POINT TIME TICK TO LANA
 0067  E4 21                                    IN      AL,INTA01               ;GET IMR
 0069  EB 00                                    JMP     SHORT $ + 2             ;DELAY
 006B  24 FE                                    AND     AL,TIMER_INT_ENA        ;ALLOW INT 0 TO RUN
 006D  E6 21                                    OUT     INTA01,AL               ;WRITE THE MASK

                                                ;------RESET LANA CARDS------

 006F  33 DB                                    xor     bx,bx                   ;
 0071  B0 04                                    MOV     AL,RESET                ;RESET BIT FOR HIR
 0073  BA 0363                                  MOV     DX,HOST_INTR_REG        ;ADD OF FIRST CARD
 0076  EE                                       OUT     DX,AL                   ;RESET 1ST CARD
 0077  BA 036B                                  MOV     DX,HOST_INTR_REG + 8    ;ADD OF 2ND CARD
 007A  EE                                       OUT     DX,AL                   ;RESET 2ND CARD
 007B  B9 00FF                                  MOV     CX,OFFH                 ;DELAY TIME
 007E                                   S1:
 007E  E2 FE                                    LOOP    S1                      ;DELAY

                                                ;LANA0 PRESENT?

 0080  BA 0363                                  MOV     DX,HOST_INTR_REG        ;ADD FOR LANA 0
 0083  E8 06F8 R                                CALL    LANA_PRESENT?           ;SEE IF INSTALLED

 0086  73 03                                    JNC     S2                      ;JMP LANA 0 INSTALLED
 0088  E9 014B R                                JMP     ANY_MORE_LANAS          ;SEE IF LANA 1 INSTALLED
 008B                                   S2:
                                                ;LANA0 ADAPTER INSTALLED
                                                ;--REMOVE RESET FROM LANA 0   PRIMARY CARD

 008B  E8 079C R                                CALL    REMOVE_RESET0           ;DO THE DOUBLE RESET TO LANA0
 008E  EB 13                                    JMP     SHORT RR1A              ;TEST LANA0 ADAPTER


                                                ;--REMOVE RESET FROM LANA 1

 0090                                   LANA1_PRESENT?:
 0090  BA 0363                                  MOV     DX,HOST_INTR_REG        ; HIR REG
 0093  03 D3                                    ADD     DX,BX                   ;OFFSET FOR LANA1
 0095  E8 06F8 R                                CALL    LANA_PRESENT?           ;LANA 1 INSTALLED?

 0098  73 03                                    JNC     LANA1_RESET             ;LANA 1 INSTALLED, REMOVE RESET
 009A  E9 014B R                                JMP     ANY_MORE_LANAS          ;COMMON HANDLER


 009D                                   LANA1_RESET:
                                                ;LANA1 INSTALLED TEST ADAPATER

 009D  E8 07AD R                                CALL    REMOVE_RESET1           ;DO THE DOUBLE RESET TO LANA1

 00A0  E8 0708 R                                CALL    MASK_INT_2_3            ;TURN OF INTS

                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                        ;                 TEST INTERFACE TO LANA                                 ;
```

```
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

 00A3                                   RR1A:
 00A3  E8 02F8 R                                 CALL    HIFT_TEST              ;TEST THE HIF PLA
 00A6  73 23                                     JNC     PRIME_CMD              ;JMP NO ERROR IN HIF TEST

                                                 ;INTERFACE TEST FAILED

 00A8  80 3E 100F R 80                           CMP     T_O_FLAG,80H           ;TIME OUT OCCUR?
 00AD  74 08                                     JZ      IN_SYNC?               ;JMP YES

                                                 ;HOST DETECTED A DATA FAILURE -NO TIME OUT & CARRY SET
                                                 ;NO INTERROGATION OF CMD 41

 00AF  B0 88                                     MOV     AL,HOST_DETECT_HIC     ;FAILURE FLAG
 00B1  E8 070F R                                 CALL    PC_ERROR               ;DISPLAY 30XX OR 31XX
 00B4  E9 014B R                                 JMP     ANY_MORE_LANAS         ;ANY MORE FOR TEST?

 00B7                                   IN_SYNC?:
                                                 ;DID LANA GET IN SYNC
 00B7  80 3E 1015 R 80                           CMP     NO_SYNC,80H            ;IS THE FAILED TO SYNC FLAG SET?
 00BC  75 0D                                     JNZ     PRIME_CMD              ;JMP  IF IT DID GET IN SYNC

                                                 ;DID NOT SYNC IN-ARE GO BIT AND CMD 41 ACTIVE
                                                 ;IF YES LANA HAS STATUS TO REPORT

 00BE  E8 02DF R                                 CALL    GO_BIT_CMD_41          ;RETURN WITH CF SET IF BOTH ACTIVE
 00C1  72 33                                     JC      P_C4                   ;JMP ACTIVE-GET 2ND BYTE OF CMD 41

                                                 ;FAILED TO SYNC IN AND NO GO BIT ACTIVE

 00C3  B0 89                                     MOV     AL,SYNC_FAILED         ;
 00C5  E8 070F R                                 CALL    PC_ERROR               ;GO DISPLAY ERROR
 00C8  E9 014B R                                 JMP     ANY_MORE_LANAS         ;SEE IF ANY MORE ADAPTERS TO TEST

                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                        ;           WAIT FOR PRIMARY CMD 41 INITIALIZATION COMPLETE          ;
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

 00CB                                   PRIME_CMD:
 00CB  BA 0360                                   MOV     DX,STATUS_REG          ;
 00CE  03 D3                                     ADD     DX,BX                  ;OFFSET
 00D0                                   P_C1:
 00D0  EC                                        IN      AL,DX                  ;GET THE STATUS REG
 00D1  A8 01                                     TEST    AL,01H                 ;IS THE GO BIT ON
 00D3  75 0F                                     JNZ     P_C2                   ;JMP GO ON
 00D5  80 3E 100F R 80                           CMP     T_O_FLAG,80H           ;
 00DA  75 F4                                     JNZ     P_C1                   ;JMP NO TIME OUT

                                                 ;TIME OUT AND NO GO BIT

 00DC  B0 8A                                     MOV     AL,FAIL_GO             ;NEVER GO A GO BIT
 00DE  E8 070F R                                 CALL    PC_ERROR               ;DISPLAY 30XX OR 31XX
 00E1  EB 68 90                                  JMP     ANY_MORE_LANAS         ;


 00E4                                   P_C2:

                                                 ;GO BIT ACTIVE  CHECK CMD

 00E4  BA 0361                                   MOV     DX,PARAMETER_REG       ;
 00E7  03 D3                                     ADD     DX,BX                  ;OFFSET
 00E9  EC                                        IN      AL,DX                  ;
 00EA  3C 41                                     CMP     AL,041H                ;IS IT INIT COMPLETE CMD?
 00EC  74 08                                     JZ      P_C4                   ;JMP YES

                                                 ;GO BIT AND NOT CMD 41

 00EE  B0 8B                                     MOV     AL,FAIL_ERR_REPORT     ;CMD 41 REPORT FAILURE
 00F0  E8 070F R                                 CALL    PC_ERROR               ;30XX OR 31XX
 00F3  EB 56 90                                  JMP     ANY_MORE_LANAS         ;

                                                 ;--WAS LANA SELF TEST SUCCESSFUL

 00F6                                   P_C4:
 00F6  E8 08FE R                                 CALL    STOP_TIMER             ;GROSS TIME OUT INACTIVE
 00F9  BA 0361                                   MOV     DX,PARAMETER_REG       ;
 00FC  03 D3                                     ADD     DX,BX                  ;OFFSET
 00FE  EC                                        IN      AL,DX                  ;GET 2ND BYTE OF PARAMETER REG

                                                 ;RETURN STATUS TO CMD

 00FF  50                                        PUSH    AX                     ;SAVE CMD 41 INFO
 0100  BA 0360                                   MOV     DX,STATUS_REG          ;
 0103  03 D3                                     ADD     DX,BX                  ;ADD OFFSET
 0105  B0 00                                     MOV     AL,00                  ;CC BITS = 0 AND GO = 0
 0107  EE                                        OUT     DX,AL                  ;PASS IT
 0108  58                                        POP     AX                     ;

                                                 ;WHAT WERE THE RESULTS

 0109  3C 80                                     CMP     AL,80H                 ;80=SUCCESSFUL
 010B  74 11                                     JZ      P_C5                   ;JMP LANA POST OK

                                                 ;LANA SELF TEST UNSUCCESSFUL - TEST FOR POSSIBLE HOT RF

 010D  3C 8E                                     CMP     AL,HOT_RF              ;
 010F  75 08                                     JNZ     P_C4A                  ;JMP NOT HOT RF

                                                 ;GO TEST RF

 0111  E8 0617 R                                 CALL    HOT_RF?                ;TEST RF IF HOT DISPLAY 3X41/42
                                                                                ;RF NOT HOT ,THEN CF NOT SET
 0114  73 08                                     JNC     P_C5                   ;IF RF NOT HOT INIT LANA
 0116  EB 33 90                                  JMP     ANY_MORE_LANAS         ;   OR 3X41 OR 3X42


 0119                                   P_C4A:
                                                 ;GO SORT THE ERRROR
 0119  E8 070F R                                 CALL    PC_ERROR               ;GO SORT THE ERROR AND DISPLAY IT
 011C  EB 2D                                     JMP     SHORT ANY_MORE_LANAS   ;CHECK FOR ANOTHER LANA

                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                        ;        ADAPTER TEST OK - POST STATUS INFO FOR NET BIOS             ;
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

 011E                                   P_C5:
                                                 ;--UP DATE BIOS STATUS--LANA  X OK---

 011E  B0 80                                     MOV     AL,80H                 ;PRESENT BIT
 0120  83 FB 00                                  CMP     BX,0                   ;THIS LANA 0 ?
 0123  75 15                                     JNZ     P_C7                   ;JMP NO

                                                 ;LANA 0 BIOS STATS

 0125  26: A2 04A2                               MOV     ES:LANA_0_STATS,AL     ;BIOS INFO FOR LANA 0
```

```
0129  A0 1011 R                 MOV    AL,TEMP_INT             ;ACTIVE LEVEL FOR LANA 0
012C  A2 1012 R                 MOV    LANA_0_INT,AL           ;
012F  08 06 1001 R              OR     INTS_ACTIVE,AL          ;SAVE INT LEVEL ACTIVE
0133  32 C0                     XOR    AL,AL                   ;
0135  A2 1011 R                 MOV    TEMP_INT,AL             ;CLEAR FOR TEST OF LANA 1
0138  EB 11                     JMP    SHORT ANY_MORE_LANAS    ;
013A                     P_C7:

                                ;LANA 1 BIOS STATS

013A  26: A2 04A3               MOV    ES:LANA_1_STATS,AL      ;BIOS INFO FOR LANA 1
013E  A0 1011 R                 MOV    AL,TEMP_INT             ;ACTIVE LEVEL FOR LANA 1
0141  A2 1013 R                 MOV    LANA_1_INT,AL           ;
0144  A2 1014 R                 MOV    LANA_1_ACTIVE,AL        ; + = LANA 1 ACTIVE
0147  08 06 1001 R              OR     INTS_ACTIVE,AL          ;SAVE THE LEVEL

                                ; HAVE BOTH LANAS BEEN TESTED?

014B                     ANY_MORE_LANAS:
014B  83 FB 00                  CMP    BX,0                    ;IS OFFSET=0= LANA 0
014E  75 06                     JNZ    ALL_TESTED              ;JMP BOTH LANAS TESTED
0150  BB 0008                   MOV    BX,08H                  ;OFFSET=8= LANA 1
0153  E9 0090 R                 JMP    LANA1_PRESENT?          ;GO TEST LANA 1

                                ; ALL LANAS HAVE BEEN TESTED

0156                     ALL_TESTED:
0156  26: F6 06 04A2 80         TEST   BYTE PTR ES:LANA_0_STATS,80H     ;LANA 0 TEST SUCCESSFUL?
015C  75 16                     JNZ    PR2                              ;JMP YES  TO PR2

015E  26: F6 06 04A3 80         TEST   BYTE PTR ES:LANA_1_STATS,80H     ;LANA 1 TEST SUCCESFUL?
0164  75 0E                     JNZ    PR2                              ;JMP YES

                                ;NO LANA INIT, WAS ANY ERROR REPORTED?

0166  80 3E 1005 R 80           CMP    FAIL_FLAG,80H           ;80=HARDWARE FAILURE
016B  74 0A                     JZ     AP3                     ;JMP ERROR REPORTED

                                ;NO LANA INIT, & NO ERR REPORTED, PRESENCE TEST FAILURE

016D  B0 8C                     MOV    AL,NO_CRD_PRESENT       ;UNIQUE ERROR CODE
016F  E8 070F R                 CALL   PC_ERROR                ;GO DISPLAY ERROR CODE
0172  EB 03                     JMP    SHORT AP3               ;

0174                     PR2:
0174  E8 081C R                 CALL   SET_BIOS_INTS           ;GO SET UP THE LANA BIOS INTERRUPTS
0177                     AP3:
0177  33 DB                     XOR    BX,BX                   ;LANA 0
0179  E8 08DC R                 CALL   SPECIAL_CLR             ;DISABLE ALL INTS FROM LANA 0

017C  BB 0008                   MOV    BX,08H                  ;LANA 1
017F  E8 08DC R                 CALL   SPECIAL_CLR             ;DISABLE ALL INTS FROM LANA1

0182                     PR3:
0182  E8 07C0 R                 CALL   RESTORE_TICK_VECTOR     ;
0185  E8 07CF R                 CALL   RESTORE_INT2_3          ;RESTORE INT VECTORS & INT MASKS
0188                     PR5:
0188  E8 0198 R                 CALL   CLEAR_WORK_AREA         ;GO CLEAR RAM USED IN TEST

018B  CB                        RET                            ;RETURN TO CALLING POST

018C                     HOST_INTERFACE          ENDP




                                          ;PROCEDURES
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
018C  BA 0360            CLR_LANA_INTS   PROC    NEAR          ;CLEAR ANY LANA INT REQUESTS
018C                            MOV    DX,STATUS_REG           ;
018F  03 D3                     ADD    DX,BX                   ;ADD OFFSET
0191  EC                        IN     AL,DX                   ;GET THE STATUS REG
0192  EB 00                     JMP    SHORT $ + 2             ;DELAY
0194  0C 80                     OR     AL,RESET_INT_REQ        ;OR IN BIT TO TURN OF INT REQUESTS
0196  EE                        OUT    DX,AL                   ;CLEAR INT REQUEST FROM LANA CARD
0197  C3                        RET                            ;
0198                     CLR_LANA_INTS   ENDP                  ;
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0198                     CLEAR_WORK_AREA          PROC    NEAR   ;CLEARS LOW RAM USED DURING LANA POR

0198  FC                        CLD                            ;INCREMENT
0199  33 C0                     XOR    AX,AX                   ;
019B  BF 1000 R                 MOV    DI,OFFSET RAM_AREA      ;STARTING BYTE OF RAM WORK AREA
019E  B9 0021                   MOV    CX,(RAM_AREA_END-RAM_AREA) + 1  ;RAM SIZE
01A1  F3/ AA                    REP    STOSB                   ;CLEAR THE AREA
01A3  C3                        RET                            ;
01A4                     CLEAR_WORK_AREA          ENDP         ;
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
01A4                     DISPLAY         PROC    NEAR          ;DISPLAY A MESSAGE
01A4  50                        PUSH   AX                      ;
01A5  53                        PUSH   BX                      ;
01A6                     D_1:
01A6  2E: 8A 04                          MOV    AL,CS:[SI]     ;PUT CHACTER IN AL
01A9  46                                 INC    SI             ;
01AA  50                                 PUSH   AX             ;SAVE PRINT CHAR
01AB  E8 01B6 R                          CALL   PRT_HEX        ;CALL VIDEO IO
01AE  58                                 POP    AX             ;RECOVER PRINT CHAR
01AF  3C 0A                              CMP    AL,10          ;WAS IT A LINE FEED?
01B1  75 F3                              JNE    D_1            ;NO KEEP PRINTING
01B3  5B                                 POP    BX             ;
01B4  58                                 POP    AX             ;
01B5  C3                                 RET
01B6                     DISPLAY ENDP                          ;

01B6                     PRT_HEX         PROC    NEAR          ;DISPLAY CHAR IN AL
01B6  B4 0E                              MOV    AH,14          ;
01B8  B7 00                              MOV    BH,0           ;
01BA  CD 10                              INT    10H            ;CALL VIDEO_IO
01BC  C3                                 RET                   ;
01BD                     PRT_HEX         ENDP                  ;
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


                                ;--PRIMARY CARD--

01BD  33 30 30 31 0D 0A  M1     DB     '3001',13,10   ;CPU FAILURE
01C3  33 30 30 32 0D 0A  M2     DB     '3002',13,10   ;ROS FAILURE
01C9  33 30 30 33 0D 0A  M3     DB     '3003',13,10   ;ID  FAILURE
01CF  33 30 30 34 0D 0A  M4     DB     '3004',13,10   ;RAM FAILURE
01D5  33 30 30 35 0D 0A  M5     DB     '3005',13,10   ;HIC  FAILURE
01DB  33 30 30 36 0D 0A  M6     DB     '3006',13,10   ;+-12V FAILURE
01E1  33 30 30 37 0D 0A  M7     DB     '3007',13,10   ;DIGITAL LOOPBACK FAILURE
01E7  33 30 30 38 0D 0A  M8     DB     '3008',13,10   ;HOST DETECTED HIC FAILURE
01ED  33 30 30 39 0D 0A  M9     DB     '3009',13,10   ;SYNC FAIL & NO GO BIT
01F3  33 30 31 30 0D 0A  M10    DB     '3010',13,10   ;HIC TEST OK & NO GO BIT
01F9  33 30 31 31 0D 0A  M11    DB     '3011',13,10   ;GO BIT & NO CMD 41
```

```
01FF  33 30 31 32 0D 0A      M12     DB      '3012',13,10    ;CARD NOT PRESENT
0205  33 30 31 33 0D 0A      M13     DB      '3013',13,10    ;DIGITAL FAILURE (FALL THRU)
020B  33 30 31 35 0D 0A      M15     DB      '3015',13,10    ;ANALOG FAILURE
0211  33 30 34 31 0D 0A      M50     DB      '3041',13,10    ;HOT CARRIER NOT ME
0217  33 30 34 32 0D 0A      M51     DB      '3042',13,10    ;HOT CARRIER  ME!
                                     ;--SECONDARY CARD--
021D  33 31 30 31 0D 0A      M21     DB      '3101',13,10    ;CPU FAILURE
0223  33 31 30 32 0D 0A      M22     DB      '3102',13,10    ;ROS FAILURE
0229  33 31 30 33 0D 0A      M23     DB      '3103',13,10    ;ID  FAILURE
022F  33 31 30 34 0D 0A      M24     DB      '3104',13,10    ;RAM FAILURE
0235  33 31 30 35 0D 0A      M25     DB      '3105',13,10    ;HIC  FAILURE
023B  33 31 30 36 0D 0A      M26     DB      '3106',13,10    ;+-12V FAILURE
0241  33 31 30 37 0D 0A      M27     DB      '3107',13,10    ;DIGITAL LOOPBACK FAILURE
0247  33 31 30 38 0D 0A      M28     DB      '3108',13,10    ;HOST DETECTED HIC FAILURE
024D  33 31 30 39 0D 0A      M29     DB      '3109',13,10    ;SYNC FAILED & NO GO BIT
0253  33 31 31 30 0D 0A      M30     DB      '3110',13,10    ;HIC TEST OK & NO GO BIT
0259  33 31 31 31 0D 0A      M31     DB      '3111',13,10    ;GO BIT & NO CMD 41
025F  33 31 31 32 0D 0A      M32     DB      '3112',13,10    ;CARD NOT PRESENT
0265  33 31 31 33 0D 0A      M33     DB      '3113',13,10    ;DIGITAL FAILURE (FELL THRU)
026B  33 31 31 35 0D 0A      M35     DB      '3115',13,10    ;ANALOG FAILURE 2ND CARD
0271  33 31 34 31 0D 0A      M60     DB      '3141',13,10    ;HOT CARRIER NOT ME
0277  33 31 34 32 0D 0A      M61     DB      '3142',13,10    ;HOT CARRIER ME!
                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
027D                         DMA_ADDRESS     PROC    NEAR            ;CONVERT SEG AND OFFSET TO 20 BIT ADD
                                                                     ;CALLED WITH SEGMENT(CS OR DS) IN AX
                                                                     ;& OFFSET IN DI
027D  B9 0004                        MOV     CX,4                    ;CH=0,CL=ROL COUNT
0280  D3 C0                          ROL     AX,CL                   ;AX=A<11::00,15::12>
0282  8A C8                          MOV     CL,AL                   ;CX=(0,0,A<3::0,15::12>)
0284  80 E1 0F                       AND     CL,00FH                 ;CX=SEG (0,0,0,A<19::16>)
0287  24 F0                          AND     AL,0F0H                 ;AX=SEG (A<15::04>,0)
0289  03 C7                          ADD     AX,DI                   ;AX=REAL (A<15::00>) &CX= SEGMENT ADJ
028B  12 CD                          ADC     CL,CH                   ;CX=REAL (0,0,0,A<19::16>)
028D  E6 0C                          OUT     DMA + 0CH,AL            ;SET THE BYTE F/F
028F  EB 00                          JMP     SHORT $ + 2             ;DELAY
0291  E6 06                          OUT     DMA + 6,AL              ;OUT A <07::00>
0293  8A C4                          MOV     AL,AH                   ;
0295  EB 00                          JMP     SHORT $ + 2             ;DELAY
0297  E6 06                          OUT     DMA + 6,AL              ;OUT A<15::08>
0299  EB 00                          JMP     SHORT $ + 2             ;DELAY
029B  8A C1                          MOV     AL,CL                   ;
029D  E6 82                          OUT     DMA_3_PAGE,AL           ;OUT A<19::16>
029F  C3                             RET                             ;
02A0                         DMA_ADDRESS     ENDP
                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
02A0                         ENABLE_INTS     PROC        NEAR        ;
                                     ;IF LANA0 ENABLE INTS 2 & 3  & 9
                                     ;IF LANA1 DETERMINE INT USED BY LANA0 & ENABLE THE REMAINING INT
                                     ;WITH LANA0 MASKED OFF
02A0  83 FB 00                       CMP     BX,0                    ;LANA 0 ?
02A3  74 21                          JZ      E13                     ;JMP LANA0
                                     ;WAS LANA0 TEST SUCCESSFUL
02A5  26: F6 06 04A2 80              TEST    BYTE PTR ES:LANA_0_STATS,80H     ;LANA 0 TEST SUCCESSFUL?
02AB  74 19                          JZ      E13                     ;JMP NO-ENABLE ALL INTS
02AD  F6 06 1012 R 08                TEST    LANA_0_INT,08           ;IS LANA0 ON INT 3?
02B2  75 09                          JNZ     E12                     ;JMP YES
                                     ;LANA0 ON INT 2
                                     ;MASK OF INT REQ 2
02B4  E4 21                          IN      AL,INTA01               ;GET CURRENT MASK
02B6  24 F7                          AND     AL,ENABLE_INT3          ;ENABLE INT 3
02B8  EB 00                          JMP     SHORT $ + 2             ;
02BA  E6 21                          OUT     INTA01,AL               ;NEW MASK  3 ON 2 OFF
02BC  C3                             RET                             ;
02BD                         E12:
                                     ;LANA0 IS ON INT 3
                                     ;MASK OF INT REQ 3
02BD  E4 21                          IN      AL,INTA01               ;GET CURRENT MASK
02BF  24 FB                          AND     AL,ENABLE_INT2          ;ENABLE INT 2
02C1  EB 00                          JMP     SHORT $ + 2             ;
02C3  E6 21                          OUT     INTA01,AL               ;NEW MASK 2 ON 3 OFF
02C5  C3                             RET
02C6                         E13:
                                     ;LANA0 ENABLE INT 2 & 3
02C6  E4 21                          IN      AL,INTA01               ;
02C8  24 F3                          AND     AL,ENABLE_INT2_3        ;
02CA  EB 00                          JMP     SHORT $ + 2             ;
02CC  E6 21                          OUT     INTA01,AL               ;
                                     ;ON A PC-3?
02CE  80 3E 1010 R FC                CMP     PC_ID,PC3               ;FC=3
02D3  75 09                          JNZ     E14                     ;JMP NOT = 3
02D5  E4 A1                          IN      AL,INTB01               ;GET IMR 2ND CHIP
02D7  A2 1008 R                      MOV     SAVE_MASKA,AL           ;SAVE IT
02DA  24 FD                          AND     AL,ENABLE_INT9          ;
02DC  E6 A1                          OUT     INTB01,AL               ;ENABLE 9
02DE                         E14:
02DE  C3                             RET                             ;
02DF                         ENABLE_INTS     ENDP
                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
02DF                         GO_BIT_CMD_41   PROC    NEAR            ;IF GO BIT & CMD 41 THEN SET CARRY FLAG
                                                                     ;ELSE CLEAR CARRY FLAG
02DF  BA 0360                        MOV     DX,STATUS_REG           ;GET THE GO BIT FROM
02E2  03 D3                          ADD     DX,BX                   ;THE STATUS REG
02E4  EC                             IN      AL,DX                   ;
02E5  A8 01                          TEST    AL,01                   ;GO BIT ON?
02E7  74 0D                          JZ      GBC1                    ;GO BIT NOT ACTIVE EXIT
                                     ;GO BIT ACTIVE  CHECK FOR CMD 41
02E9  BA 0361                        MOV     DX,PARAMETER_REG        ;
02EC  03 D3                          ADD     DX,BX                   ;OFFSET
02EE  EC                             IN      AL,DX                   ;
02EF  3C 41                          CMP     AL,041H                 ;IS IT INIT COMPLETE CMD?
02F1  75 03                          JNZ     GBC1                    ;JMP NO
02F3  F9                             STC                             ;GO BIT ACTIVE AND CMD 41
```

```
02F4  EB 01                     JMP     SHORT GBC2              ;
02F6                   GBC1:
02F6  F8                        CLC                             ;GO BIT*CMD 41  NOT ACTIVE
02F7                   GBC2:
02F7  C3                        RET                             ;
02F8                   GO_BIT_CMD_41    ENDP                    ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                       ;        THIS IS A COMMON PROC FOR LANA 0 & 1.THE PROC IS CALLED WITH A OFFSET;
                       ; OF 0 IN BX FOR LANA 0, AND A OFFSET OF 8 FOR LANA 1. THIS ROUTINE TESTS THE;
                       ; HIF PLA  REGISTERS AND CONTROL LOGIC.                                       ;
                       ;        1.VERIFY THE STATUS OF REGISTERS FOLLOWING RESET                     ;
                       ;        2.WRITE/READ  TO REGS AND VERIFY STATUS BITS                         ;
                       ;          THE FOLLOWING STEPS ARE DONE IN CONJUNCTION WITH THE LANA CPU     ;
                       ;        1.SET INT TO LANA (HOST SETS GO BIT)                                 ;
                       ;        2.WAIT FOR INT FROM LANA(LANA CLEARS GO BIT)                         ;
                       ;        3.DTI INT(DATA XFER LANA TO HOST)                                    ;
                       ;        4.DRE INT (DATA REG EMPTY INT LANA)                                  ;
                       ;        5.DATA XFER (HOST-->LANA INT)                                        ;
                       ;        6.DRE (LANA READS DATA CAUSES INT TO HOST)                           ;
                       ;        7.WRITE TO LANA CAUSE INT                                            ;
                       ;        8.DMA XFER HOST --> LANA 2 BYTES                                     ;
                       ;        9.DMA XFER LANA --> HOST TEST 2 BYTES DATA                           ;
                       ;                                                                             ;
                       ;                EXITS FROM PROC                                              ;
                       ;        1.SUCCESSFUL ***RETURN TO CALLER WITH CARRY FLAG CLEAR               ;
                       ;        2.FAILURE***RETURN TO CALLER WITH CARRY FLAG SET                     ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
02F8                   HIFT_TEST        PROC    NEAR            ;
02F8  32 E4                     XOR     AH,AH                   ;

                                ;----VERIFY STATUS REG 10H--

02FA  C6 06 1002 R 80           MOV     COUNT,80H               ;INITIALIZE THE COUNT  (  80 )
02FF  BA 0360                   MOV     DX,STATUS_REG           ;
0302  03 D3                     ADD     DX,BX                   ;ADD OFFSET FOR I/O ADDRESS
0304  EC                        IN      AL,DX                   ;GET STATUS REG
0305  3C 10                     CMP     AL,10H                  ;DRE SHOULD BE THE ONLY ACTIVE BIT
0307  75 6A                     JNZ     HIFT_FAIL1              ;JMP NON COMPARE

                                ;---WRT/READ DATA REG---AND TEST DRF & DRE--

0309  FE 06 1002 R              INC     COUNT                   ;( 81  )
030D  BA 0362                   MOV     DX,DATA_REG             ;
0310  03 D3                     ADD     DX,BX                   ;ADD OFFSET FOR I/O ADDRESS
0312  B0 AA                     MOV     AL,0AAH                 ;TEST BYTE
0314  EE                        OUT     DX,AL                   ;WRITE FIRST BYTE  AA
0315  EB 00                     JMP     SHORT $ + 2             ;DELAY
0317  F6 D0                     NOT     AL                      ;AA TO 55
0319  EE                        OUT     DX,AL                   ;WRITE 2ND BYTE    55

                                ;--TEST DRF & DRE---

031A  BA 0360                   MOV     DX,STATUS_REG           ;
031D  03 D3                     ADD     DX,BX                   ;
031F  EC                        IN      AL,DX                   ;GET SR
0320  3C 20                     CMP     AL,20H                  ;DRF =1  DRE = 0
0322  75 4F                     JNZ     HIFT_FAIL1              ;JMP ERROR

                                ;--GET FIRST DATA BYTE--

0324  FE 06 1002 R              INC     COUNT                   ;( 82  )
0328  BA 0362                   MOV     DX,DATA_REG             ;
032B  03 D3                     ADD     DX,BX                   ;
032D  EC                        IN      AL,DX                   ;FIRST DATA BYTE
032E  3C AA                     CMP     AL,0AAH                 ;
0330  75 41                     JNZ     HIFT_FAIL1              ;JMP DATA NON COMPARE

                                ;--TEST DRF & DRE AGAIN

0332  BA 0360                   MOV     DX,STATUS_REG           ;
0335  03 D3                     ADD     DX,BX                   ;
0337  EC                        IN      AL,DX                   ;GET SR
0338  3C 30                     CMP     AL,30H                  ;DRF = 1  DRE = 1
033A  75 37                     JNZ     HIFT_FAIL1              ;JMP NON COMPARE

                                ;--GET 2ND DATA BYTE--

033C  BA 0362                   MOV     DX,DATA_REG             ;
033F  03 D3                     ADD     DX,BX                   ;
0341  EC                        IN      AL,DX                   ;GET DATA
0342  3C 55                     CMP     AL,055H                 ;
0344  75 2D                     JNZ     HIFT_FAIL1              ;JMP DATA NON COMPARE

                                ;--TEST DRF & DRE FOR THE LAST TIME--

0346  BA 0360                   MOV     DX,STATUS_REG           ;
0349  03 D3                     ADD     DX,BX                   ;
034B  EC                        IN      AL,DX                   ;GET THE STATUS
034C  3C 10                     CMP     AL,10H                  ;DRF = 0   DRE = 1
034E  75 23                     JNZ     HIFT_FAIL1              ;JMP  NON COMPARE

                                ;--TEST REMAINING DATA---

0350  BA 0362                   MOV     DX,DATA_REG             ;
0353  03 D3                     ADD     DX,BX                   ;

0355  B0 FF                     MOV     AL,0FFH                 ;
0357  EE                       OUT     DX,AL                    ;  FF
0358  EB 00                     JMP     SHORT $ +2              ;
035A  EC                        IN      AL,DX                   ;
035B  3C FF                     CMP     AL,0FFH                 ;
035D  75 14                     JNZ     HIFT_FAIL1              ;
035F  B0 01                     MOV     AL,01H                  ;TEST BYTE
0361  EE                        OUT     DX,AL                   ;WRITE FIRST BYTE  01
0362  EB 00                     JMP     SHORT $ + 2             ;DELAY
0364  EC                        IN      AL,DX                   ;GET THE DATA
0365  3C 01                     CMP     AL,01H                  ;
0367  75 0A                     JNZ     HIFT_FAIL1              ;JMP DATA NO COMPARE
0369  B0 00                     MOV     AL,00H                  ;TEST BYTE
036B  EE                        OUT     DX,AL                   ;WRITE FIRST BYTE    00
036C  EB 00                     JMP     SHORT $ + 2             ;DELAY
036E  EC                        IN      AL,DX                   ;GET THE DATA
036F  3C 00                     CMP     AL,00H                  ;
0371  74 03                     JZ      RR4                     ;JMP DATA NO COMPARE
0373                   HIFT_FAIL1:
0373  E9 060D R                 JMP     HIFT_FAILED             ;END OF PROC

                                ;----TEST PARAMETR REG--------

0376                   RR4:
0376  FE 06 1002 R              INC     COUNT                   ;( 83  )
037A  BA 0361                   MOV     DX,PARAMETER_REG        ;
037D  03 D3                     ADD     DX,BX                   ;ADD OFFSET FOR I/O ADDRESS
037F  B0 AA                     MOV     AL,0AAH                 ;
```

```
0381  EE                    OUT     DX,AL               ;1  AA
0382  B0 55                 MOV     AL,055H             ;
0384  EB 00                 JMP     SHORT $ +2          ;
0386  EE                    OUT     DX,AL               ;2  55
0387  B0 FF                 MOV     AL,0FFH             ;
0389  EB 00                 JMP     SHORT $ +2          ;
038B  EE                    OUT     DX,AL               ;3  FF
038C  B0 01                 MOV     AL,01H              ;
038E  EB 00                 JMP     SHORT $ + 2         ;
0390  EE                    OUT     DX,AL               ;4  01
0391  B0 00                 MOV     AL,00               ;
0393  EB 00                 JMP     SHORT $ + 2         ;
0395  EE                    OUT     DX,AL               ;5  00
0396  EB 00                 JMP     SHORT $ + 2         ;
0398  EE                    OUT     DX,AL               ;6  00
0399  EB 00                 JMP     SHORT $ + 2         ;
039B  EE                    OUT     DX,AL               ;7  00

                            ;--VERIFY THE DATA-----

039C  EB 00                 JMP     SHORT $ + 2         ;
039E  EC                    IN      AL,DX               ;
039F  3C AA                 CMP     AL,0AAH             ;1 AA
03A1  75 2D                 JNZ     HIFT_FAIL2          ;
03A3  EB 00                 JMP     SHORT $ + 2         ;
03A5  EC                    IN      AL,DX               ;
03A6  3C 55                 CMP     AL,055H             ;2 55
03A8  75 26                 JNZ     HIFT_FAIL2          ;
03AA  EB 00                 JMP     SHORT $ + 2         ;
03AC  EC                    IN      AL,DX               ;
03AD  3C FF                 CMP     AL,0FFH             ;3 FF
03AF  75 1F                 JNZ     HIFT_FAIL2          ;
03B1  EB 00                 JMP     SHORT $ + 2         ;
03B3  EC                    IN      AL,DX               ;
03B4  3C 01                 CMP     AL,01H              ;4 01
03B6  75 18                 JNZ     HIFT_FAIL2          ;
03B8  EB 00                 JMP     SHORT $ + 2         ;
03BA  EC                    IN      AL,DX               ;
03BB  3C 00                 CMP     AL,00H              ;5 00
03BD  75 11                 JNZ     HIFT_FAIL2          ;
03BF  EB 00                 JMP     SHORT $ + 2         ;
03C1  EC                    IN      AL,DX               ;
03C2  3C 00                 CMP     AL,00H              ;6 00
03C4  75 0A                 JNZ     HIFT_FAIL2          ;
03C6  EB 00                 JMP     SHORT $ + 2         ;
03C8  EC                    IN      AL,DX               ;
03C9  3C 00                 CMP     AL,00H              ;7 00
03CB  75 03                 JNZ     HIFT_FAIL2          ;
03CD  EB 04 90              JMP     RR6                 ;
03D0                    HIFT_FAIL2:
03D0  E9 060D R             JMP     HIFT_FAILED         ;END OF PROC

                            ;---TEST THE HOST INTERFACE REG---

03D3                    RR6:
03D3  FE 06 1002 R          INC     COUNT               ;( 84  )
03D7  BA 0363               MOV     DX,HOST_INTR_REG    ;
03DA  03 D3                 ADD     DX,BX               ;
03DC  B0 AA                 MOV     AL,0AAH             ; AA H
03DE  E8 0955 R             CALL    WRT_REG             ;
03E1  E8 0796 R             CALL    READ_REG            ;
03E4  75 4D                 JNZ     HIFT_FAIL3          ;

03E6  B0 52                 MOV     AL,052H             ;  52H
03E8  E8 0955 R             CALL    WRT_REG             ;
03EB  E8 0796 R             CALL    READ_REG            ;
03EE  75 43                 JNZ     HIFT_FAIL3          ;

03F0  B0 FB                 MOV     AL,0FBH             ;  FBH
03F2  E8 0955 R             CALL    WRT_REG             ;
03F5  E8 0796 R             CALL    READ_REG            ;
03F8  75 39                 JNZ     HIFT_FAIL3          ;

03FA  B0 01                 MOV     AL,01H              ; 01H
03FC  E8 0955 R             CALL    WRT_REG             ;
03FF  E8 0796 R             CALL    READ_REG            ;
0402  75 2F                 JNZ     HIFT_FAIL3          ;

0404  B0 00                 MOV     AL,00H              ; 00H
0406  E8 0955 R             CALL    WRT_REG             ;
0409  E8 0796 R             CALL    READ_REG            ;
040C  75 25                 JNZ     HIFT_FAIL3          ;

                            ;---TEST THE STATUS REGISTER--

040E  FE 06 1002 R          INC     COUNT               ;( 85  )
0412  BA 0360               MOV     DX,STATUS_REG       ;
0415  03 D3                 ADD     DX,BX               ;
0417  B0 02                 MOV     AL,02H              ;
0419  E8 0955 R             CALL    WRT_REG             ;GO WRITE THE STATUS REG  02
041C  E8 0796 R             CALL    READ_REG            ;READ IT
041F  24 07                 AND     AL,07H              ;REMOVE UNWANTED BITS
0421  3C 02                 CMP     AL,02H              ;
0423  75 0E                 JNZ     HIFT_FAIL3          ;

0425  B0 05                 MOV     AL,05H              ;
0427  E8 0955 R             CALL    WRT_REG             ;GO WRITE THE STATUS REG    05
042A  E8 0796 R             CALL    READ_REG            ;READ IT
042D  24 07                 AND     AL,07H              ;REMOVE UNWANTED BITS
042F  3C 05                 CMP     AL,05H              ;
0431  74 03                 JZ      IT0                 ;status ok

                        ;;;LEAVE SR = 05-TELLS LANA THAT HOST ATTACHED;;;;

0433                    HIFT_FAIL3:
0433  E9 060D R             JMP     HIFT_FAILED         ;

                            ;----HOST INTERFACE INTERRUPT TESTS
                            ;WAIT FOR THE LANA TO SET THE CC0 BIT IN STATUS REGISTER

0436                    INTERRUPT_TESTS:
0436                    IT0:
0436  FE 06 1002 R          INC     COUNT               ;( 86  )
043A  E8 08F5 R             CALL    START_TIMER         ;
043D  BA 0360               MOV     DX,STATUS_REG       ;
0440  03 D3                 ADD     DX,BX       ;ADD OFFSET FOR I/O ADDRESS
0442                    IT_1:
                            ;--ADDED--
0442  80 3E 100F R 80       CMP     T_O_FLAG,80H        ;
0447  75 08                 JNZ     IT_1A               ;JMP NO TIME OUT
```

```
                                    ;TIME OUT HAS ELASPED SET FAIL TO SYNC FLAG
 0449  C6 06 1015 R 80              MOV     NO_SYNC,80H             ;SET FLAG
 044E  E9 04ED R                    JMP     HIFT_FAIL4              ;JMP GROSS TIME OUT
                                    ;--
 0451                         IT_1A:
 0451  EC                           IN      AL,DX                   ;GET THE STATUS REGISTER
 0452  A8 02                        TEST    AL,02H                  ;LANA SETS CCO = 1 TO SYNC IN
 0454  74 EC                        JZ      IT_1                    ;JMP IF CCO NOT SET YET
                                    ;---SET VECTORS FOR 2 & 3-----
 0456  E8 08B2 R                    CALL    SET_VECT_2_3            ;
                                    ;--SET THE GI ENABLE FOR THE HOST---
 0459  BA 0363                      MOV     DX,HOST_INTR_REG        ;
 045C  03 D3                        ADD     DX,BX          ;ADD OFFSET
 045E  B0 01                        MOV     AL,01                   ;GO INTERRUPT ENABLE FOR HOST
 0460  EE                           OUT     DX,AL                   ;ENABLE THE HOST INT
                              ;10-23
 0461  E8 018C R                    CALL    CLR_LANA_INTS           ;CLEAR ANY INTS THAT MAY BE SET
                                    ;--ENABLE INT 2 OR 3------------
                              ;10-16
 0464  E8 02A0 R                    CALL    ENABLE_INTS             ;LANA0=INT 2,3 LANA1=INT ?
 0467  90                           NOP                             ;
 0468  90                           NOP                             ;
 0469  90                           NOP                             ;
 046A  90                           NOP                             ;
 046B  C6 06 1000 R 00              MOV     INT_OCCUR?,00H          ;CLEAR THE FLAG
                                    ;------SET GO BIT TO LANA----
 0470  BA 0360                      MOV     DX,STATUS_REG           ;
 0473  03 D3                        ADD     DX,BX                   ;ADD OFFSET
 0475  B0 01                        MOV     AL,01                   ;GO BIT INTERRUPT
 0477  EE                           OUT     DX,AL                   ;THIS SHOULD CAUSE A LANA INT
 0478  FE 06 1002 R                 INC     COUNT                   ;( 87  )
                                    ;---NOW TEST THE CLEAR GO INTERRUPT TO THE HOST--
                                    ;---WAIT FOR THE LANA TO CLEAR GO AND CAUSE A HOST INT
 047C                         IT_7:
 047C  E8 093A R                    CALL    TIME_OUT_OR_INT?        ;SEE IF GROSS TIME OUT OR INT OCCURRED
 047F  74 FB                        JZ      IT_7                    ;JMP NO T_O OR INT
 0481  80 FC 80                     CMP     AH,80H                  ;WAS IT A INT?
 0484  74 02                        JZ      IT_8                    ;JMP INT OCCURRED
 0486  EB 65                        JMP     SHORT HIFT_FAIL4
 0488                         IT_8:
                                    ;--SAVE INT INFO-----
 0488  83 FB 00                     CMP     BX,0                    ;WORKING ON LANA 0?
 048B  74 09                        JZ      IT8A                    ;JMP YES
 048D  A0 1011 R                    MOV     AL,TEMP_INT             ;
 0490  A2 1013 R                    MOV     LANA_1_INT,AL           ;SAVE THE INT LEVEL  LANA 1
 0493  EB 07 90                     JMP     IT8B                    ;
 0496                         IT8A:
 0496  A0 1011 R                    MOV     AL,TEMP_INT             ;
 0499  A2 1012 R                    MOV     LANA_0_INT,AL           ;SAVE THE INT LEVEL LANA 0
 049C                         IT8B:
                              ;DATA REGISTER INTERRUPT
                                   ;TEST INTERRUPTS TO HOST AND LANA
                                    ;DATA XFER  LANA ----> HOST
 049C  FE 06 1002 R                 INC     COUNT                   ;( 88  )
 04A0  C6 06 1000 R 00              MOV     INT_OCCUR?,00H          ;CLEAR THE INT FLAG
 04A5  B0 10                        MOV     AL,10H                  ;DATA XFER INTERRUPT ENABLE DTI
 04A7  BA 0363                      MOV     DX,HOST_INTR_REG        ;
 04AA  03 D3                        ADD     DX,BX                   ;ADD OFFSET
 04AC  EE                           OUT     DX,AL                   ;ENABLE HOST FOR DTI INT FROM LANA
                                    ;WAIT FOR THE INT FROM LANA
 04AD                         IT_9:
 04AD  E8 093A R                    CALL    TIME_OUT_OR_INT?        ;SEE IF GROSS TIME OUT OR INT OCCURRED
 04B0  74 FB                        JZ      IT_9                    ;JMP NO T_O OR INT
 04B2  80 FC 80                     CMP     AH,80H                  ;WAS IT A INT?
 04B5  74 02                        JZ      IT_10                   ;JMP INT OCCURRED
 04B7  EB 34                        JMP     SHORT HIFT_FAIL4
 04B9                         IT_10:
                                    ;---READ BYTE FROM DATA REG (DRE) CAUSE INT TO LANA
 04B9  FE 06 1002 R                 INC     COUNT                   ;( 89  )
 04BD  BA 0362                      MOV     DX,DATA_REG             ;
 04C0  03 D3                        ADD     DX,BX                   ;ADD OFFSET
 04C2  EC                           IN      AL,DX                   ;READ DATA REG/ CAUSE DRE /INT LANA
                                    ;--CLEAR THE DTI BIT IN HIR
 04C3  BA 0363                      MOV     DX,HOST_INTR_REG        ;
 04C6  03 D3                        ADD     DX,BX                   ;ADD OFFSET
 04C8  B0 00                        MOV     AL,00H                  ;CLEAR THE DTI BIT
 04CA  EE                           OUT     DX,AL                   ;DO IT
                                    ;--SYNC THE LANA WITH CCO BIT--
 04CB  B0 02                        MOV     AL,02H                  ;SET CCO = 1
 04CD  BA 0360                      MOV     DX,STATUS_REG           ;
 04D0  03 D3                        ADD     DX,BX                   ;OFFSET
 04D2  EE                           OUT     DX,AL                   ;SYNC IT
                                    ;---DATA XFER HOST--->LANA
 04D3  C6 06 1000 R 00              MOV     INT_OCCUR?,00H          ;CLEAR THE FLAG
 04D8  BA 0363                      MOV     DX,HOST_INTR_REG        ;
 04DB  03 D3                        ADD     DX,BX                   ;ADD OFFSET
 04DD  B0 08                        MOV     AL,08H                  ;DD=1=HOST TO LANA
 04DF  EE                           OUT     DX,AL                   ;
                                    ;--NOW ENABLE THE DTI INTERRUPT--
 04E0  B0 18                        MOV     AL,18H                  ;DD=1 AND DTI = 1
 04E2  EE                           OUT     DX,AL                   ;
```

```
                                                ;--SHOULD GET A DRE INTERRUPT--

 04E3                                   IT11:

 04E3  E8 093A R                                CALL    TIME_OUT_OR_INT?        ;SEE IF GROSS TIME OUT OR INT OCCURRED
 04E6  74 FB                                    JZ      IT11                    ;JMP NO T_O OR INT
 04E8  80 FC 80                                 CMP     AH,80H                  ;WAS IT A INT?
 04EB  74 03                                    JZ      IT12                    ;JMP INT OCCURRED
 04ED                                   HIFT_FAIL4:
 04ED  E9 060D R                                JMP     HIFT_FAILED             ;END OF PROC

 04F0                                   IT12:
                                                ;---CLEAR DTI---

 04F0  B0 08                                    MOV     AL,08H                  ;CLEAR THE DTI LEAVE DD = 1
 04F2  BA 0363                                  MOV     DX,HOST_INTR_REG        ;
 04F5  03 D3                                    ADD     DX,BX                   ;
 04F7  EE                                       OUT     DX,AL                   ;CLEAR THE DTI IN HIR


                                        ;---WRITE TO DATA REG, CAUSE LANA INT

 04F8  BA 0362                                  MOV     DX,DATA_REG             ;
 04FB  03 D3                                    ADD     DX,BX                   ;ADD OFFSET
 04FD  EE                                       OUT     DX,AL                   ;


                                        ;4.3.6.4 DATA REGISTER DMA



                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                        ;                           DMA ADDRESSING                                          ;
                                        ;    CHAN 0             CHAN 1              CHAN 2              CHAN 3               ;
                                        ; 00 R/W ADDRESS   02  R/W ADDRESS   04  R/W ADDRESS   06  R/W ADDDRE                ;
                                        ; 01 R/W WD/CNT    03  R/W WD/CNT    05  R/W WD/CNT    07  R/W WD/CNT                ;
                                        ;                                                                                   ;
                                        ;                                                                                   ;
                                        ;;;;;;;;        08  READ STATUS REG                            ;;;;;;;;;;;;;;;;;;;;;
                                                ;       WRITE COMMAND REG                              ;
                                                ;       09  WRITE REQUEST REG                          ;
                                                ;       0A  WRITE SINGLE MASK REG BIT                  ;
                                                ;       0B  WRITE MODE REG                             ;
                                                ;       0C  CLEAR BYTE POINTER  F/F                    ;
                                                ;       0D  READ TEMPORARY REG                         ;
                                                ;           MASTER CLEAR                               ;
                                                ;       0E  ILLEGAL                                    ;
                                                ;       0F  WRITE ALL MASK REG BITS                    ;
                                                ;                                                      ;
                                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


                                              ;DMA XFER HOST ---> LANA  ( 2 BYTES AA/55 )

                                                ;SET UP THE DMA

 04FE  FE 06 1002 R                             INC     COUNT                   ;( 8A  )
                                                ;MASK OF DMA CHAN 3

 0502  B0 07                                    MOV     AL,DMA_MASK + DMA_CHAN3 ;MASK OFF DMA3 REQUESTS
 0504  E6 0A                                    OUT     DMA_SINGLE_MASK,AL      ;

 0506  C6 06 1000 R 00                          MOV     INT_OCCUR?,00H          ;CLEAR THE INT FLAG

                                                ;SUPPLY THE 20 BIT DMA ADDRESS

 050B  C6 06 101E R 55                          MOV     DMA_DATA,055H           ;
 0510  C6 06 101F R AA                          MOV     DMA_DATA + 1,0AAH       ;SUPPLY DATA FOR DMA XFER

 0515  33 C0                                    XOR     AX,AX                   ;FOR CONVERSION ROUTINE
 0517  BF 101E R                                MOV     DI,OFFSET DMA_DATA      ;
 051A  E8 027D R                                CALL    DMA_ADDRESS             ;SUPPLY THE DMA WITH ADDRESS


                                                ;SET THE WORD COUNT

 051D  E8 08CD R                                CALL    SET_WD_CNT              ;GO SET WORD COUNT FOR XFER OF 2

 0520  C6 06 1000 R 00                          MOV     INT_OCCUR?,00H          ;CLEAR THE INT FLAG

                                                ;SET THE MODE FOR READ

 0525  B0 4B                                    MOV     AL,4BH                  ;SINGLE MODE,ADD INC,AUTO DIS,READ,03
 0527  E6 0B                                    OUT     DMA + 0BH,AL            ;WRITE THE MODE REGISTER

                                                ;SET HIR FOR DMA XFER  FROM HOST TO LANA

 0529  BA 0363                                  MOV     DX,HOST_INTR_REG        ;
 052C  03 D3                                    ADD     DX,BX                   ;ADD OFFSET
 052E  B0 68                                    MOV     AL,68H                  ;DD=1=HOST TO LANA
                                                                                ;DTD=1=DMA
                                                                                ;TC1=1=TERMINAL COUNT INT TO HOST
 0530  EE                                       OUT     DX,AL                   ;SET UP HIR

                                                ;REMOVE THE MASK FROM CHAN 3

 0531  B0 03                                    MOV     AL,03                   ;CLEAR MASK BIT FOR CHAN 3
 0533  E6 0A                                    OUT     DMA + 0AH,AL            ;WRT SINGLE MASK REG
                                                                                ;2 DAM XFERS SHOULD OCCUR

                                               ;FIELD THE INT FROM TERMINAL COUNT

 0535                                   DMA1:

 0535  E8 093A R                                CALL    TIME_OUT_OR_INT?        ;SEE IF GROSS TIME OUT OR INT OCCURRED
 0538  74 FB                                    JZ      DMA1                    ;JMP NO T_O OR INT
 053A  80 FC 80                                 CMP     AH,80H                  ;WAS IT A INT?
 053D  74 02                                    JZ      DMA2                    ;JMP INT OCCURRED
 053F  EB 5D                                    JMP     SHORT HIFT_FAIL5        ;
 0541                                   DMA2:
                                                ;--CLEAR THE DTD BIT--

 0541  FE 06 1002 R                             INC     COUNT                   ;( 8B  )
 0545  B0 08                                    MOV     AL,08H                  ;CLEAR THE DTD BIT LEAVE DD ON FOR NOW
 0547  BA 0363                                  MOV     DX,HOST_INTR_REG        ;
 054A  03 D3                                    ADD     DX,BX                   ;ADD OFFSET
 054C  EE                                       OUT     DX,AL

                                                ;--CLEAR THE DD BIT --

 054D  B0 00                                    MOV     AL,00H                  ;
 054F  EE                                       OUT     DX,AL                   ;DTD AND DD BOTH CLEAR NOW

                                               ;---DMA XFER LANA --->HOST

                                                ;MASK OF DMA CHAN 3
```

```
0550  B0 07                     MOV    AL,DMA_MASK + DMA_CHAN3 ;MASK OFF DMA3 REQUESTS
0552  E6 0A                     OUT    DMA_SINGLE_MASK,AL      ;

0554  C6 06 1000 R 00           MOV    INT_OCCUR?,00H          ;CLEAR THE INT FLAG

                                ;SET DMA ADDRESS

0559  BF 1003 R                 MOV    DI,OFFSET DATA_XFER     ;PUT THE DMA DATA HERE
055C  33 C0                     XOR    AX,AX                   ;SEGMENT = 0
055E  E8 027D R                 CALL   DMA_ADDRESS             ;CONVERT SEGMENT & OFFSET TO 20 BIT ADD

                                ;SET THE WORD COUNT

0561  E8 08CD R                 CALL   SET_WD_CNT              ;SET FOR XFER 2 BYTES

                                ;SET THE MODE FOR WRITE

0564  B0 47                     MOV    AL,47H                  ;SINGLE MODE,ADD INC,AUTO DIS,WRITE,03
0566  E6 0B                     OUT    DMA + 0BH,AL            ;WRITE THE MODE REGISTER

                                ;SET HIR FOR DMA XFER LANA TO HOST

0568  BA 0363                   MOV    DX,HOST_INTR_REG        ;
056B  03 D3                     ADD    DX,BX          ;ADD OFFSET
056D  B0 60                     MOV    AL,60H                  ;DD=0=LANA TO HOST
                                                               ;DTD=1=DMA
                                                               ;TC1=1=TERMINAL COUNT INT TO HOST
056F  EE                        OUT    DX,AL                   ;SET UP HIR

                                ;REMOVE THE MASK FROM CHAN 3

0570  B0 03                     MOV    AL,03                   ;CLEAR MASK BIT FOR CHAN 3
0572  E6 0A                     OUT    DMA + 0AH,AL            ;WRT SINGLE MASK REG

                                ;FIELD TERMINAL COUNT INTERRUPT FROM DMA XFER

0574                      DMA3:

0574  E8 093A R                 CALL   TIME_OUT_OR_INT?        ;SEE IF GROSS TIME OUT OR INT OCCURRED
0577  74 FB                     JZ     DMA3                    ;JMP NO T_O OR INT
0579  80 FC 80                  CMP    AH,80H                  ;WAS IT A INT?
057C  74 02                     JZ     DMA4                    ;JMP INT OCCURRED
057E  EB 1E                     JMP    SHORT HIFT_FAIL5        ;

0580                      DMA4:
                                ;--CLEAR THE DTD AND TCI BIT--

0580  FE 06 1002 R              INC    COUNT                   ;( 8C  )
0584  B0 00                     MOV    AL,00H                  ;CLEAR THE HIR
0586  BA 0363                   MOV    DX,HOST_INTR_REG        ;
0589  03 D3                     ADD    DX,BX                   ;OFFSET
058B  EE                        OUT    DX,AL                   ;

                                ;--CHECK THE DATA FROM THE DMA XFER

058C  33 C0                     XOR    AX,AX                   ;
058E  8E D8                     MOV    DS,AX                   ;SET UP DS
0590  A0 1003 R                 MOV    AL,DATA_XFER            ;GET THE FIRST DATA BYTE
0593  3C 55                     CMP    AL,055H                 ;
0595  75 07                     JNZ    HIFT_FAIL5              ;JMP DATA DOESN'T COMP
0597  A0 1004 R                 MOV    AL,DATA_XFER +1         ;GET SECOND BYTE
059A  3C AA                     CMP    AL,0AAH                 ;
059C  74 03                     JZ     DMA5                    ;JMP DATA COMPARE

059E                      HIFT_FAIL5:
059E  EB 6D 90                  JMP    HIFT_FAILED             ;

05A1                      DMA5:
05A1  BA 0362                   MOV    DX,DATA_REG             ;CLEAR THE DATA REG
05A4  03 D3                     ADD    DX,BX                   ;
05A6  EC                        IN     AL,DX                   ;



                          ;4.3.6.5.  INTERFACE CONTROL

05A7                      IC4:

                                ;GET READY TO INT ON THE HOST CONTROL REQ BEING ENABLED

05A7  FE 06 1002 R              INC    COUNT                   ;( 8D  )
05AB  C6 06 1000 R 00           MOV    INT_OCCUR?,00           ;CLEAR THE FLAG

05B0  BA 0363                   MOV    DX,HOST_INTR_REG        ;
05B3  03 D3                     ADD    DX,BX                   ;ADD OFFSET
05B5  B0 82                     MOV    AL,82H                  ;HCI=1=INT ON HOST CONTROL
                                                               ;HCR=1=HOST CONTROL REQUEST

05B7  EE                        OUT    DX,AL                   ;SET  UP THE HOST


                               ;FIELD THE INT

05B8                      IC5:

05B8  FE 06 1002 R              INC    COUNT                   ;( 8E  )
05BC  E8 093A R                 CALL   TIME_OUT_OR_INT?        ;SEE IF GROSS TIME OUT OR INT OCCURRED
05BF  74 F7                     JZ     IC5                     ;JMP NO T_O OR INT
05C1  80 FC 80                  CMP    AH,80H                  ;WAS IT A INT?
05C4  74 02                     JZ     IC5A                    ;JMP INT OCCURRED
05C6  EB 45                     JMP    SHORT HIFT_FAILED

                                ;--CHECK THE HC BIT IN STATUS REG--
05C8                      IC5A:
05C8  EC                        IN     AL,DX                   ;GET STATUS REG
05C9  A8 80                     TEST   AL,80H                  ;HC SHOULD BE ACTIVE
05CB  75 03                     JNZ    IC6                     ;JMP HC ACTIVE
05CD  EB 3E 90                  JMP    HIFT_FAILED             ;HC NOT ACTIVE

                                ;--WAIT FOR LANA TO SET CCO BIT = 0

05D0                      IC6:

                                ;--ADDED--
05D0  80 3E 100F R 80           CMP    T_O_FLAG,80H            ;
05D5  75 03                     JNZ    IC6A                    ;JMP NO TIME OUT
05D7  EB 34 90                  JMP    HIFT_FAILED             ;JMP GROSS TIME OUT
                                ;--
05DA                      IC6A:
05DA  BA 0360                   MOV    DX,STATUS_REG           ;
05DD  03 D3                     ADD    DX,BX                   ;
05DF  EC                        IN     AL,DX                   ;GET STATUS REG
05E0  A8 02                     TEST   AL,02H                  ;IS CCO = 0
05E2  75 EC                     JNZ    IC6                     ;JMP IF CCO = 1
```

```
                                    ;GENERATE A HC CLEAR INT TO LANA

05E4  BA 0363                       MOV    DX,HOST_INTR_REG        ;
05E7  03 D3                         ADD    DX,BX                   ;ADD OFFSET
05E9  B0 00                         MOV    AL,00                   ;TURN OFF THE HOST CONTROL REQUEST
05EB  EE                            OUT    DX,AL                   ;INTERRUPT THE LANA

                                    ;--INSURE HCR IS NOT ACCEPTED WHEN LANA HAS HCE = 0
05EC  90                            NOP                            ;
05ED  B0 02                         MOV    AL,02H                  ;HOST CONTROL REQUEST
05EF  EE                            OUT    DX,AL                   ;REQUEST HOST CONTROL
05F0  90                            NOP                            ;

                                    ;--VERIFY NO HC SET--
05F1  FE 06 1002 R                  INC    COUNT                   ;( 8F   )
05F5  BA 0360                       MOV    DX,STATUS_REG           ;
05F8  03 D3                         ADD    DX,BX                   ;
05FA  EC                            IN     AL,DX                   ;GET THE STATUS REGISTER
05FB  A8 80                         TEST   AL,80H                  ;IS THE HC SET
05FD  75 0E                         JNZ    HIFT_FAILED             ;JMP  FAILURE HC IS ON

                                    ;--NOW CLEAR THE HIR---
05FF  FE 06 1002 R                  INC    COUNT                   ;( 90   )
0603  B0 00                         MOV    AL,00                   ;
0605  BA 0363                       MOV    DX,HOST_INTR_REG        ;
0608  03 D3                         ADD    DX,BX                   ;ADD OFFSET
060A  EE                            OUT    DX,AL                   ;CLEAR THE HIR REG
060B  F8                            CLC                            ;FLAG SET FOR GOOD RETURN
060C  C3                            RET                            ;

060D                         HIFT_FAILED:
                                    ;--CLEAR THE HIR---INSURE HCR IS OFF SO LANA CAN GET INTERFACE
060D  B0 00                         MOV    AL,00                   ;
060F  BA 0363                       MOV    DX,HOST_INTR_REG        ;
0612  03 D3                         ADD    DX,BX                   ;ADD OFFSET
0614  EE                            OUT    DX,AL                   ;CLEAR THE HIR REG
0615  F9                            STC                            ;FLAG SET FOR ERROR RETURN
0616  C3                            RET                            ;
0617                         HIFT_TEST         ENDP                 ;
                             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0617                         HOT_RF?           PROC    NEAR         ;
                                    ;LANA RETURNED A 8E ERROR CODE, INDICATING A POTENTIAL HOT CARRIER
                                    ;LOOK FOR THE GO BIT AND CMD 45.  LANA MAY TAKE UP TO 45 SECONDS TO
                                    ;RETURN A HOT CARRIER, IF AFTER 45 SECONDS NO CMD 45 THEN NORMAL EXIT.
                                    ;IF CMD 45 DETERMINE  IF HOT CARRIER ON THIS CARD OR
                                    ;ON THE NET. DISPLAY THE APPROIATE MESSAGE.


0617  E8 08F5 R                     CALL   START_TIMER             ;INITIALIZE THE TIME
061A  C6 06 1006 R 80               MOV    RF_TEST,80H             ;SET FLAG FOR 'TICKS PROC'
                                                                   ;INDICATES THAT HOT CARRIER TEST IN
                                                                   ;PROCESS
061F                         HR1:
061F  BA 0360                       MOV    DX,STATUS_REG           ;
0622  03 D3                         ADD    DX,BX                   ;OFFSET
0624  EC                            IN     AL,DX                   ;GET STATUS REG
0625  A8 01                         TEST   AL,01                   ;GO BIT ON?
0627  75 0A                         JNZ    HR2                     ;JMP YES

0629  80 3E 100F R 80               CMP    T_O_FLAG,80H            ;HAS 45 SEC'S ELASPED?
062E  75 EF                         JNZ    HR1                     ;JMP NO-CONTINUE TO LOOK
0630  EB 58 90                      JMP    HC8                     ;JMP TIME EXPIERED-EXIT NORMAL

0633                         HR2:
                                    ;GO BIT ON - CMD 45 ?

0633  BA 0361                       MOV    DX,PARAMETER_REG        ;
0636  03 D3                         ADD    DX,BX                   ;OFFSET
0638  EC                            IN     AL,DX                   ;
0639  3C 45                         CMP    AL,45H                  ;
063B  74 11                         JZ     HC5                     ;

                                    ;NOT CMD 45

                                    ;NEVER SHOULD GET HERE

063D  83 FB 00                      CMP    BX,0                    ;
0640  75 06                         JNZ    HR3                     ;
0642  BE 0205 R                     MOV    SI,OFFSET M13           ;'DIGITAL FAILURE'
0645  EB 45 90                      JMP    HC9                     ;ERROR

0648                         HR3:
0648  BE 0265 R                     MOV    SI,OFFSET M33           ; ' DIGITAL FAILURE'
064B  EB 3F 90                      JMP    HC9                     ;

064E                         HC5:
064E  EC                            IN     AL,DX                   ;GET SECOND BYTE OF PARAM REG
064F  3C 41                         CMP    AL,41H                  ;CC NOT ME?
0651  75 11                         JNZ    HC6                     ;

                                    ;CONTINUOUS CARRIER & ITS NOT ME  (3X41)
0653  83 FB 00                      CMP    BX,0                    ;FIRST CARD
0656  75 06                         JNZ    HC5A                    ;
0658  BE 0211 R                     MOV    SI,OFFSET M50           ;3041
065B  EB 2F 90                      JMP    HC9                     ;ERROR
065E                         HC5A:
065E  BE 0271 R                     MOV    SI,OFFSET M60           ;3141
0661  EB 29 90                      JMP    HC9                     ;ERROR EXIT

0664                         HC6:
0664  3C 42                         CMP    AL,42H                  ;CONTINUOUS CARRIER ME?
0666  75 11                         JNZ    HC7                     ;

                                    ;CONTINUOUS CARRIER AND ITS ME   (3X42)

0668  83 FB 00                      CMP    BX,0                    ;FIRST CARD?
066B  75 06                         JNZ    HC6A                    ;
066D  BE 0217 R                     MOV    SI,OFFSET M51           ;3042
0670  EB 1A 90                      JMP    HC9                     ;
0673                         HC6A:
0673  BE 0277 R                     MOV    SI,OFFSET M61           ;3142
0676  EB 14 90                      JMP    HC9                     ;

0679                         HC7:
                                    ;DMA FAILURE ?
0679  83 FB 00                      CMP    BX,0                    ;FIRST CARD
067C  75 06                         JNZ    HC7A                    ;
067E  BE 0205 R                     MOV    SI,OFFSET  M13          ;3013
0681  EB 09 90                      JMP    HC9                     ;
0684                         HC7A:
0684  BE 0265 R                     MOV    SI,OFFSET M33           ;3113
0687  EB 03 90                      JMP    HC9                     ;

068A                         HC8:
                                    ;NO HOT CARRIER EXIT
068A  F8                            CLC                            ;
```

```
068B C3                       RET                              ;
068C                   HC9:
                              ;ERROR EXIT
068C E8 01A4 R                CALL   DISPLAY                   ;
068F C6 06 1005 R 80          MOV    FAIL_FLAG,80H             ;80 = FAILURE OCCURED
0694 0E                       PUSH   CS                        ;
0695 5D                       POP    BP                        ;PUT THE CS ID IN BP
0696 F9                       STC                              ;FAILURE
0697 C3                       RET                              ;
0698                   HOT_RF?       ENDP                      ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0698                   INT_OK?       PROC    NEAR              ;TEST INTERRUPT STATUS FOR CONFLICTS
0698 80 3E 1014 R 00          CMP    LANA_1_ACTIVE,0           ;IS 2ND LANA INSTALLED AND ACTIVE ?
069D 74 0B                    JZ     IO_2                      ;JMP NO
069F                   IO_1:
069F A0 1013 R                MOV    AL,LANA_1_INT             ;
06A2 38 06 1012 R             CMP    LANA_0_INT,AL             ;ARE LANAS ON DIFFERENT INT LEVELS?
06A6 75 02                    JNZ    IO_2                      ;JMP YES
06A8 F9                       STC                              ;CARRY FLAG SET FOR ERROR RETURN
06A9 C3                       RET                              ;
06AA                   IO_2:
06AA F8                       CLC                              ;CARRY FLAG SET FOR GOOD RETURN
06AB C3                       RET                              ;
06AC                   INT_OK?       ENDP                      ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
06AC                   INT_2         PROC    NEAR              ;INT HANDLER FOR INT 2
06AC 52                       PUSH   DX                        ;
06AD 53                       PUSH   BX                        ;
06AE 1E                       PUSH   DS                        ;
06AF 50                       PUSH   AX                        ;
06B0 33 C0                    XOR    AX,AX                     ;REESTABLISH DS
06B2 8E D8                    MOV    DS,AX                     ;
06B4 C6 06 1000 R 80          MOV    INT_OCCUR?,80H            ;SET FLAG TO INDICATE A INT OCCURRED
06B9 C6 06 1011 R 04          MOV    TEMP_INT,04H              ;FLAG FOR INT 2 ACTIVE
06BE BA 0360                  MOV    DX,STATUS_REG             ;
06C1 03 D3                    ADD    DX,BX                     ;ADD OFFSET
06C3 EC                       IN     AL,DX                     ;GET THE STATUS REG
06C4 EB 00                    JMP    SHORT $ + 2               ;DELAY
06C6 0C 80                    OR     AL,RESET_INT_REQ          ;OR IN BIT TO TURN OF INT REQUESTS
06C8 EE                       OUT    DX,AL                     ;CLEAR INT REQUEST FROM LANA CARD
06C9 B0 20                    MOV    AL,EOI                    ;ENABLE GENERAL INTERRUPTS
06CB E6 20                    OUT    INTA00,AL                 ;
06CD 58                       POP    AX                        ;RESTORE REGS
06CE 1F                       POP    DS                        ;
06CF 5B                       POP    BX                        ;
06D0 5A                       POP    DX                        ;
06D1 CF                       IRET                             ;
06D2                   INT_2  ENDP                             ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
06D2                   INT_3         PROC    NEAR              ;INT HANDLER FOR INT 3
06D2 52                       PUSH   DX                        ;
06D3 53                       PUSH   BX                        ;
06D4 1E                       PUSH   DS                        ;
06D5 50                       PUSH   AX                        ;
06D6 33 C0                    XOR    AX,AX                     ;REESTABLISH DS
06D8 8E D8                    MOV    DS,AX                     ;
06DA C6 06 1000 R 80          MOV    INT_OCCUR?,80H            ;SET FLAG TO INDICATE A INT OCCURRED
06DF C6 06 1011 R 08          MOV    TEMP_INT,08H              ;FLAG INDICATING INT 3 ACTIVE
06E4 BA 0360                  MOV    DX,STATUS_REG             ;
06E7 03 D3                    ADD    DX,BX                     ;ADD OFFSET
06E9 EC                       IN     AL,DX                     ;GET THE STATUS REG
06EA EB 00                    JMP    SHORT $ + 2               ;DELAY
06EC 0C 80                    OR     AL,RESET_INT_REQ          ;OR IN BIT TO TURN OF INT REQUESTS
06EE EE                       OUT    DX,AL                     ;CLEAR INT REQUEST FROM LANA CARD
06EF B0 20                    MOV    AL,EOI                    ;ENABLE GENERAL INTERRUPTS
06F1 E6 20                    OUT    INTA00,AL                 ;
06F3 58                       POP    AX                        ;RESTORE REGS
06F4 1F                       POP    DS                        ;
06F5 5B                       POP    BX                        ;
06F6 5A                       POP    DX                        ;
06F7 CF                       IRET                             ;
06F8                   INT_3  ENDP                             ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
06F8                   LANA_PRESENT? PROC    NEAR              ;CALLED WITH HIR ADD IN DX
                                                               ;LANA PRESENT = NC **NOT PRESENT = CF
06F8 B1 04                    MOV    CL,04                     ;RESET BIT ,SAVE FOR LATER USE
06FA B8 FFFF                  MOV    AX,0FFFFH                 ;GARBAGE
06FD 50                       PUSH   AX                        ;
06FE 58                       POP    AX                        ;TOGGEL THE INTERFACE
06FF EC                       IN     AL,DX                     ;GET HIR REG
0700 3A C1                    CMP    AL,CL                     ;AL=04=PRESENT
0702 75 02                    JNZ    LP1                       ;JMP NOT PRESENT
0704 F8                       CLC                              ;ADAPTER PRESENT
0705 C3                       RET                              ;
0706                   LP1:
                              ;LANA NOT PRESENT
0706 F9                       STC                              ;
0707 C3                       RET                              ;
0708                   LANA_PRESENT? ENDP                      ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0708                   MASK_INT_2_3  PROC    NEAR              ;MASK INTERRUPTS
0708 E4 21                    IN     AL,INTA01                 ;GET IMR
070A 0C 0C                    OR     AL,MASK_IRQ2_3            ;MAKE SURE INT 2,3 ARE DISABLED
070C E6 21                    OUT    INTA01,AL                 ;TURN THEM OFF
070E C3                       RET                              ;
070F                   MASK_INT_2_3  ENDP                      ;
                       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
070F                   PC_ERROR      PROC    NEAR
070F 3C 8F                    CMP    AL,ANALOG_FAIL            ;ANALOG FAILURE=8F
0711 75 0F                    JNZ    PC_4                      ;JMP NOT ANALOG
0713                   PC_1:
                              ;ANALOG FAILURE
0713 83 FB 00                 CMP    BX,0                      ;PRIMARY CARD?
0716 75 05                    JNZ    PC_2                      ;JMP SECONDARY CARD
                              ;1ST CARD HAD ANALOG FAILURE
0718 BE 020B R                MOV    SI,OFFSET M15             ;"3015"
071B EB 66                    JMP    SHORT P_E3                ;DISPLAY IT
071D                   PC_2:
                              ;ANALOG FAILURE 2ND CARD
071D BE 026B R                MOV    SI,OFFSET M35             ;"3115"
0720 EB 61                    JMP    SHORT   P_E3              ;DISPLAY IT
0722                   PC_4:
                              ;DIGITAL FAILURE
0722 83 FB 00                 CMP    BX,0                      ;WAS IT THE PRIMARY CARD?
0725 75 05                    JNZ    P_E1                      ;JMP NO
0727 BE 01BD R                MOV    SI,OFFSET M1              ;POINT TO PRIMARY ERROR LIST
072A EB 03                    JMP    SHORT P_E2                ;
```

```
072C                 P_E1:
072C  BE 021D R              MOV     SI,OFFSET M21          ;POINT TO SECONDARY ERROR LIST
072F                 P_E2:
072F  3C 81                  CMP     AL,81H                 ;
0731  74 50                  JZ      P_E3                   ;PROCESSOR ERROR
0733  83 C6 06               ADD     SI,6                   ;3X01
0736  3C 82                  CMP     AL,82H                 ;
0738  74 49                  JZ      P_E3                   ;ROS FAILED
073A  83 C6 06               ADD     SI,6                   ;3X02
073D  3C 83                  CMP     AL,83H                 ;
073F  74 42                  JZ      P_E3                   ;ID MODULE FAILED
0741  83 C6 06               ADD     SI,6                   ;3X03
0744  3C 84                  CMP     AL,84H                 ;
0746  74 3B                  JZ      P_E3                   ;RAM FAILURE
0748  83 C6 06               ADD     SI,6                   ;3X04
074B  3C 85                  CMP     AL,85H                 ;
074D  74 34                  JZ      P_E3                   ;HIC FAILURE
074F  83 C6 06               ADD     SI,6                   ;3X05
0752  3C 86                  CMP     AL,86h                 ;
0754  74 2D                  JZ      P_E3                   ;+ - 12V FAILURE
0756  83 C6 06               ADD     SI,6                   ;3X06
0759  3C 87                  CMP     AL,87h                 ;
075B  74 26                  JZ      P_E3                   ;DIGITAL WRAP FAILURE
075D  83 C6 06               ADD     SI,6                   ;3X07

0760  3C 88                  CMP     AL,88h                 ;
0762  74 1F                  JZ      P_E3                   ;HOST DETECTED HIC FAILURE
0764  83 C6 06               ADD     SI,6                   ;3X08

0767  3C 89                  CMP     AL,89h                 ;
0769  74 18                  JZ      P_E3                   ;SYNC FAIL & NO GO BIT
076B  83 C6 06               ADD     SI,6                   ;3X09

076E  3C 8A                  CMP     AL,8Ah                 ;
0770  74 11                  JZ      P_E3                   ;HIC TEST OK & NEVER GOT GO BIT
0772  83 C6 06               ADD     SI,6                   ;3X10

0775  3C 8B                  CMP     AL,8Bh                 ;
0777  74 0A                  JZ      P_E3                   ;GO BIT & NO CMD 41
0779  83 C6 06               ADD     SI,6                   ;3X11

077C  3C 8C                  CMP     AL,8Ch                 ;
077E  74 03                  JZ      P_E3                   ;FAILED PRESENCE TEST  3X12

0780  83 C6 06               ADD     SI,6                   ;FELL THRU SHOULD NOT GET HERE  3X13

0783                 P_E3:
0783  E8 01A4 R              CALL    DISPLAY                ;

0786                 PC_7:
0786  0E                     PUSH    CS                     ;
0787  5D                     POP     BP                     ;ERROR ID   "F1" STOP
0788  C6 06 1005 R 80        MOV     FAIL_FLAG,80H          ;SET THE FAILED FLAG
078D  A0 1002 R              MOV     AL,COUNT               ;GET ERROR LOG
0790  BA 0362                MOV     DX,DATA_REG            ;
0793  EE                     OUT     DX,AL                  ;
0794  EE                     OUT     DX,AL                  ;2 BYTES TO DATA REGISTER

0795  C3                     RET                            ;

0796                 PC_ERROR        ENDP                   ;
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0796                 READ_REG        PROC            NEAR   ;ENTERED WITH BASE I/O ADD IN DX
0796  8A E0                  MOV     AH,AL                  ;& DATA TO BE COMPARED IN AL
0798  EC                     IN      AL,DX                  ;GET THE DATA
0799  3A E0                  CMP     AH,AL                  ;
079B  C3                     RET                            ;RETURN WITH FLAGS
079C                 READ_REG        ENDP
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
079C                 REMOVE_RESET0   PROC   NEAR            ;REMOVE RESET FROM LANA 0
079C  33 C0                  XOR     AX,AX                  ;REMOVE THE RESET BIT IN HIR/ AH=00
079E  BA 0363                MOV     DX,HOST_INTR_REG       ;
07A1  EE                     OUT     DX,AL                  ;REMOVE THE RESET FROM LANA 0

                             ;2ND RESET TO CARD
07A2  EB 00                  JMP     SHORT $ + 2            ;

07A4  B0 04                  MOV     AL,RESET               ;
07A6  EE                     OUT     DX,AL                  ;RESET LANA 0 2ND TIME
07A7  EB 00                  JMP     SHORT $ + 2            ;

07A9  32 C0                  XOR     AL,AL                  ;
07AB  EE                     OUT     DX,AL                  ;REMOVE RESET

07AC  C3                     RET                            ;
07AD                 REMOVE_RESET0   ENDP                   ;
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
07AD                 REMOVE_RESET1    PROC   NEAR           ;REMOVE THE RESET FROM LANA 1
                                                            ;LANA ADD 368-36B
07AD  33 C0                  XOR     AX,AX                  ;REMOVE THE RESET BIT IN HIR/ AH=00
07AF  BA 0363                MOV     DX,HOST_INTR_REG       ;
07B2  03 D3                  ADD     DX,BX                  ;ADD OFFSET FOR I/O ADDRESS
07B4  EE                     OUT     DX,AL                  ;REMOVE THE RESET FROM LANA 1
07B5  EB 00                  JMP     SHORT $ + 2            ;

                             ;2ND RESET TO CARD

07B7  B0 04                  MOV     AL,RESET               ;
07B9  EE                     OUT     DX,AL                  ;RESET LANA 1 2ND TIME
07BA  EB 00                  JMP     SHORT $ + 2            ;

07BC  32 C0                  XOR     AL,AL                  ;
07BE  EE                     OUT     DX,AL                  ;REMOVE RESET

07BF  C3                     RET                            ;
07C0                 REMOVE_RESET1   ENDP                   ;
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
07C0                 RESTORE_TICK_VECTOR     PROC    NEAR   ;
07C0  FA                     CLI                            ;NO INTERRUPTS
07C1  A1 1009 R              MOV     AX,SAVE_TICK_INT       ;GIT THE ORIGINAL VECTOR INFO
07C4  A3 0070 R              MOV     TICK_INT,AX            ;RESTORE
07C7  A1 100B R              MOV     AX,SAVE_TICK_INT +2    ;
07CA  A3 0072 R              MOV     TICK_INT +2 ,AX        ;RESTORE
07CD  FB                     STI                            ;
07CE  C3                     RET                            ;
07CF                 RESTORE_TICK_VECTOR     ENDP           ;
                     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
07CF                 RESTORE_INT2_3    PROC          NEAR   ;RESTORE INT 2 & 3

07CF  FA                     CLI                            ;
                             ;RESTORE LEVEL 2

07D0  A1 1016 R              MOV     AX,SAVE_INT2           ;GET ORIGINAL OFFSET LVL 2
```

```
07D3  26: A3 0028               MOV   ES:INT2_VECT,AX        ;RESTORE IT
07D7  A1 1018 R                 MOV   AX,SAVE_INT2+2         ;GET SEGMENT
07DA  26: A3 002A               MOV   ES:INT2_VECT+2,AX      ;RESTORE IT
07DE                      RI1:
                                ;RESTORE LEVEL 3
07DE  A1 101A R                 MOV   AX,SAVE_INT3           ;GET ORIGINAL OFFSET LVL 3
07E1  26: A3 002C               MOV   ES:INT3_VECT,AX        ;RESTORE IT
07E5  A1 101C R                 MOV   AX,SAVE_INT3+2         ;GET SEG
07E8  26: A3 002E               MOV   ES:INT3_VECT+2,AX      ;RESTORE IT
                                ;RESTORE INT MASKS
07EC  A0 1007 R                 MOV   AL,SAVE_MASK           ;GET ORIGINAL IMR MASK IST CHIP
07EF  E6 21                     OUT   INTA01,AL              ;RESTORE IT
07F1  80 3E 1010 R FC           CMP   PC_ID,PC3              ;IS THIS A PC 3
07F6  75 05                     JNZ   RI2                    ;JMP NO
07F8  A0 1008 R                 MOV   AL,SAVE_MASKA          ;GET ORIGINAL MASK FOR 2ND INT CHIP
07FB  E6 A1                     OUT   INTB01,AL              ;RESTORE IT
07FD                      RI2:
07FD  FB                        STI                          ;
07FE  C3                        RET                          ;
07FF                      RESTORE_INT2_3  ENDP               ;
                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
07FF                      SAVE_INT_VECT   PROC          NEAR ;SAVE POST VECTORS FOR INT 2 & 3
                                                             ;INT VECTORS WILL BE RESTORED ON EXIT
                                                             ;TO POST
                                ;INT 2 SAVE
07FF  26: A1 0028               MOV   AX,ES:INT2_VECT        ;GET INT 2 OFFSET
0803  A3 1016 R                 MOV   SAVE_INT2,AX           ;SAVE IT
0806  26: A1 002A               MOV   AX,ES:INT2_VECT+2      ;GET SEG
080A  A3 1018 R                 MOV   SAVE_INT2+2,AX         ;SAVE IT
                               ;INT 3 SAVE
080D  26: A1 002C               MOV   AX,ES:INT3_VECT        ;GET INT 3 OFFSET
0811  A3 101A R                 MOV   SAVE_INT3,AX           ;SAVE IT
0814  26: A1 002E               MOV   AX,ES:INT3_VECT+2      ;GET SEG
0818  A3 101C R                 MOV   SAVE_INT3+2,AX         ;SAVE IT
081B  C3                        RET                          ;
081C                      SAVE_INT_VECT   ENDP               ;
                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
081C                      SET_BIOS_INTS   PROC          NEAR ;A CARD TESTED OK,SET OPERATING SYSTEM
                                                             ;INTS,18,5C,LANA0,LANA1,13
081C  FA                        CLI                          ;
                                ;MOVE INT 18(IPL) TO 86 (BASIC)
081D  A1 0060 R                 MOV   AX,INT_18              ;GET ORIGINAL OFFSET FOR INT 18
0820  26: A3 0218               MOV   ES:INT86_VECT,AX       ;MOVE IT TO BASIC VECTOR  86
0824  A1 0062 R                 MOV   AX,INT_18 +2           ;GET ORIGINAL SEGMENT
0827  26: A3 021A               MOV   ES:INT86_VECT+2,AX     ;MOV IT TO 86 +2
                                ;NOW POINT 18 TO BIOS HANDLER
082B  C7 06 0060 R 0000 E       MOV   INT_18,OFFSET REM_IPL  ;OFFSET OF BIOS HANDLER TO 18
0831  8C C8                     MOV   AX,CS                  ;GET CODE SEG
0833  A3 0062 R                 MOV   INT_18+2,AX            ;MOV SEG TO 18
                                ;NOW POINT THE SOFT VECTOR ( INT 5C) TO LANA BIOS
0836  FA                        CLI                          ;
0837  C7 06 0170 R 0000 E       MOV   LANA_BIOS_INT,OFFSET MAIN  ;
083D  8C C8                     MOV   AX,CS                  ;
083F  A3 0172 R                 MOV   LANA_BIOS_INT + 2,AX   ;
                                ;SET INT LEV INFO FOR BIOS
                                ;LANA_X_STATUS BIT 4=0=INT 2,BIT 4=1=INT 3
                                ;IS LANA0  ACTIVE?
0842  26: F6 06 04A2 80         TEST  BYTE PTR ES:LANA_0_STATS,80H       ;LANA0 ACTIVE ?
0848  74 0D                     JZ    SBI_2                  ;JMP NOT ACTIVE
                                ;SET LANA 0 INT FLAG FOR BIOS
084A  80 3E 1012 R 04           CMP   LANA_0_INT,04          ;IS LANA 0 ON LEVEL 2 ?
084F  74 06                     JZ    SBI_2                  ;JMP YES, LEAVE BIT 04 = 0
0851                      SBI_1:
                                ;SET LANA 0 FOR LEVEL 3
0851  26: 80 0E 04A2 10         OR  BYTE PTR ES:LANA_0_STATS, INT3_FLAG
0857                      SBI_2:
                                ;ANY INT LEVEL FOR LANA 1?
0857  26: F6 06 04A3 80         TEST  BYTE PTR ES:LANA_1_STATS,80H       ;LANA 1 ACTIVE ?
085D  74 0D                     JZ    SBI_3A                 ;JMP NO
085F  80 3E 1013 R 04           CMP   LANA_1_INT,04          ;LEVEL 2 ACTIVE ON LANA 1 ?
0864  74 06                     JZ    SBI_3A                 ;JMP YES ,LEAVE BIT 4=0
0866                      SBI_3:
                                ;SET LANA 1 FOR INT 3
0866  26: 80 0E 04A3 10         OR  BYTE PTR ES:LANA_1_STATS, INT3_FLAG
086C                      SBI_3A:
                                ;MOVE THE HARD FILE INT VECTOR TO
                                ;LANA RESERVED LO MEMORY
086C  A1 004C R                 MOV   AX,HARD_FILE_BIOS      ;GET OFFSET
086F  26: A3 04A4               MOV   ES:LANA_1_STATS + 1,AX ;MOV TO LANA AREA
0873  A1 004E R                 MOV   AX,HARD_FILE_BIOS +2   ;GET CS
0876  26: A3 04A6               MOV   ES:LANA_1_STATS + 3,AX ;MOV TO LANA AREA
                                ;ARE WE ON A PC 3
087A  80 3E 1010 R FC           CMP   PC_ID,0FCH             ;FC=3
087F  74 0B                     JZ    SBI_4                  ;JMP YES
                                ;REDIRECT HARD FILE SOFT INT 13
                                ;CALLS TO DISKETTE AND HARD FILE DONE THRU 13
0881  C7 06 004C R 0000 E       MOV   HARD_FILE_BIOS,OFFSET HARD_FILE ;INT13  HANDLER
0887  8C C8                     MOV   AX,CS                  ;
0889  A3 004E R                 MOV   HARD_FILE_BIOS + 2,AX ;CODE SEG
088C                      SBI_4:
088C  FB                        STI                          ;
088D  C3                        RET                          ;
088E                      SET_BIOS_INTS   ENDP               ;
                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
088E                      SET_UP_TIME_TICK     PROC    NEAR  ;SAVE TIMER TICK INT VECTOR AND
```

```
                                                                        ;REDIRECT THE VECTOR TO LANA
                                             ;--SAVE BIOS VECTOR 1C--

 088E  33 C0                                 XOR    AX,AX                ;
 0890  A3 100D R                             MOV    TICKS,AX             ;CLEAR THE TICK COUNT
 0893  A2 100F R                             MOV    T_O_FLAG,AL          ;CLEAR THE TIME OUT FLAG
 0896  26: A1 0070 R                         MOV    AX,ES:TICK_INT       ;GET BIOS OFFSET FOR 1C
 089A  A3 1009 R                             MOV    SAVE_TICK_INT,AX     ;SAVE IT, PGM WILL RESTORE ON EXIT
 089D  26: A1 0072 R                         MOV    AX,ES:TICK_INT +2    ;GET THE BIOS CS FOR INT 1C
 08A1  A3 100B R                             MOV    SAVE_TICK_INT +2,AX  ;SAVE IT

                                      ;REDIRECT TIMER TICK VECTOR TO ME

 08A4  FA                                    CLI                         ;DONT  LET ANY INT IN
 08A5  C7 06 0070 R 0906 R                   MOV    TICK_INT,OFFSET TICK_IT     ;OFFSET OF LANA 1C HANDLER
 08AB  8C C8                                 MOV    AX,CS                ;LANA CS
 08AD  A3 0072 R                             MOV    TICK_INT +2,AX       ;1C NOW POINTS TO LANA
 08B0  FB                                    STI                         ;LET INT IN
 08B1  C3                                    RET                         ;
 08B2                                 SET_UP_TIME_TICK       ENDP        ;

                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 08B2                                 SET_VECT_2_3    PROC    NEAR       ;
 08B2  FA                                    CLI                         ;
 08B3  26: C7 06 0028 06AC R                 MOV    ES:INT2_VECT,OFFSET INT_2  ;INT HANDLER FOR LEVEL 2
 08BA  26: 8C 0E 002A                        MOV    ES:INT2_VECT + 2,CS        ;

 08BF  26: C7 06 002C 06D2 R                 MOV    ES:INT3_VECT,OFFSET INT_3  ;INT HANDLER FOR LEVEL 3
 08C6  26: 8C 0E 002E                        MOV    ES:INT3_VECT + 2,CS        ;
 08CB  FB                                    STI                         ;
 08CC  C3                                    RET                         ;
 08CD                                 SET_VECT_2_3    ENDP               ;

                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 08CD                                 SET_WD_CNT      PROC    NEAR       ;SET DMA 3 FOR 2 BYTE XFER
 08CD  E6 0C                                 OUT    DMA + 0CH,AL         ; CLEAR F/F..AL DOSN"T MATTER
 08CF  EB 00                                 JMP    SHORT $ + 2          ;DELAY
 08D1  B0 01                                 MOV    AL,01                ;COUNT N-1
 08D3  E6 07                                 OUT    DMA + 7,AL           ;WRITE THE LSB
 08D5  EB 00                                 JMP    SHORT $ + 2          ;DELAY
 08D7  32 C0                                 XOR    AL,AL                ;MSB
 08D9  E6 07                                 OUT    DMA + 7,AL           ; WORD COUNT 0002
 08DB  C3                                    RET                         ;
 08DC                                 SET_WD_CNT      ENDP               ;

                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 08DC                                 SPECIAL_CLR     PROC    NEAR       ;
 08DC  FA                                    CLI                         ;
 08DD  BA 0363                               MOV    DX,HOST_INTR_REG
 08E0  03 D3                                 ADD    DX,BX
 08E2  EC                                    IN     AL,DX                ;GET CURRENT REG
 08E3  24 0C                                 AND    AL,00001100B         ;MASK OF INT BITS,DMA ENABLE,HCR
 08E5  EB 00                                 JMP    SHORT $ + 2
 08E7  EE                                    OUT    DX,AL                ;SET HIR

 08E8  BA 0360                               MOV    DX,STATUS_REG        ;
 08EB  03 D3                                 ADD    DX,BX                ;
 08ED  EC                                    IN     AL,DX                ;GET STATUS REG
 08EE  EB 00                                 JMP    SHORT $ + 2          ;DELAY
 08F0  0C 80                                 OR     AL,RESET_INT_REQ     ;OR IN BIT TO CLEAR INT REQUESTS
 08F2  EE                                    OUT    DX,AL                ;CLEAR INT REQ
 08F3  FB                                    STI                         ;
 08F4  C3                                    RET                         ;
 08F5                                 SPECIAL_CLR     ENDP               ;

                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 08F5                                 START_TIMER     PROC    NEAR       ;ENABLE FLAG FOR GROSS TIME OUT
                                                                         ;RESETS TIME COUNT VALUES AND FLAGS
 08F5  33 C0                                 XOR    AX,AX                ;FLAG TESTED IN TICKS
 08F7  A3 100D R                             MOV    TICKS,AX             ;CLEAR THE TIME COUNT
 08FA  A2 100F R                             MOV    T_O_FLAG,AL          ;CLEAR THE FLAG
 08FD  C3                                    RET                         ;
 08FE                                 START_TIMER     ENDP               ;

                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 08FE                                 STOP_TIMER      PROC    NEAR       ;DISABLE FLAG FOR GROSS TIME OUT
 08FE  FA                                    CLI                         ;DISABLE INT
 08FF  33 C0                                 XOR    AX,AX                ;
 0901  A3 100D R                             MOV    TICKS,AX             ;TIMER COUNT TO ZERO
 0904  FB                                    STI                         ;
 0905  C3                                    RET                         ;
 0906                                 STOP_TIMER      ENDP               ;

                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 0906                                 TICK_IT         PROC    FAR        ;HANDLES THE TIMER TICK INT 1C
 0906  50                                    PUSH   AX                   ;SAVE SEGMENT REGS
 0907  1E                                    PUSH   DS                   ;

 0908  33 C0                                 XOR    AX,AX                ;
 090A  8E D8                                 MOV    DS,AX                ;ESTABLISH DS

 090C  FF 06 100D R                          INC    TICKS                ;

                                             ;IS HOT RF TEST IN PROCESS?

 0910  80 3E 1006 R 80                       CMP    RF_TEST,80H          ;80=SUSPECT HOT RF
 0915  75 0A                                 JNZ    TI0                  ;JMP NO HOT RF TEST

                                             ;HOT RF TEST-HAS 45 SECONDS EXPIRED

 0917  81 3E 100D R 0333                     CMP    TICKS,RF_COUNT       ;HAS 45 SECONDS EXPIRED
 091D  77 0C                                 JA     TI1                  ;JMP TIME EXPIRED -SET FLAG

 091F  EB 12                                 JMP    SHORT TI2            ;EXIT-TIME HAS NOT EXPIRED

 0921                                 TI0:
                                             ;GENERAL TIME OUT TEST
 0921  81 3E 100D R 0222                     CMP    TICKS,T_O_CNT        ;30 SEC'S ELASPED?
 0927  77 02                                 JA     TI1                  ;JMP YES-SET TIME OUT FLAG

 0929  EB 08                                 JMP    SHORT TI2            ;EXIT-TIME HAS NOT EXPIRED

 092B                                 TI1:
                                             ;TIME HAS EXPIRED SET FLAG
 092B  C6 06 100F R 80                       MOV    T_O_FLAG,80H         ;SET TIME OUT INDICATOR
 0930  A3 100D R                             MOV    TICKS,AX             ; ZERO THE TICK COUNT

 0933                                 TI2:
                                             ;EXIT
 0933  B0 20                                 MOV    AL,EOI               ;ENABLE GENERAL INTERRUPTS
 0935  E6 20                                 OUT    INTA00,AL            ;

 0937  1F                                    POP    DS                   ;RESTORE SEGS
 0938  58                                    POP    AX                   ;
 0939  CF                                    IRET                        ;
 093A                                 TICK_IT         ENDP               ;

                                      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 093A                                 TIME_OUT_OR_INT?    PROC        NEAR   ;TEST FOR A GROSS TIME OUR OR INT
                                                                         ;& RETURN WITH STATUS IN AH
                                                                         ;&AH=80=INT**AH=08=T_O**AH=0=NEITHER

 093A  32 E4                                 XOR    AH,AH                ;
```

```
093C  80 3E 100F R 80           CMP     T_O_FLAG,80H                ;
0941  75 04                     JNZ     TOOI1                       ;JMP NO GROSS TIME OUT
0943  80 CC 08                  OR      AH,08H                      ;SET TIME OUT FLAG
0946  C3                        RET
0947                    TOOI1:
0947  80 3E 1000 R 80           CMP     INT_OCCUR?,80H              ;
094C  74 03                     JZ      TOOI2                       ;JMP INT OCCUR
094E  32 E4                     XOR     AH,AH                       ;AH=00
0950  C3                        RET                                 ;
0951                    TOOI2:
0951  80 CC 80                  OR      AH,80H                      ;SET INT FLAG
0954  C3                        RET                                 ;AH=80
0955                    TIME_OUT_OR_INT?        ENDP                ;
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
0955                    WRT_REG         PROC            NEAR        ;ENTERED WITH I/O ADD IN DX, DATA IN AL
0955  EE                        OUT     DX,AL                       ;
0956  C3                        RET                                 ;
0957                    WRT_REG ENDP                                ;
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


0957                    NETWORK                 ENDS
                                        END     START
                TITLE  NETBIOS MAIN
                        ;
                        ; NETBIOS.ASM
                        ;
                        ;    NETBIOS MAIN SOURCE.

                ;____________________________________________________________________________
                ;                                                                            |
                ;       MODULE : NETWORK                                                     |
                ;     COMPONENT : NETWORK BIOS                                               |
                ;                                                                            |
                ;                                                                            |
                ;          NETBIOS IS THE NETWORK BIOS. IT RESIDES IN A ROM.                 |
                ;     NETWORK SERVICES  ARE THEN  PROVIDED VIA THE INT NET_INT INSTRUCTION.  |
                ;     SERVICES PROVIDED ARE:                                                 |
                ;                                                                            |
                ;           RESET              - RESET AND RECONFIGURE NETWORK ADAPTER CARD  |
                ;           ADAPTER STATUS     - RECEIVE LOCAL OR REMOTE ADAPTER STATUS      |
                ;           CANCEL             - CANCEL THE COMMAND STARTED BY THIS NCB      |
                ;           ADD_NAME           - ADD NAME TO NAME TABLE                      |
                ;           ADD_GROUP_NAME     - ADD A GROUP NAME TO NAME TABLE              |
                ;           DELETE_NAME        - DELETE NAME FROM NAME TABLE                 |
                ;           CALL               - OPEN A SESSION WITH A REMOTE OR LOCAL NAME  |
                ;           LISTEN             - WAIT FOR NAME/ANYONE TO OPEN SESSION TO YOU |
                ;           HANG_UP            - CLOSE A SESSION WITH THE GIVEN NAME         |
                ;           SEND               - SEND DATA THROUGH A SPECIFIC SESSION        |
                ;           CHAIN_SEND         - SEND TWO SEPARATE BUFFERS                   |
                ;           RECEIVE            - RECEIVE DATA FROM A SPECIFIC SESSION        |
                ;           RECEIVE_ANY        - RECEIVE DATA FROM ANY SESSION SENDING TO YOU|
                ;           SESSION STATUS     - RECEIVE STATUS OF ALL ACTIVE SESSIONS       |
                ;                                FOR YOUR NAME                               |
                ;           SEND_DATAGRAM      - SEND DATAGRAM TO NAME                       |
                ;           RECEIVE_DATAGRAM   - RECEIVE DATAGRAM FROM NAME                  |
                ;           SEND_BROADCAST DATAGRAM     - SEND BROADCAST DATAGRAM TO         |
                ;                                         EVERYONE                           |
                ;           RECEIVE_BROADCAST DATAGRAM  - RECEIVE BROADCAST DATAGRAM FROM    |
                ;                                         ANYONE                             |
                ;                                                                            |
                ;     THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                           |
                ;                                                                            |
                ;        REGISTER    |   ACCESS  |                 USAGE                     |
                ;        ------------+-----------+---------------------------------------    |
                ;        ES:BX       |   CONST   | ADDRESS OF NCB TO PROCESS                 |
                ;           AL       |   RESULT  | NETBIOS RETURN CODE                       |
                ;                                                                            |
                ;                                                                            |
                ;       NETBIOS.LIB CONTAINS THE NETBIOS INTERFACE EQUATES AND STRUCTURES    |
                ;       LANAS.INC CONTAINS THE LANA INTERFACE EQUATES AND STRUCTURES         |
                ;                                                                            |
                ;____________________________________________________________________________|

0000            NETWORK    SEGMENT    PARA PUBLIC    'CODE'

                           ASSUME     CS:NETWORK
                           ASSUME     DS:NOTHING
                           ASSUME     SS:NOTHING
                           ASSUME     ES:NOTHING

                           EXTRN      LANA_0_HNDLR   : NEAR        ; LANA 0'S INTERRUPT HANDLER
                           EXTRN      PROGRAM_IO_CHUNK : NEAR      ;   LANA 0'S CODE TO DO PROGRAM I/O
                           EXTRN      LANA_1_HNDLR   : NEAR        ; LANA 1'S INTERRUPT HANDLER

                           PUBLIC     DMA_START_UP    ; NEAR       ; COMMON RTN TO STARTUP DMA XFER'S
                           PUBLIC     MAIN            ; NEAR       ; INT 5C ENTRY POINT
                           PUBLIC     HARD_FILE       ; NEAR       ; CHECK TO SHARE DMA WITH HARD FILE
                           PUBLIC     REM_IPL         ; NEAR       ; CHECK IF REMOTE IPL REQUESTED

                .LIST

= 00FF          NCBNOT_DONE?      EQU      0FFH              ; NCB IS NOT DONE YET

= 00FF          ALL_BITS          EQU      0FFH


= 0000          CONFIG_CPLT_TO    EQU      0000H
= 0000          RESET_CPLT_TO     EQU      0000H               ; HOW LONG TO WAIT FOR RESET TO COMPLETE
= 0000          CANCEL_CPLT_TO    EQU      0000H               ; HOW LONG TO WAIT FOR CANCEL TO COMPLETE
= 0000          DRE_LIMIT         EQU      0000H               ; HOW LONG TO WAIT FOR DATA REGISTER
= 0000          GO_LIMIT          EQU      0000H               ; HOW LONG TO WAIT FOR GO BIT TO CLEAR
= 0000          DMA_LIMIT         EQU      0000H               ; HOW LONG TO WAIT FOR DMA TO COMPLETE
= 0000          HC_LIMIT          EQU      0000H               ; HOW LONG TO WAIT FOR HC TO BE SET
= 0000          W_TIL_ACC         EQU      0000H


= 00FB          ENABLE_INT2       EQU      0FBH                ; USE FOR IMR TO ENABLE INT REQ 2
= 00F7          ENABLE_INT3       EQU      0F7H                ; USE TO ENABLE INT 3
= 00FD          ENABLE_INT9       EQU      0FDH                ; FOR INTERUPT 2 NEED TO SET INT 9
= 0021          INTA01            EQU      021H                ; DMA CHIP 1
= 00A1          INTB01            EQU      0A1H                ; DMA CHIP 2
= 0028          INTR_2            EQU      028H                ; PHYSICAL ADDRESS FOR INT VECTOR 2 (A*4)
= 002C          INTR_3            EQU      02CH                ; PHYSICAL ADDRESS FOR INT VRCTOR 3 (B*4)
= 000E          PC_ID_ADD         EQU      0EH                 ; OFFSET FOR PC ID
= 00FC          PC3               EQU      0FCH                ; PC3 ID
= FFFF          ROS_CODE          EQU      0FFFFH              ; POINTER INTO ROS

= 9080          DEV_BUSY_WAIT     EQU       9080H
= 9180          DEV_BUSY_POST     EQU       9180H

= 0013          HARD_INT          EQU       013H             ; DISK I/O
= 004C          HARD_INT@         EQU       HARD_INT*4       ; OFFSET FOR THIS INTERUPT

= 0018          INT_18H           EQU       18H              ; ROM BASIC
```

```
= 0060                    ROM_BASIC@        EQU       INT_18H*4        ; PHYSICAL ADDRES OF INT 18H
= 0086                    INT_86H           EQU       86H              ;
= 0218                    INT_STORAGE       EQU       INT_86H*4        ; PHYSICAL ADDRES OF INT 86H TO STORE VALUE OF INT
                           18H
= 0000                    INTERRUPT_VECTOR_SEGMENT  EQU 0000H          ; SEGMENT POINTER TO LOW MEMORY

                                   ;REMOTE IPL VALUES

0000  2A                  LOCAL_STAT        DB        "*"              ; USE TO GET LOCAL ADAPTER STATUS
0001  49 42 4D 4E 45 54   SERVER_NAME       DB        "IBMNETBOOT"     ; USE BY RPL ROUTIONE TO GET SESSION STABLISHED WI
                          TH RPLS
      42 4F 4F 54
= 0013                    LST_4_BYTE        EQU       0013H
= 0013                    MEMORY_SIZE       EQU       0013H            ; LOCATION FOR MEMORY SIZE IN K-BYTES
000B  00 7C 00 00         BOOT_LOCN         DD        00007C00H        ; LOCATION WHERE TO LOAD BOOT_RECORD
000F  0400                T1K               DW        0400H            ; MULTIPLIER FOR 1K BYTES

                                   ; VALID COMMANDS TABLE OFFH INDICATES NOT VALID COMMAND
                                   ; ANY OTHER VALUE INDICATES VALID COMMAND
                                   ; THIS VALUE IS USED TO INDICATE HOW MUCH OF THE NCB NEEDS
                                   ; BE SENT OVER TO LANA.

                          ;                     0       1       2       3       4       5       6       7
                          ;                     8       9       A       B       C       D       E       F

0011  FF FF FF FF FF FF   COMMAND_TBL  DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0019  FF FF FF FF FF FF   COMMAND_08   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0021  2C 2C 03 FF 0A 0A   COMMAND_10   DB      44,     44,      3,   OFFH,     10,     10,     10,     16
      0A 10
0029  FF FF FF FF FF FF   COMMAND_18   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0031  1A 1A 0A 1A FF FF   COMMAND_20   DB      26,     26,     10,     26,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0039  FF FF FF FF FF FF   COMMAND_28   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0041  2A 2A 2C 1A 2A 00   COMMAND_30   DB      42,     42,     44,     26,     42,      0,     42,   OFFH
      2A FF
0049  FF FF FF FF FF FF   COMMAND_38   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0051  FF FF FF FF FF FF   COMMAND_40   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0059  FF FF FF FF FF FF   COMMAND_48   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0061  FF FF FF FF FF FF   COMMAND_50   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0069  FF FF FF FF FF FF   COMMAND_58   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0071  FF FF FF FF FF FF   COMMAND_60   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0079  FF FF FF FF FF FF   COMMAND_68   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0081  00 FF FF FF FF FF   COMMAND_70   DB       0,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0089  FF FF FF FF FF FF   COMMAND_78   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF

                          ;                     0       1       2       3       4       5       6       7
                          ;                     8       9       A       B       C       D       E       F
0091  FF FF FF FF FF FF   COMMAND_80   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0099  FF FF FF FF FF FF   COMMAND_88   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00A1  2C 2C 03 FF 0A 0A   COMMAND_90   DB      44,     44,      3,   OFFH,     10,     10,     10,     16
      0A 10
00A9  FF FF FF FF FF FF   COMMAND_98   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00B1  1A 1A 0A 1A FF FF   COMMAND_A0   DB      26,     26,     10,     26,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00B9  FF FF FF FF FF FF   COMMAND_A8   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00C1  2A 2A FF 1A 2A FF   COMMAND_B0   DB      42,     42,   OFFH,     26,     42,   OFFH,     42,   OFFH
      2A FF
00C9  FF FF FF FF FF FF   COMMAND_B8   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00D1  FF FF FF FF FF FF   COMMAND_C0   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00D9  FF FF FF FF FF FF   COMMAND_C8   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00E1  FF FF FF FF FF FF   COMMAND_D0   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00E9  FF FF FF FF FF FF   COMMAND_D8   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00F1  FF FF FF FF FF FF   COMMAND_E0   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
00F9  FF FF FF FF FF FF   COMMAND_E8   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0101  FF FF FF FF FF FF   COMMAND_F0   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF
0109  FF FF FF FF FF FF   COMMAND_FF   DB    OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH,   OFFH
      FF FF


                          ;______________________________________________________________________________
                          ; MAIN                                                                         |
                          ;                                                                              |
                          ;    HANDLES INT NET_INT INSTRUCTIONS.                                         |
                          ;                                                                              |
                          ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                              |
                          ;                                                                              |
                          ;        REGISTER   |   ACCESS  |                USAGE                         |
                          ;        -----------+-----------+-------------------------------------------   |
                          ;        ES:BX      |   CONST   | ADDRESS OF NCB TO PROCESS                    |
                          ;          AL       |   RESULT  | NETBIOS RETURN CODE:                         |
                          ;                               |     SEE NETBIOS INTERFACE SPEC               |
                          ;                                                                              |
                          ; INTERRUPTS WILL BE MASKED BY EXECUTING THE INT INSTRUCTION.                  |
                          ;                                                                              |
                          ;______________________________________________________________________________|

0111                      MAIN        PROC        NEAR                        ;*; NETBIOS MAIN HANDLER ENTRY


                                      SAVE        <DS,CX,DX,SI,DI>
0111  1E              +                  PUSH     DS
0112  51              +                  PUSH     CX
0113  52              +                  PUSH     DX
0114  56              +                  PUSH     SI
0115  57              +                  PUSH     DI
```

```
                                        ; ESTABLISH GLOBAL ASSUMPTIONS

0116  FC                                CLD                                     ; STRING GO UP
0117  B8 0040                           MOV       AX,LO_MEM_SEG                 ; POINT TO LO_MEM_SEG LATER
011A  8E D8                             MOV       DS,AX                         ; DS PTS TO LANA LO_MEM BYTES

                                        ; DEFAULT RETURN CODE IS NCBGOOD_RET?

011C  B4 00                             MOV       AH,NCBGOOD_RET?

                                        ; FLAG NCB AS "NOT DONE"

011E  26: C6 47 01 FF                   MOV       ES:BYTE PTR [BX].NCB_RETCODE,NCBNOT_DONE?
0123  26: C6 47 31 FF                   MOV       ES:BYTE PTR [BX].NCB_CMD_CPLT,NCBNOT_DONE?

                                        ; CHECK FOR ANY IMMEDIATE ERRORS (& GET ADAPTOR PARMS)

0128  26: 8A 0F                         MOV       CL,ES:[BX].NCB_COMMAND        ;SAVE COMMAND VALUE FOR LATER USE
012B  E8 01A5 R                         CALL      CHK_STATE                     ;INITIALIZE CARD IF IT NEEDS TO BE ,AND
                                                                                ;CHECK FOR ERRORS
012E  80 FC 00              NO_CHK:     CMP       AH,NCBGOOD_RET?               ;DID EVERY THING GO OK
0131  74 04                             JE        CONFIG?                       ;YES GO AND PERFORM COMMAND
0133  FB                                STI                                     ;ENABLE INTERUPTS
0134  EB 5F 90                          JMP       BIOS_EXIT                     ;LEAVE AND REPORT ERROR


0137  80 F9 32              CONFIG?:    CMP       CL,NCBRESET                   ;IS IT A RESET COMMAND
013A  75 06                             JNE       CANCEL?                       ;NO GO AND CHECK FOR CANCEL
013C  E8 02BE R                         CALL      CONFIG                        ;GO AND RESET AND RECONFIGURE CARD
013F  EB 54 90                          JMP       BIOS_EXIT                     ;DONE WITH RESET THEN LEAVE


                                        ; "CANCEL" CMD?

0142  80 F9 35              CANCEL?:    CMP       CL,NCBCANCEL                  ;IS IT A CANCEL COMMAND
0145  75 06                             JNE       RDRCT_DIO                     ;NO GO AND CHECK FOR UNLINK COMMAND
0147  E8 03CB R                         CALL      CANCEL_CMD                    ;PERFORM CANCEL COMMAND
014A  EB 49 90                          JMP       BIOS_EXIT                     ;DONE WITH CANCEL THEN LEAVE

                                        ; IS IT CHANGE DISK I/O REDIRECTION

014D                        RDRCT_DIO:
014D  80 F9 70                          CMP       CL,TGGL_RDRC                  ;IS IT UNLINK COMMAND
0150  75 06                             JNE       OTHER_CMD                     ;NO GO NAD CHECK FOR OTHER COMMANDS
0152  E8 08BA R                         CALL      CHG_RDRC                      ;PERFORM UNLINK COMMAND
0155  EB 3E 90                          JMP       BIOS_EXIT                     ;DONE WITH UNLINK THEN LEAVE


                                        ; IT IS A NORMAL_CMD.

0158  E8 036B R             OTHER_CMD:  CALL      NORMAL_CMD                    ;GO AND PERFORM ANY OF THE OTHER COMMANDS

                                        ; ANY ERRORS?

015B  80 FC 00              BIOS_CHK:   CMP       AH,NCBGOOD_RET?               ;ANY IMMEDIATE ERRORS TO REPORT
015E  75 35                             JNE       BIOS_EXIT                     ;GO REPORT ERROR AND LEAVE

                                        ; NOPE.  IS IT A WAIT CASE?

0160  F6 C1 80                          TEST      CL,NCBNO_WAIT                 ;IS IT A NO_WAIT COMMAND
0163  75 38                             JNZ       BIOS_EXIT1                    ; IF IT IS LEAVE

                                        ; YES.  WELL THEN...  WAIT.

0165  B8 9080                           MOV       AX,DEV_BUSY_WAIT              ;INDICATE THAT WE ARE GOING INTO A WAIT LOOP
0168  CD 15                             INT       15H
016A  FB                                STI                                     ; ENABLE INTERRUPTS IN CASE THEY ARE OFF


                                        ; CHECK IF ANY ERRORS OCCURED
                                        ; WHILE WATING FOR COMMAND TO COMPLETE


016B  8A 04                 NCB_DONE?:  MOV       AL,DS:[SI]                    ;GET STATUS FOR LANA_X (X=0 OR 1)
016D  A8 04                             TEST      AL,LANA_HARD_ERR              ;DID NAY HARDWARE ERRORS OCCUR
016F  74 0E                             JZ        NCB_DONE1?                    ;NO ERROR GO AND KEEP WAITING FOR
                                                                                ;COMMAND TO COMPLETE
0171  B4 40                             MOV       AH,NCBSYS_ERR?                ;ASSUME A TIMEOUT ERROR
0173  A8 02                             TEST      AL,LANA_HARD_ERR1             ;THEN CHECK ERROR REPORTED FROM LANA
0175  74 16                             JZ        CHK_EXIT_ERR                  ;NO ERROR REPORT THEN IT WAS A TIMEOUT ERROR
0177  52                                PUSH      DX                            ;
0178  83 C2 02                          ADD       DX,DR                         ;POINT TO DATA REGISTER
017B  EC                                IN        AL,DX                         ;AND GET VALUE OF ERROR REPORTED BY LANA
017C  8A E0                             MOV       AH,AL                         ;
017E  5A                                POP       DX                            ;

017F                        NCB_DONE1?:
017F  26: 80 7F 31 FF                   CMP       ES:BYTE PTR [BX].NCB_CMD_CPLT,NCBNOT_DONE?   ;CHECK IF COMMAND COMPLETED
0184  74 E5                             JE        NCB_DONE?                     ;NO THEN GO AND CHECK IF ANY ERRORS
                                                                                ;HAVE OCCURED WHILE WATING
0186  26: 8A 67 01                      MOV       AH,ES:[BX].NCB_RETCODE        ;DONE THEN GET VALUE FOR RETURN_CODE
018A  EB 09 90                          JMP       BIOS_EXIT                     ;GO GET READY TO EXIT

018D                        CHK_EXIT_ERR:

018D  50                                PUSH      AX
018E  B8 9180                           MOV       AX,DEV_BUSY_POST              ;INDICATE THAT WE ARE COMMING OUT OF WAIT LOOP
0191  CD 15                             INT       15H
0193  FB                                STI                                     ;MAKE SURE INTERRUPS ARE ENABLE
0194  58                                POP       AX


                                        ; ALL DONE.  UPDATE RETURN CODE

0195  26: 88 67 31          BIOS_EXIT:  MOV       ES:[BX].NCB_CMD_CPLT,AH       ;UPDATE COMMAND COMPLETED FIELD AND
0199  26: 88 67 01                      MOV       ES:[BX].NCB_RETCODE,AH        ;UPDATE RETURN CODE FIELD

019D  8A C4                 BIOS_EXIT1: MOV       AL,AH                         ;MAKE SURE AL CONTAINS VALUE FOR RETURN CODE

                                        ; RETURN

                                        RESTORE   <DI,SI,DX,CX,DS>
019F  5F                  +                POP      DI
01A0  5E                  +                POP      SI
01A1  5A                  +                POP      DX
01A2  59                  +                POP      CX
01A3  1F                  +                POP      DS

01A4  CF                                IRET                                    ;  END.

01A5                        MAIN        ENDP
```

```
                          ;------------------------------------------------------------------------
                          ; CHK_STATE
                          ;
                          ;    CHECKS FOR ANY IMMEDIATELY DETECTABLE PROBLEMS.
                          ;
                          ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:
                          ;
                          ;       REGISTER  !   ACCESS  !              USAGE
                          ;       ----------+-----------+------------------------------------
                          ;       DS        !   CONST   ! LO_MEM_SEG
                          ;       ES:BX     !   CONST   ! NCB @
                          ;          CL     !   CONST   ! NCB_COMMAND
                          ;          DX     !   RESULT  ! LANA'S BASE PORT (LANA_X_SR) @
                          ;          DI     !   RESULT  ! LANA_X_HIR PORT @
                          ;       DS:SI     !   RESULT  ! LANA_X_STATUS @
                          ;          AL     !   DESTROY ! INTERNAL
                          ;          AH     !   VAR     ! NETBIOS RETURN CODE:
                          ;                 !           !    NCBLANA_NUM?
                          ;                 !           !    NCBSYS_ERR?
                          ;                 !           !    NCBLANA_LOCKED?
                          ;                 !           !    NCBBAD_CMD?
                          ;
                          ; INTERRUPTS SHOULD BE MASKED ON ENTRY.
                          ;
                          ;------------------------------------------------------------------------

01A5                      CHK_STATE  PROC      NEAR

                                     ; IS NCB_LANA_NUM A VALID LANA# ?

01A5  26: 80 7F 30 01                CMP       ES:BYTE PTR [BX].NCB_LANA_NUM,01H  ;CHECK FOR VALID LANA NUMBER
01AA  76 05                          JNA       SET_LANA                    ;VALID LANA GO AND SETUP
01AC  B4 23                          MOV       AH,NCBLANA_NUM?             ;BAD LANA NUMBER
01AE  E9 02BD R                      JMP       CHK_EXIT                    ;LEAVE AND REPORT ERROR

                                     ; YES.  GET LANA'S BASE PORT, HIR PORT, & STATUS BYTE @'S

01B1  74 0D               SET_LANA:  JE        IS_LANA_1                   ;IS IT LANA 1 GO AND SET UP
01B3  BA 0360                        MOV       DX,LANA_0                   ;LANA_0 GET BASE ADDRESS=360H
01B6  BF 0363                        MOV       DI,LANA_0_HIR               ;LANA_0 HIR ADDRESS=363H
01B9  8D 36 00A2                     LEA       SI,DS:LANA_0_STATUS         ;POINT TO LANA_0 STATUS
01BD  EB 4C 90                       JMP       PRESENT0?                   ;GO AND SEE IF IT IS PRESENT
01C0  BA 0368             IS_LANA_1: MOV       DX,LANA_1                   ;LANA_1 GET BASE ADDRESS
01C3  BF 036B                        MOV       DI,LANA_1_HIR               ;LANA_1 HIR ADDRESS=36BH
01C6  8D 36 00A3                     LEA       SI,DS:LANA_1_STATUS         ;POINT TO LANA_1 STATUS

                                     ; INSURE LANA IS PRESENT IN MACHINE

01CA  F6 04 01            PRESENT1?: TEST      BYTE PTR DS:[SI],LANA_INIT  ;HAS LANA_1 BEEN INITIALIZE
01CD  74 03                          JZ        CHK_P_INIT                  ;NO,THEN GO AND SEE IF IT IS PRESENT
01CF  E9 028C R                      JMP       C_ERR?                      ;PRESENT AND INITIALIZE,GO AND CHECK FOR ERRORS
01D2  F6 04 80            CHK_P_INIT: TEST     BYTE PTR DS:[SI],LANA_PRESENT ;IS LANA_1 PRESENT
01D5  75 05                          JNZ       L1_IRQ                      ;YES THEN GO AND CHECK INTERRUPT LEVEL
01D7  B4 23                          MOV       AH,NCBLANA_NUM?             ;BAD LANA NUMBER ERROR
01D9  E9 02BD R                      JMP       CHK_EXIT                    ;LEAVE AND REPORT ERROR
01DC  F6 04 10            L1_IRQ:    TEST      BYTE PTR DS:[SI],LANA_IRQ   ;IS THIS LANA RUNNING USING INTERRUPT LEVEL 2
01DF  75 15                          JNZ       L1_IRQ3                     ;NO THEN IT MUST BE THREE
01E1  50                             PUSH      AX
01E2  B8 0000                        MOV       AX,INTERRUPT_VECTOR_SEGMENT;
01E5  8E D8                          MOV       DS,AX                       ;POINT TO LOW MEMORY SEGMENT
01E7  C7 06 0028 0000 E              MOV       DS:WORD PTR INTR_2,OFFSET LANA_1_HNDLR;POINT TO INTERRUPT HANDLER FOR LANA_1
01ED  8C C8                          MOV       AX,CS                       ;
01EF  A3 002A                        MOV       DS:WORD PTR INTR_2+2,AX     ;SEGMENT OF INTERRUPT HANDLER
01F2  58                             POP       AX                          ;
01F3  EB 54 90                       JMP       EI2                         ;GO AND ENABLE HARDWARE INTERRUPT 2

01F6  50                  L1_IRQ3:   PUSH      AX
01F7  B8 0000                        MOV       AX,INTERRUPT_VECTOR_SEGMENT;
01FA  8E D8                          MOV       DS,AX                       ;POINT TO LOW MEMORY SEGMENT
01FC  C7 06 002C 0000 E              MOV       DS:WORD PTR INTR_3,OFFSET LANA_1_HNDLR;POINT TO INTERRUPT HANDLER FOR LANA_1
0202  8C C8                          MOV       AX,CS
0204  A3 002E                        MOV       DS:INTR_3+2,AX              ;SEGMENT OF INTERRUPT HANDLER
0207  58                             POP       AX
0208  EB 66 90                       JMP       EI3                         ;GO NAD ENABLE HARDWARE INTERRUPT 3

020B  F6 04 01            PRESENT0?: TEST      BYTE PTR DS:[SI],LANA_INIT  ;HAS LANA_0 BEEN INITIALIZE
020E  75 7C                          JNZ       C_ERR?                      ;PRESENT AND INITIALIZE,GO AND CHECK FOR ERRORS
0210  F6 04 80                       TEST      BYTE PTR DS:[SI],LANA_PRESENT ;IS LANA_0 PRESENT
0213  75 05                          JNZ       L0_IRQ                      ;YES THEN GO AND CHECK INTERRUPT LEVEL
0215  B4 23                          MOV       AH,NCBLANA_NUM?             ;BAD LANA NUMBER ERROR
0217  E9 02BD R                      JMP       CHK_EXIT                    ;LEAVE AND REPORT ERROR
021A  F6 04 10            L0_IRQ:    TEST      BYTE PTR DS:[SI],LANA_IRQ   ;IS THIS LANA RUNNING USING INTERRUPT LEVEL 2
021D  75 15                          JNZ       L0_IRQ3                     ;NO THEN IT MUST BE THREE
021F  50                             PUSH      AX
0220  B8 0000                        MOV       AX,INTERRUPT_VECTOR_SEGMENT
0223  8E D8                          MOV       DS,AX                       ;POINT TO LOW MEMORY SEGMENT
0225  C7 06 0028 0000 E              MOV       DS:WORD PTR INTR_2,OFFSET LANA_0_HNDLR;POINT TO INTERRUPT HANDLER FOR LANA_0
022B  8C C8                          MOV       AX,CS
022D  A3 002A                        MOV       DS:INTR_2+2,AX              ;SEGMENT OF INTERRUPT HANDLER
0230  58                             POP       AX
0231  EB 16 90                       JMP       EI2                         ;GO AND ENABLE HARDWARE INTERRUPT 2

0234  50                  L0_IRQ3:   PUSH      AX
0235  B8 0000                        MOV       AX,INTERRUPT_VECTOR_SEGMENT
0238  8E D8                          MOV       DS,AX                       ;POINT TO LOW MEMORY SEGMENT
023A  C7 06 002C 0000 E              MOV       DS:WORD PTR INTR_3,OFFSET LANA_0_HNDLR;POINT TO INTERRUPT HANDLER FOR LANA_0
0240  8C C8                          MOV       AX,CS
0242  A3 002E                        MOV       DS:WORD PTR INTR_3+2,AX     ;SEGMENT OF INTERRUPT HANDLER
0245  58                             POP       AX
0246  EB 28 90                       JMP       EI3                         ;GO NAD ENABLE HARDWARE INTERRUPT 3

                                     ; ENABLE

0249  E4 21               EI2:       IN        AL,INTA01                   ;
024B  24 FB                          AND       AL,ENABLE_INT2              ;
024D  EB 00                          JMP       $+2
024F  E6 21                          OUT       INTA01,AL                   ;ENABLE HARWARE INTERRUPT 2

0251  1E                             PUSH      DS
0252  53                             PUSH      BX
0253  50                             PUSH      AX
0254  B8 FFFF                        MOV       AX,ROS_CODE                 ;POINT TO ROS_CODE AREA
0257  8E D8                          MOV       DS,AX
0259  58                             POP       AX
025A  BB 000E                        MOV       BX,PC_ID_ADD                ;CHECK WHAT PC ARE WE RUNNING ON
025D  8A 07                          MOV       AL,[BX]
025F  5B                             POP       BX
0260  1F                             POP       DS
0261  3C FC                          CMP       AL,PC3                      ;IS IT A PC3
0263  75 13                          JNZ       ENABLE_GO_INT               ;NO ,THEN GO AND ENABLE GO INTERRUPT
0265  E4 A1                          IN        AL,INTB01
0267  24 FD                          AND       AL,ENABLE_INT9
0269  EB 00                          JMP       $+2
026B  E6 A1                          OUT       INTB01,AL                   ;ENABLE INTERRUPT 9
026D  EB 09 90                       JMP       ENABLE_GO_INT               ;GO AND ENABLE GO INTERRUPT
```

```
0270  E4 21                E13:     IN        AL,INTA01
0272  24 F7                         AND       AL,ENABLE_INT3
0274  EB 00                         JMP       $+2
0276  E6 21                         OUT       INTA01,AL              ;ENABLE HATDWARE INTERRUPT 3

0278                       ENABLE_GO_INT:
0278  50                            PUSH      AX
0279  B8 0040                       MOV       AX,LO_MEM_SEG
027C  8E D8                         MOV       DS,AX
027E  58                            POP       AX

027F  80 0C 01                      OR        BYTE PTR DS:[SI],LANA_INIT ;SET UP FLAG FOR LANA PRESENT AND INITIALIZED

0282  87 FA                         XCHG      DI,DX                  ;POINT TO HIR
0284  EC                            IN        AL,DX
0285  EB 00                         JMP       $+2
0287  0C 01                         OR        AL,GO                  ;ENABLE GO INTERRUPT
0289  EE                            OUT       DX,AL
028A  87 D7                         XCHG      DX,DI

                                    ; HAS THIS LANA HAD A HARDWARE ERROR?

028C  8A 04                C_ERR?:  MOV       AL,DS:[SI]             ; GET LANA_X_STATUS BYTE
028E  A8 04                         TEST      AL,LANA_HARD_ERR       ; TEST FOR ANY HARDAWARE ERRORS
0290  74 11                         JZ        LOCKED?                ; IF NOT, CHK OTHER THINGS
0292  B4 40                         MOV       AH,NCBSYS_ERR?         ; ASSUME A TIMEOUT ERROR
0294  A8 02                         TEST      AL,LANA_HARD_ERR1      ; CHECK FOR FATAL ERROR REPORTED BY LANA
0296  74 25                         JZ        CHK_EXIT               ; NO,THEN IT WAS A TIMEOUT
0298  52                            PUSH      DX                     ;
0299  83 C2 02                      ADD       DX,DR                  ; POINT TO DATA REGISTER
029C  EC                            IN        AL,DX                  ; GET VALUE FOR FATAL ERROR
029D  8A E0                         MOV       AH,AL                  ;
029F  5A                            POP       DX                     ;
02A0  EB 1B 90                      JMP       CHK_EXIT               ; LEAVE AND REPORT ERROR


                                    ; IS THIS LANA LOCKED?

02A3  A8 40                LOCKED?: TEST      AL,LANA_LOCKED         ; CHECK FOR INTERFACE BUSY
02A5  74 05                         JZ        BAD_CMD?               ; IF NOT, CHK CMD CODE
02A7  B4 21                         MOV       AH,NCBLANA_LOCKED?     ; INTERFACE BUSY
02A9  EB 12 90                      JMP       CHK_EXIT               ; LEAVE AND REPORT ERROR

                                    ; VALID NETBIOS COMMAND?
                                                                     ;
02AC                       BAD_CMD?: SAVE     <BX>                   ;
02AC  53                 +             PUSH     BX
02AD  8A C1                         MOV       AL,CL                  ; GET COMMAND VALUE
02AF  2E: 8D 1E 0011 R              LEA       BX,CS:COMMAND_TBL      ; POINT TO COMMAND TABLE
02B4  2E: D7                        XLAT      CS:COMMAND_TBL         ;
02B6  3C 00                         CMP       AL,0                   ; VALID COMMAND
                                    RESTORE   <BX>                   ;
02B8  5B                 +             POP      BX
02B9  7D 02                         JGE       CHK_EXIT               ; LEAVE AND CONTINUE PROCESSING COMMAND
02BB  B4 03                BAD_CMD: MOV       AH,NCBBAD_CMD?         ; REPORT BAD COMMAND ERROR

                                    ; RETURN

02BD  C3                   CHK_EXIT: RET

02BE                       CHK_STATE  ENDP
                           ;_______________________________________________________________________
                           ;                                                                       |
                           ; CONFIG                                                                |
                           ;                                                                       |
                           ;   RECONFIGURES THE SPECIFIED LANA. GIVES LANA UP TO 35 SECONDS TO COMPLETE.|
                           ;                                                                       |
                           ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                       |
                           ;                                                                       |
                           ;       REGISTER   |   ACCESS  |               USAGE                    |
                           ;       -----------+-----------+---------------------------------------- |
                           ;       DS         |   CONST   | LO_MEM_SEG                             |
                           ;       DS:SI      |   CONST   | LANA_X_STATUS @                        |
                           ;          DX      |   CONST   | LANA'S BASE PORT (LANA_X_SR) @         |
                           ;          DI      |   CONST   | LANA_X_HIR PORT @                      |
                           ;          AL      |  DESTROY  | INTERNAL                               |
                           ;          AH      |    VAR    | NETBIOS RETURN CODE:                   |
                           ;                              |     NCBSYS_ERR?                        |
                           ;                                                                       |
                           ; INTERRUPTS SHOULD BE UNMASKED ON ENTRY.                               |
                           ;                                                                       |
                           ;_______________________________________________________________________|

02BE                       CONFIG    PROC     NEAR

                                    SAVE      <CX,ES,DI,DX,BX>
02BE  51                 +             PUSH     CX
02BF  06                 +             PUSH     ES
02C0  57                 +             PUSH     DI
02C1  52                 +             PUSH     DX
02C2  53                 +             PUSH     BX

02C3  E8 062C R                     CALL      GET_INTERFACE          ; TRY TO GET HOLD OF LANA_X
                                                                     ;
02C6  80 FC 00                      CMP       AH,NCBGOOD_RET?        ; DID WE GET HOLD OF INTERFACE
02C9  74 03                         JZ        SPIO_LANA              ; IF SO THEN CONTINUE WITH RESET COMMAND
02CB  E9 0365 R                     JMP       CONFIG_EXI             ; LEAVE AND REPORT ERROR
                                                                     ;
02CE                       SPIO_LANA:

                                    ; SETUP FOR PROGRAMMED I/O FROM LANA

02CE  87 FA                         XCHG      DI,DX                       ; POIT TO HIR
02D0  EC                            IN        AL,DX                       ; GET CURRENT INTERFACE SETUP
02D1  24 C6                         AND       AL,ALL_BITS-IO_METHOD-DD_BIT-GI ; MASK OFF AREA OF INTEREST
02D3  0C 08                         OR        AL,PC_TO_LANA               ; FIRST SET DIRECTION
02D5  EB 00                         JMP       $+2
02D7  EE                            OUT       DX,AL
02D8  0C 00                         OR        AL,PROGRAMMED_IO            ; THEN SET I/O METHOD
02DA  EB 00                         JMP       $+2
02DC  EE                            OUT       DX,AL
02DD  87 D7                         XCHG      DX,DI

                                    ; SETUP PR FOR "RE_CONFIG" PRIMARY CMD

                                    SAVE      <DX>
02DF  52                 +             PUSH     DX
02E0  83 C2 01                      ADD       DX,PR                          ; POINT TO PARAMETERS REGISTER
02E3  B0 05                         MOV       AL,RE_CONFIG                   ; CMD CODE
02E5  EE                            OUT       DX,AL
02E6  26: 8A 47 02                  MOV       AL,ES:BYTE PTR [BX].NCB_LSN    ; NUMBER OF SESSIONS
02EA  EB 00                         JMP       $+2
02EC  EE                            OUT       DX,AL
```

```
02ED  26: 8A 47 03               MOV      AL,ES:BYTE PTR [BX].NCB_NUM       ; NUMBER OF CB'S
02F1  EB 00                      JMP      $+2
02F3  EE                         OUT      DX,AL
02F4  EB 00                      JMP      $+2
02F6  EE                         OUT      DX,AL
02F7  EB 00                      JMP      $+2
02F9  EE                         OUT      DX,AL                             ; PAD TO 7-BYTES FULL
02FA  EB 00                      JMP      $+2
02FC  EE                         OUT      DX,AL
02FD  EB 00                      JMP      $+2
02FF  EE                         OUT      DX,AL
                                 RESTORE  <DX>
0300  5A                +           POP      DX

                                 ; SAY GO

0301  B0 01                      MOV      AL,GO
0303  EB 00                      JMP      $+2
0305  EE                         OUT      DX,AL                   ;TELL LANA THAT WE HAVE A COMMAND
                                                                  ;READY FOR EXECUTION

                                 ; WAIT ON COMMAND ACCEPTED

0306  B9 0000                    MOV      CX,W_TIL_ACC            ;INDICATE WAIT TIMEOUT

0309  EC                WAIT_ACC: IN      AL,DX                   ;
030A  A8 01                      TEST     AL,GO                   ;WAIT FOR GO TO CLEAR INDICATING CMD ACCEPTED
030C  74 08                      JZ       CMD_ACC                 ;
030E  E2 F9                      LOOP     WAIT_ACC                ;
0310  E8 05A6 R                  CALL     CATASTROPHIC_ERROR      ;IF TIME OUT THEN WE MUST HAVE HARDWARE ERROR
0313  EB 3A 90                   JMP      CONFIG_EXIT             ;LEAVE AND REPORT ERROR


                                 ; CLEAR HCR

0316  87 FA             CMD_ACC:  XCHG    DI,DX                   ;POINT TO HIR
0318  EC                         IN       AL,DX                   ;
0319  24 FD                      AND      AL,ALL_BITS-HCR         ;RELEASE INTERFACE
031B  EB 00                      JMP      $+2                     ;
031D  EE                         OUT      DX,AL                   ;
031E  87 D7                      XCHG     DX,DI                   ;

                                 ; NOW WAIT UP TO A WHILE FOR RE_COFIG TO COMPLETE.

0320  BB 000A                    MOV      BX,10                   ;

0323                    CONFIG_CPLTO?:

0323  B9 0000                    MOV      CX,CONFIG_CPLT_TO        ;SET UP TIME OUT VALUE

0326                    CONFIG_CPLT?:
0326  EC                         IN       AL,DX                   ;
0327  A8 01                      TEST     AL,GO                   ;
0329  75 0B                      JNZ      INIT_CPLT?              ;
032B  E2 F9                      LOOP     CONFIG_CPLT?            ;
032D  4B                         DEC      BX                      ;
032E  75 F3                      JNZ      CONFIG_CPLTO?           ;
0330  E8 05A6 R                  CALL     CATASTROPHIC_ERROR      ;
0333  EB 1A 90                   JMP      CONFIG_EXIT             ;

                                 ; "INITIALIZATION COMPLETE" CMD?
0336                    INIT_CPLT?:SAVE   <DX>
0336  52                +           PUSH     DX
0337  83 C2 01                   ADD      DX,PR
033A  EC                         IN       AL,DX
033B  3C 41                      CMP      AL,INIT_CPLT
033D  74 07                      JE       CPLT_OK?
                                 RESTORE  <DX>
033F  5A                +           POP      DX
0340  E8 05A6 R                  CALL     CATASTROPHIC_ERROR
0343  EB 0A 90                   JMP      CONFIG_EXIT

                                 ; DID INIT COMPLETE OK?

0346  EC                CPLT_OK?: IN      AL,DX
                                 RESTORE  <DX>
0347  5A                +           POP      DX
0348  3C 80                      CMP      AL,INIT_CPLT_RET
034A  74 03                      JE       CONFIG_EXIT
034C  E8 05A6 R                  CALL     CATASTROPHIC_ERROR

                                 ; ACKNOWLEDGE HOST INT + TURN GI BACK ON

034F                    CONFIG_EXIT:
034F  EC                         IN       AL,DX
0350  0C 80                      OR       AL,80H
0352  EB 00                      JMP      $+2
0354  EE                         OUT      DX,AL

0355  B0 00                      MOV      AL,00H
0357  EB 00                      JMP      $+2
0359  EE                         OUT      DX,AL

035A  87 FA                      XCHG     DI,DX
035C  EC                         IN       AL,DX
035D  0C 01                      OR       AL,GI
035F  EB 00                      JMP      $+2
0361  EE                         OUT      DX,AL

                                 ; UNLOCK INTERFACE

0362  80 24 BF                   AND      BYTE PTR DS:[SI],ALL_BITS-LANA_LOCKED

                                 ; RETURN
0365                    CONFIG_EXI:

                                 RESTORE  <BX,DX,DI,ES,CX>
0365  5B                +           POP      BX
0366  5A                +           POP      DX
0367  5F                +           POP      DI
0368  07                +           POP      ES
0369  59                +           POP      CX
036A  C3                         RET

036B                    CONFIG   ENDP

                        ;___________________________________________________________________
                        ; NORMAL_CMD                                                         |
                        ;                                                                    |
                        ;    SETS UP THE NCB_RESERVE AREA & SENDS THE NCB OVER TO THE LANA.  |
                        ;                                                                    |
                        ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                    |
```

```
;        REGISTER   ¦  ACCESS  ¦                 USAGE
;       ------------+----------+------------------------------------------
;        DS         ¦  CONST   ¦ LO_MEM_SEG
;        ES:BX      ¦  CONST   ¦ NCB @
;        DS:SI      ¦  CONST   ¦ LANA_X_STATUS @
;            DX     ¦  CONST   ¦ LANA'S BASE PORT (LANA_X_SR) @
;            DI     ¦  CONST   ¦ LANA_X_HIR PORT @
;            CL     ¦  CONST   ¦ NCB_COMMAND
;            AL     ¦  DESTROY ¦ INTERNAL
;            AH     ¦  VAR     ¦ NETBIOS RETURN CODE:
;                   ¦          ¦     NCBMAX_CMD?
;                   ¦          ¦     NCBSYS_ERR?
;
; INTERRUPTS SHOULD BE MASKED ON ENTRY (EXCEPT FOR SET_PARM).
; INTERRUPTS WILL BE UNMASKED ON EXIT.
;
;_______________________________________________________________________

036B                      NORMAL_CMD PROC      NEAR

                                     SAVE      <CX>
036B  51                  +             PUSH      CX

                                     ; GET AND LOCK INTERFACE

036C  E8 062C R                      CALL      GET_INTERFACE           ; TRY AND GET HOLD OF LANA_X
036F  80 FC 00                       CMP       AH,NCBGOOD_RET?         ; ERROR TRYING TO GET HOLD OF LANA_X
0372  75 48                          JNE       NORM_EXIT               ; LEAVE AND REPORT ERROR

                                     ; ADJUST DEFAULTS, BUFFER@, & RESERVE AREA

0374  E8 05B9 R           ADJ_NCB:   CALL      ADJUST_NCB              ;CHANGES BUFFERS@ TO 32-BIT ADDR.

                                     ; GET NCB LENGTH (SANS POST@ & RESERVE)

                                     SAVE      <BX>
0377  53                  +             PUSH      BX
0378  2E: 8D 1E 0011 R               LEA       BX,CS:COMMAND_TBL       ;POINT TO COMMAND TABLE
037D  8A C1                          MOV       AL,CL                   ;GET COMMAND VALUE
037F  2E: D7                         XLAT      CS:COMMAND_TBL          ;GET LENGTH OF NCB TO BE SENT OVER TO LANA
0381  8A C8                          MOV       CL,AL
0383  B5 00                          MOV       CH,0
                                     RESTORE   <BX>
0385  5B                  +             POP       BX

                                     ; SETUP HIR & PR FOR "XFER NCB TO LANA" CMD

0386  B0 01                          MOV       AL,NCB_TO_LANA          ;PRIMARY COMMAND INDICATING TRANSFER
                                                                       ;OF NCB TO LANA
0388  E8 0475 R                      CALL      SETUP_HIR_AND_PR        ;SET UP LANA_X HIR & PR FOR SOME PRIMARY CMD.

                                     ; PROGRAM I/O THE NCB OVER TO THE LANA

038B  E8 053A R                      CALL      PROGRAM_IO_NCB          ;

                                     ; IF ERRORS, RESTORE NCB_BUFFER@ TO SEG:OFF

038E  80 FC 00                       CMP       AH,NCBGOOD_RET?
0391  74 29                          JE        NORM_EXIT               ;NO ERROR CONTINUE WITH CMD.
0393  26: FF 77 38                   PUSH      ES:WORD PTR [BX].NCB_RESERVE_BUFFER@
0397  26: 8F 47 04                   POP       ES:WORD PTR [BX].NCB_BUFFER@
039B  26: FF 77 3A                   PUSH      ES:WORD PTR [BX].NCB_RESERVE_BUFFER@+2
039F  26: 8F 47 06                   POP       ES:WORD PTR [BX].NCB_BUFFER@+2

03A3  26: 8A 07                      MOV       AL,ES:[BX].NCB_COMMAND  ;GET COMMAND
03A6  24 7F                          AND       AL,7FH                  ;CLEAR HIGH ORDER BIT
03A8  3C 17                          CMP       AL,NCBSENDMULTIPLE      ;IF COMMAND WAS A CHAIN SEND THEN
                                                                       ;RESTORE SECOND BUFFER
03AA  75 10                          JNZ       NORM_EXIT               ;NO THEN CONTINUE PROCESSING


03AC  26: FF 77 3C                   PUSH      ES:WORD PTR [BX].NCB_RESERVE_BUFFER2@
03B0  26: 8F 47 0C                   POP       ES:WORD PTR [BX].NCB_BUFFER@+8
03B4  26: FF 77 3E                   PUSH      ES:WORD PTR [BX].NCB_RESERVE_BUFFER2@+2
03B8  26: 8F 47 0E                   POP       ES:WORD PTR [BX].NCB_BUFFER@+0AH

                                     ; RELEASE & UNLOCK INTERFACE

03BC  87 FA               NORM_EXIT: XCHG      DI,DX
03BE  EC                             IN        AL,DX                   ; POINT TO HIR
03BF  24 FD                          AND       AL,ALL_BITS-HCR         ; TURN OFF HCR
03C1  EB 00                          JMP       $+2
03C3  EE                             OUT       DX,AL
03C4  87 D7                          XCHG      DX,DI
03C6  80 24 BF                       AND       BYTE PTR DS:[SI],ALL_BITS-LANA_LOCKED ;CLEAR INTERFACE BUSY FLAG

                                     ; RETURN

                                     RESTORE   <CX>
03C9  59                  +             POP       CX
03CA  C3                             RET

03CB                      NORMAL_CMD ENDP


                          ;_______________________________________________________________________
                          ; CANCEL_CMD
                          ;
                          ;    ASKS THE LANA TO CANCEL A NETBIOS CMD.
                          ;
                          ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:
                          ;
                          ;        REGISTER   ¦  ACCESS  ¦                 USAGE
                          ;       ------------+----------+------------------------------------------
                          ;        DS         ¦  CONST   ¦ LO_MEM_SEG
                          ;        ES:BX      ¦  CONST   ¦ NCB @
                          ;        DS:SI      ¦  CONST   ¦ LANA_X_STATUS @
                          ;            DX     ¦  CONST   ¦ LANA'S BASE PORT (LANA_X_SR) @
                          ;            DI     ¦  CONST   ¦ LANA_X_HIR PORT @
                          ;            AL     ¦  DESTROY ¦ INTERNAL
                          ;            AH     ¦  VAR     ¦ NETBIOS RETURN CODE:
                          ;                   ¦          ¦     NCBCMD_CNL?
                          ;                   ¦          ¦     NCBMAX_CMD?
                          ;                   ¦          ¦     NCBSYS_ERR?
                          ;
                          ; INTERRUPTS SHOULD BE MASKED ON ENTRY.
                          ; INTERRUPTS WILL BE UNMASKED ON EXIT.
                          ;
                          ;_______________________________________________________________________

03CB                      CANCEL_CMD PROC      NEAR

                                     SAVE      <CX,ES,BX>
03CB  51                  +             PUSH      CX
```

```
03CC 06              +          PUSH    ES
03CD 53              +          PUSH    BX

                            ; GET INTERFACE  (RETURN RC IF ERROR)

03CE E8 062C R              CALL    GET_INTERFACE             ;TRY TO GET HOLD OF LANA_X
03D1 80 FC 00               CMP     AH,NCBGOOD_RET?           ;DID WE GET INTERFACE TO LANA_X

03D4 74 03                  JZ      GET_NCB_C                 ;YES ,THEN GO AND DO CANCEL
03D6 E9 0471 R              JMP     CAN_EXIT                  ;NO,LEAVE AND REPORT ERROR

                            ; PT TO NCB OF CMD TO CANCEL

03D9 26: C4 5F 04  GET_NCB_C: LES   BX,ES:[BX].NCB_BUFFER@    ;GET ADDRESS OF COMMAND TO BE CANCEL

03DD 26: 8A 07              MOV     AL,ES:[BX].NCB_COMMAND    ;GET VALUE OF COMMAND TO BE CANCEL
03E0 24 7F                  AND     AL,7FH                    ;GET HIGH ORDER BIT OUT OF THE WAY


03E2 3C 32                  CMP     AL,NCBRESET               ;IS IT A RESET COMMAND
03E4 75 05                  JNZ     IS_CNCL                   ;NO,THEN SEE IF IT IS CANCEL COMMAND
03E6 B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NOT VALID TO CANCEL
03E8 EB 7A 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

03EB 3C 35         IS_CNCL: CMP     AL,NCBCANCEL              ;IS IT A CANCEL COMMAND
03ED 75 05                  JNZ     IS_DTGRM                  ;NO,GO AND CHECK IF SEND DATAGRAM
03EF B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NO VALID TO CANCEL
03F1 EB 71 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

03F4 3C 20         IS_DTGRM: CMP    AL,NCBSENDDATAGRAM        ;IS IT SEND DATAGRAM
03F6 75 05                  JNZ     IS_BDGRM                  ;NO,GO AND CHECK IF SEND BROADCAST
03F8 B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NOT VALID TO CANCEL
03FA EB 68 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

03FD 3C 22         IS_BDGRM: CMP    AL,NCBSENDBROADCAST       ;IS IT A SEND BROADCAST
03FF 75 05                  JNZ     IS_ADDNME                 ;NO,GO AND CHECK IF ADDNAME
0401 B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NOT VALID TO CANCEL
0403 EB 5F 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

0406 3C 30         IS_ADDNME: CMP   AL,NCBADDNAME             ;IS IT AN ADDNAME
0408 75 05                  JNZ     IS_ADDGNME                ;NO,GO AND CHECK IF ADDGROUPNAME
040A B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NOT VALID TO CANCEL
040C EB 56 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

040F 3C 36         IS_ADDGNME:CMP   AL,NCBADDGROUPNAME        ;IS IT AN ADDGROUP NAME
0411 75 05                  JNZ     IS_DNME                   ;NO,GO AND CHECK FOR DELETE NAME
0413 B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NOT VALID TO CANCEL
0415 EB 4D 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

0418 3C 31         IS_DNME:  CMP    AL,NCBDELETENAME          ;IS IT A DELETE NAME
041A 75 05                  JNZ     IS_SSTAT                  ;NO,GO CHECK SESSION STATUS
041C B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NOT VALID TO CANCEL
041E EB 44 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

0421 3C 34         IS_SSTAT: CMP    AL,NCBSESSIONSTATUS       ;IS IT A SESSION STATUS
0423 75 05                  JNZ     START_CAN                 ;NO,GO TRY TO CANCEL OUTSTANDING  COMMAND
0425 B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NOT VALID TO CANCEL
0427 EB 3B 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

                            ; SETUP HIR & PR FOR "ABORT NCB" PRIMARY CMD

042A B0 02         START_CAN: MOV   AL,ABORT_NCB              ;PRIMARY COMMAND FOR ABORT NCB
042C 26: 8A 0F              MOV     CL,ES:[BX].NCB_COMMAND    ;
042F E8 0475 R              CALL    SETUP_HIR_AND_PR          ;SET UP LANA_X HIR AND PR FOR SOME PRIMARY COMMAND

                            ; TELL LANA TO GO

0432 B0 01                  MOV     AL,GO
0434 EB 00                  JMP     $+2
0436 EE                     OUT     DX,AL                     ;TELL LANA COMMAND READY FOR EXECUTION

                            ; WAIT UPT TO "AWHILE" FOR LANA TO COMPLETE THE CANCEL

0437 B9 0000                MOV     CX,CANCEL_CPLT_TO         ;SET UP TIME OUT VALUE
043A EB 00                  JMP     $+2
043C EC            CNCL_CPLT?:IN    AL,DX                     ;
043D A8 01                  TEST    AL,GO                     ;
043F 74 08                  JZ      CNCL_OK?                  ;
0441 E2 F9                  LOOP    CNCL_CPLT?                ;

                            ; CATASTROPHIC ERROR.  TIMEOUT ON CANCEL

0443 E8 05A6 R              CALL    CATASTROPHIC_ERROR        ;TIMEOUT THEN INTERFACE ERROR
0446 EB 1C 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

                            ; IF COMPLETED WITH ERROR,

0449 24 06         CNCL_OK?: AND    AL,CPLT_CODE              ;GET COMPLITION CODE
044B 3C 00                  CMP     AL,GOOD_RET?              ;DID IT COMPLETE OK
044D 74 15                  JE      CAN_EXIT2                 ;EVERY THING OK THEN LEAVE
044F 3C 02                  CMP     AL,BAD_PARM?              ;
0451 75 05                  JNZ     T02                       ;
0453 B4 24                  MOV     AH,NCBCMD_CNL?            ;ERROR COMMAND COMPLETED WHILE CANCEL OCCURRING
0455 EB 0D 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR
0458 3C 04         T02:     CMP     AL,CANT_CPLT?             ;
045A 75 05                  JNZ     OTHERR                    ;
045C B4 26                  MOV     AH,NCBNO_CNL?             ;ERROR COMMAND NOT VALID TO CANCEL
045E EB 04 90               JMP     CAN_EXIT2                 ;LEAVE AND REPORT ERROR

0461 E8 05A6 R     OTHERR:  CALL    CATASTROPHIC_ERROR        ;SHOULD NOT GET OTHER COMPLITION CODES

                            ; RELEASE & UNLOCK INTERFACE

0464 87 FA         CAN_EXIT2: XCHG  DI,DX                     ;POINT TO HIR
0466 EC                     IN      AL,DX                     ;
0467 24 FD                  AND     AL,ALL_BITS-HCR           ;TURN OFF HCR
0469 EB 00                  JMP     $+2                       ;
046B EE                     OUT     DX,AL                     ;
046C 87 D7                  XCHG    DX,DI                     ;
046E 80 24 BF               AND     BYTE PTR DS:[SI],ALL_BITS-LANA_LOCKED ;CLEAR INTERFACE BUSY FLAG

                            ; RETURN

0471               CAN_EXIT: RESTORE <BX,ES,CX>
0471 5B              +          POP     BX
0472 07              +          POP     ES
0473 59              +          POP     CX
0474 C3                     RET

0475               CANCEL_CMD ENDP

                   ;
                   ; SETUP_HIR_AND_PR
                   ;
```

```
                                  ;    SETS UP THE LANA'S HIR & PR PORTS FOR SOME PRIMARY CMD.         |
                                  ;                                                                    |
                                  ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                    |
                                  ;                                                                    |
                                  ;       REGISTER   |  ACCESS  |          USAGE                       |
                                  ;     -------------+----------+------------------------------------  |
                                  ;       ES:BX      |  CONST   | NCB @                                |
                                  ;          AL      |  CONST   | PRIMARY COMMAND CODE                 |
                                  ;          CX      |  CONST   | DEPENDS ON CMD (NCB LEN OR <OLD CMD,DUMMY>) |
                                  ;          DX      |  CONST   | LANA'S BASE PORT (LANA_X_SR) @       |
                                  ;          DI      |  CONST   | LANA_X_HIR PORT @                    |
                                  ;                                                                    |
                                  ;____________________________________________________________________|

0475                              SETUP_HIR_AND_PR PROC      NEAR

                                          SAVE      <AX>
0475  50                        +            PUSH      AX

                                          ; PUT CMD CODE IN PR

                                          SAVE      <DX>
0476  52                        +            PUSH      DX
0477  83 C2 01                            ADD       DX,PR                  ;POINT TO PR
047A  EE                                  OUT       DX,AL                  ;SEND PRIMARY COMMAND  CODE

                                          ; SETUP INTERFACE FOR PROGRAMMED I/O TO LANA

047B  87 FA                               XCHG      DI,DX                  ;POINT TO HIR
047D  EB 00                               JMP       $+2
047F  EC                                  IN        AL,DX
0480  24 C7                               AND       AL,ALL_BITS-IO_METHOD-DD_BIT    ; LEAVE GI ON, CLR DD_BIT
0482  0C 08                               OR        AL,PC_TO_LANA                   ; FIRST SET DIRECTION
0484  EB 00                               JMP       $+2
0486  EE                                  OUT       DX,AL
0487  0C 00                               OR        AL,PROGRAMMED_IO                ; THEN SET I/O METHOD
0489  EB 00                               JMP       $+2
048B  EE                                  OUT       DX,AL
048C  87 D7                               XCHG      DX,DI

                                          ; XLATE NCB@ TO 32-BIT ADDR & PUT IN PR

                                          SAVE      <CX>
048E  51                        +            PUSH      CX
048F  8C C0                               MOV       AX,ES
0491  B1 04                               MOV       CL,4
0493  D3 C0                               ROL       AX,CL
0495  8A C8                               MOV       CL,AL
0497  24 F0                               AND       AL,0F0H
0499  03 C3                               ADD       AX,BX
049B  73 02                               JNC       TOP_4_OK
049D  FE C1                               INC       CL
049F  EE                          TOP_4_OK:  OUT    DX,AL
04A0  8A C4                               MOV       AL,AH
04A2  EB 00                               JMP       $+2
04A4  EE                                  OUT       DX,AL
04A5  81 E1 000F                          AND       CX,000FH
04A9  8A C1                               MOV       AL,CL
04AB  EB 00                               JMP       $+2
04AD  EE                                  OUT       DX,AL
04AE  8A C5                               MOV       AL,CH
04B0  EB 00                               JMP       $+2
04B2  EE                                  OUT       DX,AL
                                          RESTORE   <CX>
04B3  59                        +            POP       CX

                                          ; PUT NCB_LENGTH/<OLD_CMD,DUMMY> IN PR

04B4  8A C1                               MOV       AL,CL
04B6  EE                                  OUT       DX,AL
04B7  8A C5                               MOV       AL,CH
04B9  EB 00                               JMP       $+2
04BB  EE                                  OUT       DX,AL
                                          RESTORE   <DX>
04BC  5A                        +            POP       DX

                                          ; RETURN

                                          RESTORE   <AX>
04BD  58                        +            POP       AX
04BE  C3                                  RET

04BF                              SETUP_HIR_AND_PR ENDP


                                  ;____________________________________________________________________
                                  ; DMA_START_UP                                                       |
                                  ;                                                                    |
                                  ;    STARTS UP A DMA XFER WITH A LANA.                               |
                                  ;                                                                    |
                                  ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                    |
                                  ;                                                                    |
                                  ;       REGISTER   |  ACCESS  |          USAGE                       |
                                  ;     -------------+----------+------------------------------------  |
                                  ;       DS         |  CONST   | LO_MEM_SEG                           |
                                  ;       ES:DI      |  CONST   | ADDRESS OF REMAINING DATA            |
                                  ;          DX      |  CONST   | LANA_X_HIR PORT @                    |
                                  ;          CX      |  DESTROY | LENGTH OF REMAINING DATA             |
                                  ;          AH      |  DESTROY | PRIMARY CMD CODE                     |
                                  ;          AL      |  DESTROY | INTERNAL                             |
                                  ;                                                                    |
                                  ; INTERRUPTS SHOULD BE MASKED ON ENTRY.                              |
                                  ; INTERRUPTS WILL BE MASKED ON EXIT.                                 |
                                  ;                                                                    |
                                  ;____________________________________________________________________|

04BF                              DMA_START_UP PROC     NEAR

                                          SAVE      <AX>
04BF  50                        +            PUSH      AX

                                          ; STOP OTHERS FROM USING DMA OR THE INTERFACE (TIL DMA DONE)

04C0  80 0E 00A1 80                       OR        BYTE PTR DS:[LANA_HARDWARE],DMA_3_BUSY ;SET DMA BUSY FLAG
04C5  3C 01                               CMP       AL,01                  ;IS IT LANA_0 REQUEST
04C7  75 08                               JNZ       SET_1                  ;NO THEN IT IS LANA_1 REQUEST
04C9  80 0E 00A2 20                       OR        BYTE PTR DS:[LANA_0_STATUS],LANA_DMAING ;SET FLAG FOR LANA_0 DMA
04CE  EB 06 90                            JMP       DMA_CONT
04D1                              SET_1:
04D1  80 0E 00A3 20                       OR        BYTE PTR DS:[LANA_1_STATUS],LANA_DMAING ;SET FLAG FOR LANA_1 DMA

04D6                              DMA_CONT:

                                          ; MASK DREQ_3 WHILE SETTING UP DMA
```

```
04D6  B0 07                     MOV      AL,DMA_MASK+DMA_CHNL_3
04D8  E6 0A                     OUT      DMA_MASK_CHNL,AL

                                ; SET MODE OF DMA_3 CHANNEL

04DA  B0 43                     MOV      AL,DMA_SINGLE_MODE+DMA_CHNL_3
04DC  80 FC 44                  CMP      AH,DATA_TO_LANA
04DF  74 05                     JE       DMA_READ
04E1  0C 04                     OR       AL,DMA_WRITE_XFER
04E3  EB 03 90                  JMP      OUT_MODE
04E6  0C 08         DMA_READ:   OR       AL,DMA_READ_XFER
04E8  E6 0B         OUT_MODE:   OUT      DMA_MODE,AL

                                ; CLEAR "HI/LO"  FLIP/FLOP

04EA  B0 FF                     MOV      AL,ALL_BITS              ; MAINLY FOR DELAY
04EC  EB 00                     JMP      $+2
04EE  E6 0C                     OUT      DMA_HI_LO_FF,AL          ; OUT ANYTHING TO DO CLR

                                ; OUT "LENGTH OF REMAINING DATA" TO DMA COUNT REGISTER

04F0  49                        DEC      CX                       ; COUNT IS N-1
04F1  8A C1                     MOV      AL,CL
04F3  E6 07                     OUT      DMA_3_COUNT,AL
04F5  8A C5                     MOV      AL,CH
04F7  EB 00                     JMP      $+2
04F9  E6 07                     OUT      DMA_3_COUNT,AL

                                ; CONVERT ES:DI TO 20-BIT ADDR & OUT TO DMA BASE & PAGE REGISTERS

04FB  8C C0                     MOV      AX,ES                    ; AX=SEG{ A<19::16>, A<15::04> }
04FD  B9 0004                   MOV      CX,4                     ; CH=0,  CL=ROL COUNT
0500  D3 C0                     ROL      AX,CL                    ; AX=SEG{ A<15::04>, A<19::16> }
0502  8A C8                     MOV      CL,AL                    ; CX=SEG{     0,    X, A<19::16> }
0504  80 E1 0F                  AND      CL,00FH                  ; CX=SEG{     0,    0, A<19::16> }
0507  24 F0                     AND      AL,0F0H                  ; AX=SEG{ A<15::04>,          0 }
                                                                  ;
0509  03 C7                     ADD      AX,DI                    ; AX=REAL{ A<15::00> } & CF=SEGMENT ADJ.
050B  12 CD                     ADC      CL,CH                    ; CX=REAL{ A<19::16> }
050D  E6 06                     OUT      DMA_3_BASE,AL            ; OUT A<07::00>
050F  8A C4                     MOV      AL,AH
0511  EB 00                     JMP      $+2
0513  E6 06                     OUT      DMA_3_BASE,AL            ; OUT A<15::08>
0515  8A C1                     MOV      AL,CL
0517  EB 00                     JMP      $+2
0519  E6 82                     OUT      DMA_3_PAGE,AL            ; OUT A<19::16>

                                ; SETUP HIR FOR DMA IN INDICATED DIRECTION

                                RESTORE  <AX>
051B  58                +          POP      AX
051C  EB 00                     JMP      $+2
051E  EC                        IN       AL,DX
051F  24 C7                     AND      AL,ALL_BITS-IO_METHOD-DD_BIT
0521  80 FC 44                  CMP      AH,DATA_TO_LANA          ;WHICH WAY ARE WE GOING
0524  74 05                     JE       TO_LANA                  ;GOING TO LANA
0526  0C 00                     OR       AL,LANA_TO_PC            ;ENABLE DIRECTION LANA_TO_PC
0528  EB 03 90                  JMP      SET_DD                   ;GO INDICATE DIRECTION
052B  0C 08         TO_LANA:    OR       AL,PC_TO_LANA            ;ENABLE DIRECTION PC_TO_LANA
052D  EE            SET_DD:     OUT      DX,AL
052E  0C 60                     OR       AL,DMA_IO+TCI
0530  EB 00                     JMP      $+2
0532  EE                        OUT      DX,AL
                                ; UNMASK DREQ_3 TO START DMA

0533  B0 03                     MOV      AL,DMA_CHNL_3
0535  EB 00                     JMP      $+2
0537  E6 0A                     OUT      DMA_MASK_CHNL,AL

                                ; RETURN

0539  C3                        RET

053A                DMA_START_UP ENDP

                    ;_________________________________________________________________
                    ;                                                                 |
                    ; PROGRAM_IO_NCB                                                  |
                    ;                                                                 |
                    ;     PROGRAM I/O'S THE NCB OVER TO THE LANA.                     |
                    ;                                                                 |
                    ;     THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                |
                    ;                                                                 |
                    ;        REGISTER  |  ACCESS  |             USAGE                 |
                    ;        ----------+----------+---------------------------------- |
                    ;        DS        |  CONST   | LO_MEM_SEG                        |
                    ;        ES:BX     |  CONST   | NCB @                             |
                    ;        DS:SI     |  CONST   | LANA_X_STATUS @                   |
                    ;          DX      |  CONST   | LANA'S BASE PORT (LANA_X_SR) @    |
                    ;          DI      |  CONST   | LANA_X_HIR PORT @                 |
                    ;          CX      | DESTROY  | NCB LENGTH (SANS POST@ & RESERVE) |
                    ;          AL      | DESTROY  | INTERNAL                          |
                    ;          AH      |   VAR    | NETBIOS RETURN CODE:              |
                    ;                  |          |    NCBSYS_ERR?                    |
                    ;                                                                 |
                    ; INTERRUPTS SHOULD BE UNMASKED ON ENTRY.                         |
                    ;                                                                 |
                    ;_________________________________________________________________|

053A                PROGRAM_IO_NCB PROC      NEAR

                                SAVE     <BX>
053A  53                +          PUSH     BX

                                ; SET "XFERRING NCB TO LANA" FLAG

053B  80 0C 08                  OR       BYTE PTR DS:[SI],LANA_GETTING_NCB

                                ; SETUP FOR XFER

                                SAVE     <DS,SI,DI,AX,DX>
053E  1E                +          PUSH     DS
053F  56                +          PUSH     SI
0540  57                +          PUSH     DI
0541  50                +          PUSH     AX
0542  52                +          PUSH     DX
0543  8C C0                     MOV      AX,ES
0545  8E D8                     MOV      DS,AX
0547  8B F3                     MOV      SI,BX                    ; DS:SI  = NCB @
0549  8B FA                     MOV      DI,DX                    ; DI     = LANA_X_SR PORT @
054B  83 C2 02                  ADD      DX,DR                    ; DX     = LANA_X_DR PORT @
054E  B4 10                     MOV      AH,DRE                   ; AH     = FLAG TO TEST

                                ; PUT 1ST 2 BYTES OF NCB INTO DR  (ASSUMES ROOM FOR 2)
```

```
0550  AC                               LODSB
0551  EE                               OUT       DX,AL                ;WRIET TO DATA RAGISTER A BYTE
0552  AC                               LODSB
0553  EB 00                            JMP       $+2                  ;
0555  EE                               OUT       DX,AL                ;ANOTHER BYTE
0556  83 E9 02                         SUB       CX,2

                                       ; SAY GO

0559  87 FA                            XCHG      DI,DX                ;POINT TO SR
055B  EB 00                            JMP       $+2
055D  EC                               IN        AL,DX
055E  24 79                            AND       AL,NOT (CPLT_CODE+HC)
0560  0C 01                            OR        AL,GO
0562  EB 00                            JMP       $+2
0564  EE                               OUT       DX,AL

                                       ; CHECK IF ROOM FOR ANOTHER BYTE NOW

0565  EB 00                            JMP       $+2
0567  EC                   DATA_ROOM?: IN        AL,DX
0568  84 C4                            TEST      AL,AH                ;CHECK IF ROOM FOR ANOTHER BYTE
056A  74 0B                            JZ        NO_ROOM              ;NO ROOM GO WAIT UNTIL ROOM AVAILABLE
056C  87 D7                GOT_ROOM:   XCHG      DX,DI                ;POINT TO DATA REGISTER
056E  AC                               LODSB
056F  EE                               OUT       DX,AL                ;SEND A BYTE
0570  87 FA                            XCHG      DI,DX                ;POIT TO SR
0572  E2 F3                            LOOP      DATA_ROOM?           ;GO SEE IF WE CAN SEND ANOTHER BYTE
0574  EB 14 90                         JMP       GO_CLR?              ;GO WAIT FOR GO TO CLEAR

                                       ; WAIT FOR ROOM FOR NCB BYTE

0577  BB 0000              NO_ROOM:    MOV       BX,DRE_LIMIT         ;SET TIME OUT VALUE
057A  EC                   ROOM_NOW?:  IN        AL,DX
057B  84 C4                            TEST      AL,AH
057D  75 ED                            JNZ       GOT_ROOM
057F  4B                               DEC       BX
0580  75 F8                            JNE       ROOM_NOW?
                                       RESTORE   <DX,AX,DI,SI,DS>
0582  5A                +                 POP       DX
0583  58                +                 POP       AX
0584  5F                +                 POP       DI
0585  5E                +                 POP       SI
0586  1F                +                 POP       DS
0587  EB 18 90                         JMP       NCB_ERROR

                                       ; WAIT FOR GO TO CLEAR

058A                       GO_CLR?:    RESTORE   <DX,AX,DI,SI,DS>
058A  5A                +                 POP       DX
058B  58                +                 POP       AX
058C  5F                +                 POP       DI
058D  5E                +                 POP       SI
058E  1F                +                 POP       DS
058F  B9 0000                          MOV       CX,GO_LIMIT
0592  F6 04 08             CLR_NOW?:   TEST      BYTE PTR DS:[SI],LANA_GETTING_NCB
0595  74 05                            JZ        OK_CPLT?
0597  E2 F9                            LOOP      CLR_NOW?
0599  EB 06 90                         JMP       NCB_ERROR

                                       ; HOW DID THE XFER COMPLETE?

059C  F6 04 06             OK_CPLT?:   TEST      DS:BYTE PTR [SI],CPLT_CODE
059F  74 03                            JZ        PROG_EXIT

                                       ; ERROR XFERRING NCB TO LANA

05A1  E8 05A6 R            NCB_ERROR:  CALL      CATASTROPHIC_ERROR

                                       ; RETURN

05A4                       PROG_EXIT:  RESTORE   <BX>
05A4  5B                +                 POP       BX
05A5  C3                               RET

05A6                       PROGRAM_IO_NCB ENDP


                           ;________________________________________________________________
                           ;                                                               |
                           ; CATASTROPHIC_ERROR                                            |
                           ;                                                               |
                           ;    HANDLES CATASTROPHIC INTERFACE ERRORS.                     |
                           ;                                                               |
                           ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:               |
                           ;                                                               |
                           ;       REGISTER   |  ACCESS  |              USAGE              |
                           ;       -----------+----------+-------------------------------- |
                           ;       DS         |   CONST  | LO_MEM_SEG                      |
                           ;       DS:SI      |   CONST  | LANA_X_STATUS @                 |
                           ;          DX      |   CONST  | LANA'S BASE PORT (LANA_X_SR) @  |
                           ;          DI      |   CONST  | LANA_X_HIR PORT @               |
                           ;          AL      |  DESTROY | INTERNAL                        |
                           ;          AH      |    VAR   | NETBIOS RETURN CODE:            |
                           ;                             |    NCBSYS_ERR? (ALWAYS!)        |
                           ;                                                               |
                           ;_______________________________________________________________|

05A6                       CATASTROPHIC_ERROR PROC       NEAR

                                       ; RELEASE THE INTERFACE

05A6  87 FA                            XCHG      DI,DX                ;POINT TO HCR
05A8  EC                               IN        AL,DX
05A9  24 FD                            AND       AL,ALL_BITS-HCR      ;CLEAR HCR
05AB  EB 00                            JMP       $+2
05AD  EE                               OUT       DX,AL
05AE  87 D7                            XCHG      DX,DI

                                       ; SET HARDWARE ERROR FOR THIS LANA

05B0  80 0C 04                         OR        BYTE PTR DS:[SI],LANA_HARD_ERR  ;SET FLAG HARDWARE ERROR IN LANA_X

                                       ; UNLOCK INTERFACE

05B3  80 24 BF                         AND       BYTE PTR DS:[SI],ALL_BITS-LANA_LOCKED  ;CLEAR INTERFACE BUSY FLAG

                                       ; RETURN

05B6  B4 40                            MOV       AH,NCBSYS_ERR?       ;REPORT TIMEOUT ERROR
05B8  C3                               RET

05B9                       CATASTROPHIC_ERROR ENDP
```

```
                        ;------------------------------------------------------------------------
                        ; ADJUST_NCB                                                             |
                        ;                                                                        |
                        ;    CHANGES BUFFER@ TO 32-BIT ADDR.                                     |
                        ;                                                                        |
                        ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                        |
                        ;                                                                        |
                        ;       REGISTER   |  ACCESS  |              USAGE                       |
                        ;      ------------+----------+----------------------------------------  |
                        ;       ES:BX      |   CONST  | NCB @                                    |
                        ;          CL      |   CONST  | NCB_COMMAND                              |
                        ;                                                                        |
                        ;------------------------------------------------------------------------|

05B9                    ADJUST_NCB PROC         NEAR

                                SAVE            <AX,CX,CX>
05B9  50            +              PUSH         AX
05BA  51            +              PUSH         CX
05BB  51            +              PUSH         CX

                                ; SETUP NCB RESERVE AREA

05BC                    SET_RESERVE:
05BC  26: 89 5F 34              MOV     ES:[BX].NCB_RESERVE_NCB@,BX          ; FOR POSTING
05C0  26: 8C 47 36              MOV     ES:[BX].NCB_RESERVE_NCB@+2,ES
05C4  26: 8B 47 04              MOV     AX,ES:[BX].NCB_BUFFER@               ; TO RESTORE SEG:OFF TYPE ADDR ON CPLT
05C8  26: 89 47 38              MOV     ES:[BX].NCB_RESERVE_BUFFER@,AX
05CC  26: 8B 47 06              MOV     AX,ES:[BX].NCB_BUFFER@+2
05D0  26: 89 47 3A              MOV     ES:[BX].NCB_RESERVE_BUFFER@+2,AX

                                ; CHANGE BUFFER@ TO 32-BIT ADDRESS

05D4  26: 8B 47 06              MOV     AX,ES:WORD PTR [BX].NCB_BUFFER@+2    ; AX=SEG{ A<19::16>, A<15::04> }
05D8  B9 0004                   MOV     CX,4                                 ; CH=0,  CL=ROL COUNT
05DB  D3 C0                     ROL     AX,CL                                ; AX=SEG{ A<15::04>, A<19::16> }
05DD  8A C8                     MOV     CL,AL                                ; CX=SEG{     0,     X, A<19::16> }
05DF  80 E1 0F                  AND     CL,00FH                              ; CX=SEG{     0,     0, A<19::16> }
05E2  24 F0                     AND     AL,0F0H                              ; AX=SEG{ A<15::04>,            0 }
                                                                             ;
05E4  26: 03 47 04              ADD     AX,ES:WORD PTR [BX].NCB_BUFFER@      ; AX=REAL{ A<15::00> } & CF=SEGMENT ADJ.
05E8  12 CD                     ADC     CL,CH                                ; CX=REAL{ A<19::16> }
05EA  26: 89 47 04              MOV     ES:WORD PTR [BX].NCB_BUFFER@,AX
05EE  26: 89 4F 06              MOV     ES:WORD PTR [BX].NCB_BUFFER@+2,CX

05F2  59                        POP     CX
05F3  80 E1 7F                  AND     CL,7FH                               ; MASK OFF HIGH ORDER BIT
05F6  80 F9 17                  CMP     CL,NCBSENDMULTIPLE                   ; IS COMMAND IS MULTIPLE SEND
05F9  75 2E                     JNZ     ADJUST_NCB_EXIT                      ; NO THEN LEAVE

                                                                             ; OTHERWISE TAKE CARE OF SECOND BUFFER ADD
                        R.
05FB  26: 8B 47 0C              MOV     AX,ES:[BX].NCB_BUFFER@+8             ; TO RESTORE SEG:OFF TYPE ADDR ON CPLT
05FF  26: 89 47 3C              MOV     ES:[BX].NCB_RESERVE_BUFFER2@,AX
0603  26: 8B 47 0E              MOV     AX,ES:[BX].NCB_BUFFER@+0AH
0607  26: 89 47 3E              MOV     ES:[BX].NCB_RESERVE_BUFFER2@+2,AX

                                ; CHANGE BUFFER@ TO 32-BIT ADDRESS
060B  26: 8B 47 0E              MOV     AX,ES:WORD PTR [BX].NCB_BUFFER@+0AH; AX=SEG{ A<19::16>, A<15::04> }
060F  B9 0004                   MOV     CX,4                                 ; CH=0,  CL=ROL COUNT
0612  D3 C0                     ROL     AX,CL                                ; AX=SEG{ A<15::04>, A<19::16> }
0614  8A C8                     MOV     CL,AL                                ; CX=SEG{     0,     X, A<19::16> }
0616  80 E1 0F                  AND     CL,00FH                              ; CX=SEG{     0,     0, A<19::16> }
0619  24 F0                     AND     AL,0F0H                              ; AX=SEG{ A<15::04>,            0 }
                                                                             ;
061B  26: 03 47 0C              ADD     AX,ES:WORD PTR [BX].NCB_BUFFER@+8    ; AX=REAL{ A<15::00> } & CF=SEGMENT ADJ.
061F  12 CD                     ADC     CL,CH                                ; CX=REAL{ A<19::16> }
0621  26: 89 47 0C              MOV     ES:WORD PTR [BX].NCB_BUFFER@+8,AX
0625  26: 89 4F 0E              MOV     ES:WORD PTR [BX].NCB_BUFFER@+0AH,CX

0629                    ADJUST_NCB_EXIT:

                                ; RETURN

                                RESTORE         <CX,AX>
0629  59            +              POP          CX
062A  58            +              POP          AX
062B  C3                        RET

062C                    ADJUST_NCB ENDP

                        ;------------------------------------------------------------------------
                        ; GET_INTERFACE                                                          |
                        ;                                                                        |
                        ;    GETS & LOCKS THE SELECTED LANA'S INTERFACE.  IF CQF IS SET AND THE  |
                        ; CMD ISN'T RESET OR CANCEL, AN ERROR IS RETURNED.                       |
                        ;                                                                        |
                        ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                        |
                        ;                                                                        |
                        ;       REGISTER   |  ACCESS  |              USAGE                       |
                        ;      ------------+----------+----------------------------------------  |
                        ;       DS         |   CONST  | LO_MEM_SEG                               |
                        ;       DS:SI      |   CONST  | LANA_X_STATUS @                          |
                        ;          DX      |   CONST  | LANA'S BASE PORT (LANA_X_SR) @           |
                        ;          DI      |   CONST  | LANA_X_HIR PORT @                        |
                        ;          CL      |   CONST  | NCB_COMMAND                              |
                        ;          AL      |  DESTROY | INTERNAL                                 |
                        ;          AH      |    VAR   | NETBIOS RETURN CODE:                     |
                        ;                  |          |     NCBMAX_CMD?                          |
                        ;                  |          |     NCBSYS_ERR?                          |
                        ;                                                                        |
                        ; INTERRUPTS SHOULD BE MASKED ON ENTRY.                                  |
                        ; INTERRUPTS WILL BE UNMASKED ON EXIT.                                   |
                        ;                                                                        |
                        ;------------------------------------------------------------------------|

062C                    GET_INTERFACE PROC      NEAR

                                SAVE            <BX,CX>
062C  53            +              PUSH         BX
062D  51            +              PUSH         CX

                                ; LOCK INTERFACE & ENABLE INTERRUPTS

062E  80 0C 40                  OR      BYTE PTR DS:[SI],LANA_LOCKED ;SET INTERFACE BUSY FLAG
0631  FB                        STI                                  ; MAINLY TO ALLOW DMA COMPLETION

                                ; IF LANA'S DMAING, GIVE IT A CHANCE TO FINISH
```

```
0632  F6 04 20                  TEST     BYTE PTR DS:[SI],LANA_DMAING ;CHECK LANA_X DMAIN FLAG
0635  74 08                     JZ       A_RESET?                     ;NOT DMAING CHECK IF WE HAVE A RESET COMMAND
0637  E8 0688 R                 CALL     WAIT_ON_DMA                  ;IF DMAING GO AND WAIT A WHILE FOR DMA TO FINISH
063A  80 FC 00                  CMP      AH,NCBGOOD_RET?              ; TIMEOUT ON DMA?
063D  75 3E                     JNE      TIMEOUT                      ; IF SO, COMPLAIN

                                ; DON'T CHECK CQF IF RESET OR CANCEL "CMD"

063F  80 F9 32       A_RESET?:  CMP      CL,NCBRESET                  ;IS IT A RESET COMMAND
0642  74 12                     JE       REQUEST_IT                   ;IF IT IS THEN THEN GET INTERFACE AND DO IT
0644  80 F9 35                  CMP      CL,NCBCANCEL                 ;IS IT A CANCEL COMMAND
0647  74 0D                     JE       REQUEST_IT                   ;IF IT IS THEN GET INTERFACE AND DO IT

                                ; ERROR IF CQF SET.

0649  EC             CQF?:      IN       AL,DX                        ; GET LANA'S SR VALUE
064A  A8 08                     TEST     AL,CQF                       ;IS THERE ROOM FOR MORE COMMANDS
064C  74 08                     JZ       REQUEST_IT                   ;YES THEN GO REQUEST FOR THE INTERFACE
064E  80 24 BF                  AND      BYTE PTR DS:[SI],ALL_BITS-LANA_LOCKED ; UNLOCK THE INTERFACE
0651  B4 22                     MOV      AH,NCBMAX_CMD?               ;ERROR MAX. NUMBER OF CMDS OUTSTANDING
0653  EB 30 90                  JMP      GET_EXIT                     ;LEAVE AND REPORT ERROR

                                ; REQUEST INTERFACE OWNERSHIP

0656  87 FA          REQUEST_IT:XCHG     DI,DX                        ;POINT TO HIR
0658  FA                        CLI                                   ;DISABLE INTERRUPTS
0659  EC                        IN       AL,DX                        ;
065A  0C 02                     OR       AL,HCR                       ;INFORM LANA THAT WE WANT INTERFACE
065C  EB 00                     JMP      $+2
065E  EE                        OUT      DX,AL
065F  FB                        STI                                   ;ENABLE INTERRUPTS
0660  87 D7                     XCHG     DX,DI

                                ; SETUP TO WAIT ON HC

0662  BB 000A        WAIT_ON_HC:MOV      BX,10                        ;SET TIMEOUT VALUE
0665  B9 0000        HC_LOOP:   MOV      CX,HC_LIMIT                  ;SET TIMEOUT VALUE

                                ; IS HC SET YET?

0668  EB 00                     JMP      $+2
066A  EC             HC_SET?:   IN       AL,DX                        ;POINT TO SR
066B  A8 80                     TEST     AL,HC                        ;DID WE GET THE INTERFACE
066D  75 16                     JNZ      GET_EXIT                     ; IF SO, WE'RE DONE
066F  87 FA                     XCHG     DI,DX                        ;POINT TO HIR
0671  EC                        IN       AL,DX                        ;
0672  A8 02                     TEST     AL,HCR                       ;
0674  87 D7                     XCHG     DX,DI                        ;
0676  74 DE                     JZ       REQUEST_IT                   ;
0678  E2 F0                     LOOP     HC_SET?                      ;
067A  4B                        DEC      BX                           ;
067B  75 E8                     JNZ      HC_LOOP                      ;

                                ; HANDLE TIMEOUT

067D                 TIMEOUT:   RESTORE  <CX,BX>
067D  59           +               POP      CX
067E  5B           +               POP      BX
067F  E8 05A6 R                 CALL     CATASTROPHIC_ERROR
0682  EB 03 90                  JMP      GET_EXIT2

                                ; RETURN
0685                 GET_EXIT:  RESTORE  <CX,BX>
0685  59           +               POP      CX
0686  5B           +               POP      BX
0687  C3             GET_EXIT2: RET

0688                 GET_INTERFACE ENDP

                     ;________________________________________________________________
                     ; WAIT_ON_DMA                                                    |
                     ;                                                                |
                     ;    WAIT UP TO "X" FOR LANA TO FINISH DMAING.                   |
                     ;                                                                |
                     ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                |
                     ;                                                                |
                     ;       REGISTER   |  ACCESS  |            USAGE                 |
                     ;     -------------+----------+--------------------------------  |
                     ;       DS:SI      |   CONST  | LANA_X_STATUS @                  |
                     ;         AH       |    VAR   | NETBIOS RETURN CODE:             |
                     ;                  |          |     NCBSYS_ERR?                  |
                     ;                                                                |
                     ; INTERRUPTS SHOULD BE UNMASKED ON ENTRY.                        |
                     ;                                                                |
                     ;________________________________________________________________|

0688                 WAIT_ON_DMA PROC      NEAR

                                SAVE     <CX>
0688  51           +               PUSH     CX

                                ; SETUP DMA TIME LIMIT

0689  B9 0000                   MOV      CX,DMA_LIMIT                 ;SET UP TIMEOUT VALUE

                                ; IS LANA DMA DONE YET?

068C  F6 04 20       DMA_DONE?: TEST     BYTE PTR DS:[SI],LANA_DMAING ;IS LANA_X DONE DMAING
068F  74 04                     JZ       WAIT_EXIT                    ; IF SO, LEAVE
0691  E2 F9                     LOOP     DMA_DONE?                    ;WAIT SOME MORE

                                ; TIMEOUT ON DMA WAIT

0693  B4 40                     MOV      AH,NCBSYS_ERR?               ;IF TIMEOUT WAITING FOR DMA TO
                                                                      ;COMPLETE REPORT ERROR

                                ; RETURN

0695                 WAIT_EXIT: RESTORE  <CX>
0695  59           +               POP      CX
0696  C3                        RET

0697                 WAIT_ON_DMA ENDP

                     ;-----------------------------------------------------------------|
                     ; HARD_FILE:                                                      |
                     ;          THIS ROUTINE ALLOWS THE SHARING OF DMA CHANNEL 3 BETWEEN LANA |
                     ;          AND HARD_DISK ON PCXT.                                 |
                     ;          INT 13H IS SETUP TO POINT TO HERE ON PC1 AND PCXT      |
                     ;                                                                 |
                     ;                                                                 |
                     ;-----------------------------------------------------------------|
```

```
0697                          HARD_FILE  PROC      NEAR

0697  FB                                 STI

0698  1E                                 PUSH      DS                      ;
0699  50                                 PUSH      AX                      ;
069A  B8 0040                            MOV       AX,LO_MEM_SEG           ;
069D  8E D8                              MOV       DS,AX                   ;POINT TO SEGMENT 40H
069F  58                                 POP       AX                      ;

06A0  80 FA 80                           CMP       DL,80H                  ;TEST IF FIXED DISK
06A3  72 3C                              JNAE      DISKETTE                ;IF DISKETTE NO NEED TO CHECK FOR DMA SHARING

                                         ;
                                         ; CHECK FOR BUSY DMA

06A5  51                                 PUSH      CX                      ;
06A6  B9 FFFF                            MOV       CX,0FFFFH               ;SET UP TIMEOUT VALUE

06A9  F6 06 00A1 80           TRY_DMA_3H: TEST     BYTE PTR DS:[LANA_HARDWARE],DMA_3_BUSY ;CHECK TO SEE IF DMA BUSY
06AE  74 1A                              JZ        GET_DMA_3H              ;NOT BUSY GO AND GET HOLD OF IT
06B0  49                                 DEC       CX
06B1  75 F6                              JNZ       TRY_DMA_3H              ;WAIT SOME MORE
06B3  1E                                 PUSH      DS
06B4  50                                 PUSH      AX
06B5  B8 0040                            MOV       AX,LO_MEM_SEG           ;POINT TO SEGMENT 40H
06B8  8E D8                              MOV       DS,AX
06BA  58                                 POP       AX
06BB  8D 36 00A2                         LEA       SI,DS:LANA_0_STATUS     ;
06BF  80 0C 04                           OR        BYTE PTR DS:[SI],LANA_HARD_ERR ;IF TIMEOUT REPORT ERROR ON LANA_0
06C2  8D 36 00A3                         LEA       SI,DS:LANA_1_STATUS     ;
06C6  80 0C 04                           OR        BYTE PTR DS:[SI],LANA_HARD_ERR ;AND LANA_1
06C9  1F                                 POP       DS                      ;THEN GO AND PERFORM DISK REQUEST

06CA  59                      GET_DMA_3H: POP      CX
06CB  80 0E 00A1 80                       OR       BYTE PTR DS:[LANA_HARDWARE],DMA_3_BUSY ;SET DMA BUSY FLAG

06D0  9C                                 PUSHF
06D1  FA                                 CLI
06D2  FF 1E 00A4                         CALL       DWORD PTR DS:[LANA_HARD_INT]   ;GO SERVICE HARD DISK REQUEST
06D6  9C                                 PUSHF
06D7  80 26 00A1 7F                      AND        BYTE PTR DS:[LANA_HARDWARE],ALL_BITS-DMA_3_BUSY ;RELEASE DMA
06DC  9D                                 POPF
06DD  1F                                 POP        DS

06DE  EB 08 90                           JMP       HARD_FILE_END

06E1  9C                      DISKETTE:  PUSHF
06E2  FA                                 CLI
06E3  FF 1E 00A4                         CALL       DWORD PTR DS:[LANA_HARD_INT]  ;GO PERFORM DISKETTE REQUEST
06E7  1F                                 POP        DS

06E8                          HARD_FILE_END:

06E8                          HARD_FILE  ENDP

06E8                          DUMMY      PROC      FAR

06E8  CA 0002                            RET       2

06EB                          DUMMY      ENDP
                              ;-----------------------------------------------------------------------------
                              ;
                              ;     RIPL_INT13:REDIRECTED INTERRUPT 13H FOR REMOTE IPL
                              ;            THIS ROUTINE SENDS ALL REGISTERS USED FOR INT 13H ACROSS THE
                              ;            TO SOME FORM OF A DISKETTE SERVER THAT WILL HANDLE THIS REQUESTS
                              ;
                              ;
                              ;
                              ;-----------------------------------------------------------------------------
06EB                          RIPL_INT13 PROC      NEAR

                                         SAVE      <DS,SI,DI>
06EB  1E                      +              PUSH      DS
06EC  56                      +              PUSH      SI
06ED  57                      +              PUSH      DI

                                         ; CHECK FOR INT 13H REQUEST NOT SUPPORTED

06EE  80 FC 05                           CMP       AH,05                   ;IS IT FORMAT DESIRED TRACK
06F1  74 14                              JZ        BAD_CMD_I13             ;YES,GO REPORT INVALID COMMAND
06F3  80 FC 06                           CMP       AH,06                   ;
06F6  74 0F                              JZ        BAD_CMD_I13             ;
06F8  80 FC 07                           CMP       AH,07                   ;
06FB  74 0A                              JZ        BAD_CMD_I13             ;
06FD  80 FC 0A                           CMP       AH,0AH                  ;IS IT A READ LONG
0700  74 05                              JZ        BAD_CMD_I13             ;YES,GO REPORT INVALID COMMAND
0702  80 FC 0B                           CMP       AH,0BH                  ;IS IT A WRITE LONG
0705  75 06                              JNZ       DO_REQUEST              ;NO,GO AND PERFORM REQUEST

0707                          BAD_CMD_I13:

0707  B4 01                              MOV       AH,01                   ;ERROR BAD COMMAND
0709  F9                                 STC                               ;SET CARRY FLAG TO INDICATE ERROR
070A  E9 0895 R                          JMP       RIPL_INT13_EXIT         ;GO REPORT ERROR

070D                          DO_REQUEST:
070D  1E                                 PUSH      DS                      ;
070E  50                                 PUSH      AX                      ;
070F  B8 0040                            MOV       AX,LO_MEM_SEG           ;POINT TO SEGMENT 40H
0712  8E D8                              MOV       DS,AX                   ;

0714  F6 06 00A1 40                      TEST      BYTE PTR DS:[LANA_HARDWARE],ACTV_RIPL ;IS REMOTE IPL ACTIVE
0719  58                                 POP       AX
071A  1F                                 POP       DS
071B  75 12                              JNZ       RDIR                    ;YES,THEN REDIRECT INT 13H REQUEST
071D  1E                                 PUSH      DS
071E  50                                 PUSH      AX
071F  B8 0040                            MOV       AX,LO_MEM_SEG           ;POINT TO SEGMENT 40H
0722  8E D8                              MOV       DS,AX
0724  58                                 POP       AX
0725  9C                                 PUSHF
0726  FA                                 CLI
0727  FF 1E 00A4                         CALL      DWORD PTR DS:[LANA_HARD_INT]  ;HANDLE INT 13H LOCALLY
072B  1F                                 POP       DS
072C  E9 0885 R                          JMP       END_RDRC                ;DONE THEN LEAVE

072F                           RDIR:     SAVE      <AX,BX,ES>
072F  50                      +              PUSH      AX
0730  53                      +              PUSH      BX
0731  06                      +              PUSH      ES
```

```
'32  E8 089B R                      CALL    CALC_TOPM                  ;CALCULATE ADDRESS FOR NCB AND BUFFER USED BY RPL

'35  26: C6 07 14                   MOV     ES:[BX].NCB_COMMAND,NCBSEND;SET NCB WITH SEND COMMAND

739  8B C3                          MOV     AX,BX
73B  05 0041                        ADD     AX,LEN_NCB                 ;BUFFER AREA LOACTED NEXT TO NCB
73E  8B F0                          MOV     SI,AX

740  26: 89 47 04                   MOV     ES:[BX].NCB_BUFFER@,AX
744  8C C0                          MOV     AX,ES
746  26: 89 47 06                   MOV     ES:[BX].NCB_BUFFER@+2,AX
74A  26: C7 47 08 000B              MOV     ES:[BX].NCB_LENGTH,11      ;SEND 11 BYTES

                                    ;BUFFER CONSIST OF AX,CX,DX,ES,BX,?

750  26: 89 4C 02                   MOV     ES:WORD PTR [SI+2],CX      ;CH-TRACK NUMBER,CL-SECTOR NUMBER
754  26: 89 54 04                   MOV     ES:WORD PTR [SI+4],DX      ;DH-HEAD NUMBER,DL-DRIVE NUMBER
758  58                             POP     AX
759  26: 89 44 06                   MOV     ES:WORD PTR [SI+6],AX      ; ES VALUE
75D  58                             POP     AX
75E  26: 89 44 08                   MOV     ES:WORD PTR [SI+8],AX      ; BX VALUE
762  26: 88 44 0A                   MOV     ES:BYTE PTR [SI+0AH],AL
766  58                             POP     AX
767  26: 89 04                      MOV     ES:WORD PTR [SI],AX        ;AH-COMMAND,AL-NUMBER OF SECTORS

76A  50                             PUSH    AX

76B  CD 5C                          INT     NET_INT                    ;ISSUE A CALL TO NETBIOS TO PERFORM THE SEND

76D  26: 80 7F 01 00                CMP     ES:[BX].NCB_RETCODE,NCBGOOD_RET? ;DID IT COMPLETE OK
772  58                             POP     AX
773  74 08                          JZ      DO_RCV1                    ;YES THEN GO DO A RCV FOR A REPLY
775  F9                             STC                                ;OTHERWISE SET CARRY BIT TO INDICATE ERROR
776  B4 80                          MOV     AH,80H                     ;SET RETURN CODE TO ATTACHMENT FAILED TO REPOND
778  B0 00                          MOV     AL,00H
77A  E9 087F R                      JMP     END_RDRC_E                 ;LEAVE AND REPORT ERROR

77D  80 FC 02            DO_RCV1:   CMP     AH,02H                     ;IS IT A READ DATA
780  75 6C                          JNZ     DO_SND?                    ;NO THEN IS IT A WRITE DATA
782  26: C6 07 15                   MOV     ES:[BX].NCB_COMMAND,NCBRECEIVE
786  8B C6                          MOV     AX,SI

788  26: 89 47 04                   MOV     ES:[BX].NCB_BUFFER@,AX
78C  8C C0                          MOV     AX,ES
78E  26: 89 47 06                   MOV     ES:[BX].NCB_BUFFER@+2,AX
792  26: C7 47 08 000B              MOV     ES:[BX].NCB_LENGTH,11

798  CD 5C                          INT     NET_INT

79A  26: 80 7F 01 00                CMP     ES:[BX].NCB_RETCODE,NCBGOOD_RET?
79F  26: 8B 04                      MOV     AX,ES:WORD PTR [SI]
7A2  75 03                          JNZ     MSG_INCPLT?
7A4  E9 087F R                      JMP     END_RDRC_E

07A7                     MSG_INCPLT?:

07A7 26: 80 7F 01 06                CMP     ES:[BX].NCB_RETCODE,06H    ;MESSAGE INCOMPLETE
07AC 74 08                          JZ      DO_RCV2                    ;YES,THEN GO AND DO OTHER RCV
07AE F9                             STC                                ;OTHERWISE SET CARRY BIT TO INDICATE ERROR
07AF B4 80                          MOV     AH,80H                     ;SET RETURN CODE TO ATTACHMENT FAILED TO REPOND
07B1 B0 00                          MOV     AL,00H                     ;LEAVE AND REPORT ERROR
07B3 E9 087F R                      JMP     END_RDRC_E

07B6 26: C6 07 15        DO_RCV2:   MOV     ES:[BX].NCB_COMMAND,NCBRECEIVE
07BA 26: 8B 44 08                   MOV     AX,ES:WORD PTR [SI+8]

07BE 26: 89 47 04                   MOV     ES:[BX].NCB_BUFFER@,AX
07C2 26: 8B 44 06                   MOV     AX,ES:WORD PTR [SI+6]          ; BX
07C6 26: 89 47 06                   MOV     ES:[BX].NCB_BUFFER@+2,AX
07CA 26: 8B 04                      MOV     AX,ES:WORD PTR [SI]             ; AZ
07CD B4 00                          MOV     AH,00
07CF B1 09                          MOV     CL,09                          ;CALCULATE LENGTH OF BUFFER EXPECTED
07D1 D3 E0                          SHL     AX,CL                          ;NUMBER OF SECTORS*512 BYTES/SECTOR
07D3 26: 89 47 08                   MOV     ES:[BX].NCB_LENGTH,AX

07D7 CD 5C                          INT     NET_INT

07D9 26: 80 7F 01 00                CMP     ES:[BX].NCB_RETCODE,NCBGOOD_RET?  ;DID IT COMPLETE OK
07DE 26: 8B 04                      MOV     AX,ES:WORD PTR [SI]
07E1 75 03                          JNZ     ERROR_R?

07E3 E9 0885 R                      JMP     END_RDRC
07E6 F9                  ERROR_R?:  STC                                ;SET CARRY BIT TO INDICATE ERROR CONDITION
07E7 B4 80                          MOV     AH,80H                     ;SET RETURN CODE TO ATTACHMENT FAILED TO REPOND
07E9 B0 00                          MOV     AL,00H
07EB E9 0885 R                      JMP     END_RDRC                   ;LEAVE AND REPORT ERROR

07EE 80 FC 03            DO_SND?:   CMP     AH,03                      ;IS IT A WRITE DATA
07F1 75 5D                          JNZ     SM_ELSE                    ;CHECK FOR OTHER REQUESTS
07F3 26: C6 07 14        DO_SND:    MOV     ES:[BX].NCB_COMMAND,NCBSEND;YES,THEN SEND DATA TO BE USED FOR WRITE
07F7 26: 8B 44 08                   MOV     AX,ES:WORD PTR [SI+8]          ; BX

07FB 26: 89 47 04                   MOV     ES:[BX].NCB_BUFFER@,AX
07FF 26: 8B 44 06                   MOV     AX,ES:WORD PTR [SI+6]          ; ES
0803 26: 89 47 06                   MOV     ES:[BX].NCB_BUFFER@+2,AX
0807 26: 8B 04                      MOV     AX,ES:WORD PTR [SI]            ; AX
080A B4 00                          MOV     AH,00
080C B1 09                          MOV     CL,09                          ;CALCULATE LENGTH OF BUFFER EXPECTED
080E D3 C0                          ROL     AX,CL                          ;NUMBER OF SECTORS*512 BYTES/SECTOR
0810 26: 89 47 08                   MOV     ES:[BX].NCB_LENGTH,AX

0814 CD 5C                          INT     NET_INT                        ;ISSUE A CALL TO NETBIOS

0816 26: 80 7F 01 00                CMP     ES:[BX].NCB_RETCODE,NCBGOOD_RET? ;DID IT COMPLETE OK

081B 75 2B                          JNZ     ERROR_W?                   ;NO THEN GO AND REPORT ERROR

081D 26: C6 07 15                   MOV     ES:[BX].NCB_COMMAND,NCBRECEIVE ;YES, THEN DO A RECEIVE TO REGISTERS
                                                                           ;USED IN THE OPERATION

0821 8B C6                          MOV     AX,SI

0823 26: 89 47 04                   MOV     ES:[BX].NCB_BUFFER@,AX
0827 8C C0                          MOV     AX,ES
0829 26: 89 47 06                   MOV     ES:[BX].NCB_BUFFER@+2,AX
082D 26: C7 47 08 000B              MOV     ES:[BX].NCB_LENGTH,11      ;ASK FOR 11 BYTES

0833 CD 5C                          INT     NET_INT

0835 26: 80 7F 01 00                CMP     ES:[BX].NCB_RETCODE,NCBGOOD_RET?
083A 26: 8B 04                      MOV     AX,ES:WORD PTR [SI]
083D 75 09                          JNZ     ERROR_W?
083F 80 FC 00                       CMP     AH,00
0842 74 41                          JZ      END_RDRC
0844 F9                             STC
```

```
0845  EB 3E 90                              JMP         END_RDRC

0848  F9                    ERROR_W?:       STC                                     ;SET CARRY BIT TO INDICATE ERROR CONDITION
0849  B4 80                                 MOV         AH,80H                      ;SET RETURN CODE TO ATTACHMENT FAILED TO REPONI
084B  B0 00                                 MOV         AL,00H
084D  EB 36 90                              JMP         END_RDRC                    ;LEAVE AND REPORT ERROR

                                            ;DO A RECEIVE OF 11 BYTES TO STATUS OF INT 13H REQUEST

0850                        SM_ELSE:
0850  26: C6 07 15                          MOV         ES:[BX].NCB_COMMAND,NCBRECEIVE
0854  8B C6                                 MOV         AX,SI

0856  26: 89 47 04                          MOV         ES:[BX].NCB_BUFFER@,AX
085A  8C C0                                 MOV         AX,ES
085C  26: 89 47 06                          MOV         ES:[BX].NCB_BUFFER@+2,AX
0860  26: C7 47 08 000B                     MOV         ES:[BX].NCB_LENGTH,11

0866  CD 5C                                 INT         NET_INT

0868  26: 80 7F 01 00                       CMP         ES:[BX].NCB_RETCODE,NCBGOOD_RET?
086D  26: 8B 04                             MOV         AX,ES:WORD PTR [SI]
0870  26: 8A 5C 0A                          MOV         BL,ES:BYTE PTR [SI+10]
0874  75 D2                                 JNZ         ERROR_W?
0876  80 FB 00                              CMP         BL,00
0879  74 0A                                 JZ          END_RDRC
087B  F9                                    STC

087C  EB 07 90                              JMP         END_RDRC

087F                        END_RDRC_E:
087F  80 FC 00                                 CMP      AH,00H
0882  74 01                                    JZ       END_RDRC
0884  F9                                       STC

                                            ;RESTORE REGISTERS

0885  26: 8B 4C 02          END_RDRC:          MOV      CX,ES:WORD PTR [SI+2]
0889  26: 8B 54 04                             MOV      DX,ES:WORD PTR [SI+4]
088D  26: 8B 5C 08                             MOV      BX,ES:WORD PTR [SI+8]
0891  26: 8E 44 06                             MOV      ES,ES:WORD PTR [SI+6]

0895                        RIPL_INT13_EXIT:

                                               RESTORE  <DI,SI,DS>
0895  5F                  +                      POP      DI
0896  5E                  +                      POP      SI
0897  1F                  +                      POP      DS

0898                        RIPL_INT13     ENDP

0898                        DUMMY1          PROC        FAR

0898  CA 0002                               RET         2

089B                        DUMMY1          ENDP

                            ;------------------------------------------------------------------------------
                            ; CALC_TOPM:
                            ;               LOOK IN LOW STRORAGE (0040H:0013H) AND GET NUMBER OF 1K-BYTES ON
                            ;               THIS MACHINE MULTIPLY BY 1024 TO GET NUMBER OF K-BYTES.
                            ;               1K-BYTES HAVE ALREADY BEEN SUBTRACTED FROM THIS VALUE TO BE USED
                            ;               FOR NCBS AND BUFFER SPACE FOR RPL.
                            ;               ES:BX WILL POINT TO BEGINNING OF THIS MEMORY SPACE
                            ;
                            ;------------------------------------------------------------------------------
089B                        CALC_TOPM   PROC        NEAR

089B  1E                                PUSH        DS
089C  51                                PUSH        CX
089D  52                                PUSH        DX
089E  50                                PUSH        AX

089F  B8 0040                           MOV         AX,LO_MEM_SEG
08A2  8E D8                             MOV         DS,AX                       ;POINT TO SEGMENT 40H
08A4  A1 0013                           MOV         AX,WORD PTR DS:MEMORY_SIZE  ;LOCATION WHERE MEMMORY SIZE IS RECORDED

08A7  0E                                PUSH        CS
08A8  1F                                POP         DS

08A9  F7 26 000F R                      MUL         WORD PTR DS:T1K             ;MULTYPLY BY 1024 TO GET K-BYTES
08AD  B1 0C                             MOV         CL,0CH
08AF  D3 E2                             SHL         DX,CL
08B1  8E C2                             MOV         ES,DX
08B3  8B D8                             MOV         BX,AX
08B5  58                                POP         AX
08B6  5A                                POP         DX
08B7  59                                POP         CX
08B8  1F                                POP         DS

08B9  C3                                RET

08BA                        CALC_TOPM   ENDP

                            ;------------------------------------------------------------------------------
                            ;
                            ;   CHG_RDRC:
                            ;               USED BY UNLINK COMMAND TO STOP REDIRECTION FO INT 13H REQUESTS
                            ;               FROM GOING OUT TO THE NETWORK.
                            ;
                            ;
                            ;------------------------------------------------------------------------------
08BA                        CHG_RDRC    PROC        NEAR

                                        SAVE        <DS,ES,BX>
08BA  1E                  +                PUSH        DS
08BB  06                  +                PUSH        ES
08BC  53                  +                PUSH        BX

08BD  F6 06 00A1 40                     TEST        BYTE PTR DS:[LANA_HARDWARE],ACTV_RIPL ;DO ONLY IF RPL IS ACTIVE
08C2  74 10                             JZ          END_CHG_RDRC                ;NOT ACTIVE THEN LEAVE
08C4  80 26 00A1 BF                     AND         BYTE PTR DS:[LANA_HARDWARE],ALL_BITS-ACTV_RIPL ;CLEAR RPL ACTIVE FLAG

08C9  E8 089B R                         CALL       CALC_TOPM                    ;LOCATE RPL WORKING AREA
08CC  26: C6 07 12                      MOV         ES:[BX].NCB_COMMAND,NCBHANGUP ;CLOSE THE SESSION WITH IBMNETBOOT
08D0  CD 5C                             INT         NET_INT

08D2  B4 00                             MOV         AH,NCBGOOD_RET?             ;ASSUME GOOD RETURN
```

```
3D4                          END_CHG_RDRC:

                                    RESTORE     <BX,ES,DS>
3D4  5B                  +              POP         BX
3D5  07                  +              POP         ES
3D6  1F                  +              POP         DS

3D7  C3                             RET

3D8                          CHG_RDRC     ENDP

                             ;-------------------------------------------------------------------------------|
                             ;                                                                               |
                             ;   REM_IPL:                                                                    |
                             ;          USED IF PC FAILS TO BOOT FROM DISKETTE OR HARDDISK.                  |
                             ;          THIS ROUTINE WILL TRY TO BOOT FROM THE NETWORK                       |
                             ;          ASSUMING THAT W1 JUMPER HAS BEEN REMOVE FROM THE CARD                |
                             ;          OTHERWISE IT WILL GO TO ROM BASIC                                    |
                             ;                                                                               |
                             ;-------------------------------------------------------------------------------|
8D8                          REM_IPL      PROC        NEAR
8D8  FB                                   STI                                  ;ENABLE INTERRUPTS
8D9  B8 0000                              MOV         AX,INTERRUPT_VECTOR_SEGMENT
8DC  8E C0                                MOV         ES,AX
8DE  26: A1 0218                          MOV         AX,ES:INT_STORAGE         ;RESTORE INTERRUPT 18H
8E2  26: A3 0060                          MOV         ES:ROM_BASIC@,AX
8E6  26: A1 021A                          MOV         AX,ES:INT_STORAGE+2
8EA  26: A3 0062                          MOV         ES:ROM_BASIC@+2,AX
8EE  B8 0000                              MOV         AX,00H
8F1  26: A3 0218                          MOV         ES:INT_STORAGE,AX         ;SET BACK TO ZEROS
8F5  26: A3 021A                          MOV         ES:INT_STORAGE+2,AX

                                          ; CHECK FOR MEMORY IN SYSTEM AND SUBTRACT 1K
                                          ; TO BE USED BY NCBS AND BUFFER SPACE FOR RPL PROCESS

08F9  B8 0040                             MOV         AX,LO_MEM_SEG             ;POINT TO SEGMENT 40H
08FC  8E D8                               MOV         DS,AX
08FE  A1 0013                             MOV         AX,WORD PTR DS:MEMORY_SIZE ;GET VALUE FOR NUMBER OF K-BYTES IN SYSTEM
0901  2D 0001                             SUB         AX,01H                    ;TAKE AWAY 1K
0904  A3 0013                             MOV         WORD PTR DS:MEMORY_SIZE,AX ;RESTORE MEMORY SIZE MINUS 1K


0907  E8 089B R                           CALL        CALC_TOPM                 ;LOCATE RPL WORKING AREA

                                         ; CLEAR NCB AREA BEFORE USING

090A  8B FB                               MOV         DI,BX                     ;POINT TO BEGINNING OF NCB
090C  B9 0041                             MOV         CX,LEN_NCB                ;CLEAR ALL OF NCB
090F  B0 00                               MOV         AL,00H
0911  F3/ AA                              REP         STOSB


                                         ; DO AN ADAPTOR STATUS TO GET NODE ID
                                         ; IN ORDER TO FORM PERMANENT NODE NAME


0913  26: C6 07 33                        MOV         ES:[BX].NCB_COMMAND,NCBSTATUS
0917  26: C6 47 0A 2A                     MOV         ES:[BX].NCB_CALLNAME,'*'  ;LOCAL_ADAPTER STATUS
091C  8B C3                               MOV         AX,BX
091E  05 0041                             ADD         AX,LEN_NCB                ;BUFFER AREA RIGHT AFTER NCB
0921  26: 89 47 04                        MOV         ES:[BX].NCB_BUFFER@,AX
0925  8C C0                               MOV         AX,ES
0927  26: 89 47 06                        MOV         ES:[BX].NCB_BUFFER@+2,AX
092B  B8 004E                             MOV         AX,78                     ;MINIMUN LENGHT FO ADAPTER STATUS
092E  26: 89 47 08                        MOV         ES:[BX].NCB_LENGTH,AX
0932  CD 5C                               INT         NET_INT

                                          ; CHECK FOR REMOTE IPL JUMPER

0934  26: F6 44 04 40                     TEST        ES:BYTE PTR [SI+4],RIPL_OFF  ;CHECK TO SEE IF JUMPER (W1) HAS BEEN REMOVED
                                                                                   ;TO ACTIVATE RPL
0939  74 03                               JZ          DO_RESET                     ;YES,THEN GO DO A RESET FOR MAX. SESSIONS
                                                                                   ;AND MAX. COMMANDS
093B  E9 09D9 R                           JMP         DO_INT18                     ;NO,THEN GOTO ROM BASIC


                                          ; DO AN ADAPTER RESET MAX SESSION MAX COMMANDS

093E  26: C6 07 32           DO_RESET:    MOV         ES:[BX].NCB_COMMAND,NCBRESET ;DO RESET TO CONFIGURE SYSTEM FOR
0942  26: C6 47 02 20                     MOV         ES:[BX].NCB_LSN,32           ;MAX. NUMBER OF SESSIONS
0947  26: C6 47 03 20                     MOV         ES:[BX].NCB_NUM,32           ;AND MAX. NUMBER OF COMMANDS

094C  CD 5C                               INT         NET_INT


                                          ; TRY TO STABLISH A SESSION WITH DISKETTE SERVER

094E  26: 8D 7F 24                        LEA         DI,ES:[BX].NCB_NAME+10       ;FOR NAME USE PERMANENT NODE NAME
0952  26: 8B 47 04                        MOV         AX,ES:[BX].NCB_BUFFER@
0956  8B F0                               MOV         SI,AX
0958  B9 0006                             MOV         CX,06
095B  06                                  PUSH        ES
095C  1F                                  POP         DS

095D  F3/ A4                    REP       MOVSB

095F  0E                                  PUSH        CS
0960  1F                                  POP         DS
0961  2E: 8D 36 0001 R                    LEA         SI,SERVER_NAME
0966  26: C6 07 10                        MOV         ES:[BX].NCB_COMMAND,NCBCALL

096A  B9 000B                             MOV         CX,11
096D  26: 8D 7F 0A                        LEA         DI,ES:[BX].NCB_CALLNAME      ;FOR CALL NAME USE "IBMNETBOOT"

0971  F3/ A4                    REP       MOVSB

0973  B0 F0                               MOV         AL,0F0H                  ;SET TIMEOUT VALUE OF 120 SEC.
0975  26: 88 47 2A                        MOV         ES:[BX].NCB_RTO,AL       ;FOR RCV.
0979  26: 88 47 2B                        MOV         ES:[BX].NCB_STO,AL       ;AND SEND

097D  CD 5C                               INT         NET_INT

097F  26: 80 7F 01 00                     CMP         ES:[BX].NCB_RETCODE,NCBGOOD_RET? ;DID THE CONNECTION COMPLETE OK
0984  74 03                               JZ          CI13                     ;YES,THEN GO REDIRECT INT 13H
0986  EB 51 90                            JMP         DO_INT18                 ;OTHERWISE GO INTO ROM BASIC

                                          ; REDIRECT INTERRUPT 13H REQUEST TO NETWORK

0989  06                     CI13:        PUSH        ES
098A  FA                                  CLI                                  ;CLEAR INTERRUPTS WHILE WE CHANGE INT 13H
098B  B8 0000                             MOV         AX,INTERRUPT_VECTOR_SEGMENT
098E  8E C0                               MOV         ES,AX
```

```
0990  26: C7 06 004C 06EB R            MOV      ES:HARD_INT@,OFFSET RIPL_INT13 ;POINT INT 13H TO ROUTINE THAT SENDS
                                                                             ;THESE REQUESTS OVER NETWORK
0997  0E                               PUSH     CS
0998  26: 8F 06 004E                   POP      ES:HARD_INT@+2
099D  FB                               STI
099E  07                               POP      ES

099F  1E                               PUSH     DS
09A0  B8 0040                          MOV      AX,LO_MEM_SEG              ;POINT TO SEGMENT 40H
09A3  8E D8                            MOV      DS,AX
09A5  80 0E 00A1 40                    OR       BYTE PTR DS:[LANA_HARDWARE],ACTV_RIPL  ;SET UP FLAG TO INDICATE RPL ACTIVE
09AA  1F                               POP      DS

                                       ; DO A REQUEST FOR BOOT RECORD

09AB  B9 0004                          MOV      CX,04                      ; SET RETRY COUNT
09AE  51                    GET_BOOT:  PUSH     CX                         ; SAVE RETRY COUNT
09AF  B4 00                            MOV      AH,0                       ; RESET DISK
09B1  CD 13                            INT      13H                        ; DISKETTE I/O
09B3  B4 02                            MOV      AH,02                      ; READ IN A SINGLE SECTOR
09B5  BB 0000                          MOV      BX,0                       ; TO THE BOOT LOCATION
09B8  8E C3                            MOV      ES,BX
09BA  BB 7C00                          MOV      BX,7C00H                   ; BOOT LOCATION
09BD  BA 0000                          MOV      DX,0                       ; DRIVE 0, HEAD 0
09C0  B9 0001                          MOV      CX,01                      ; SECTOR 1, TRACK 0
09C3  B0 01                            MOV      AL,1                       ; READ ONE SECTOR
09C5  CD 13                            INT      13H                        ; DISKETTE I/O
09C7  59                               POP      CX
09C8  73 05                            JNC      RIPL_ACTV                  ; IF SUCCESFULL GO AND EXECUTE BOOT RECORD
09CA  E2 E2                            LOOP     GET_BOOT                   ; DO IT FOR RETRY TIMES

09CC  EB 0B 90                         JMP      DO_INT18                   ; TIMEOUT GOTO ROM BASIC

09CF                        RIPL_ACTV:
09CF  B8 0000                          MOV      AX,0000H
09D2  8E D8                            MOV      DS,AX

09D4  2E: FF 2E 000B R                 JMP      BOOT_LOCN

09D9                        DO_INT18:
                                       ; RESTORE THE 1K TAKEN OUT

09D9  B8 0040                          MOV      AX,LO_MEM_SEG
09DC  8E D8                            MOV      DS,AX
09DE  A1 0013                          MOV      AX,WORD PTR DS:MEMORY_SIZE
09E1  05 0001                          ADD      AX,01H
09E4  A3 0013                          MOV      WORD PTR DS:MEMORY_SIZE,AX
09E7  CD 18                            INT      18H                        ;GO TO ROM BASIC
09E9  CF                               IRET

09EA                        REM_IPL    ENDP

09EA                        NETWORK    ENDS
                                       END
                            TITLE LANA_0 INTERRUPT HANDLER
                                       ;
                                       ; LANA_0.ASM
                                       ;
                                       ;    LANA 0'S INTERRUPT HANDLER.

                            ;_______________________________________________________________________
                            ;                                                                       |
                            ;        MODULE : LANA_0_HNDLR                                          |
                            ;     COMPONENT : NETWORK BIOS                                          |
                            ;                                                                       |
                            ;     HANDLES THE LANA_0 (IRQ2 OR IRQ3) INTERRUPT.  VALID REASONS FOR A LANA |
                            ; INTERRUPT ARE:                                                        |
                            ;                                                                       |
                            ;         DMA COMPLETE                                                  |
                            ;         COMMAND COMPLETE   --> COULD CAUSE A NETBIOS POST             |
                            ;         LANA REQUEST FOR DATA FROM PC                                 |
                            ;         LANA REQUEST TO XFER DATA TO PC                               |
                            ;                                                                       |
                            ;                                                                       |
                            ;     ALL REGISTERS AND FLAGS ARE PRESERVED.                            |
                            ;                                                                       |
                            ;       NETBIOS.LIB CONTAINS THE NETBIOS INTERFACE EQUATES AND STRUCTURES |
                            ;       LANAS.INC CONTAINS THE LANA INTERFACE EQUATES AND STRUCTURES    |
                            ;                                                                       |
                            ;_______________________________________________________________________|

0000                        NETWORK    SEGMENT  PARA PUBLIC    'CODE'

                                       ASSUME   CS:NETWORK
                                       ASSUME   DS:NOTHING
                                       ASSUME   SS:NOTHING
                                       ASSUME   ES:NOTHING

                                       EXTRN    DMA_START_UP      : NEAR    ; STARTS UP A DMA XFER WITH A LANA

                                       PUBLIC   LANA_0_HNDLR      ; NEAR    ; LANA 0'S INTERRUPT HANDLER
                                       PUBLIC   PROGRAM_IO_CHUNK  ; NEAR    ; LANA 0'S PROGRAM I/O'ER

                            .LIST

= 00FF                      ALL_BITS        EQU   0FFH                      ; AID TO MASKING OFF BITS
= 0020                      THE_8259        EQU   20H                       ; 8259 PORT TO SEND EOI CMD TO
= 0020                      EOI             EQU   20H                       ; CMD CODE FOR EOI

= 0000                      AWHILE          EQU   0000H                     ; AMOUNT OF TIME TO WAIT ON INTERFACE
= 0070                      CHUNK_SIZE      EQU   0070H                     ; MAX. SIZE OF A PROGRAM I/O'D "CHUNK"

= 9180                      DEV_BUSY_POST   EQU   9180H

                            ;_______________________________________________________________________
                            ; LANA_0_HNDLR                                                          |
                            ;                                                                       |
                            ;     HANDLES THE LANA_0 (IRQ2 OR IRQ3) INTERRUPT.  VALID REASONS FOR A LANA |
                            ; INTERRUPT ARE:                                                        |
                            ;                                                                       |
                            ;         DMA COMPLETE                                                  |
                            ;         COMMAND COMPLETE   --> COULD CAUSE A NETBIOS POST             |
                            ;         LANA REQUEST FOR DATA FROM PC                                 |
                            ;         LANA REQUEST TO XFER DATA TO PC                               |
                            ;                                                                       |
                            ;                                                                       |
                            ;     ALL REGISTERS AND FLAGS ARE PRESERVED.                            |
                            ;                                                                       |
                            ;_______________________________________________________________________|

0000                        LANA_0_HNDLR PROC   NEAR
```

```
                                    SAVE      <AX,DX,DS>
0000  50              +                PUSH      AX
0001  52              +                PUSH      DX
0002  1E              +                PUSH      DS

                                    ; SET UP GLOBAL ASSUMPTIONS

0003  FC                            CLD                                ; ALL STRINGS GO UP
0004  B8 0040                       MOV       AX,LO_MEM_SEG            ; POINT TO LO_MEM_SEG LATER
0007  8E D8                         MOV       DS,AX                    ; DS PTS TO LANA RESERVED MEMORY

                                    ; TELL THE 8259 TO START LATCHING THIS INTERRUPT AGAIN

0009  B0 20                         MOV       AL,EOI
000B  E6 20                         OUT       THE_8259,AL

                                    ; GET LANA STATUS

000D  BA 0360                       MOV       DX,LANA_0_SR             ; GET THIS LANA'S STATUS PORT ADDR
0010  EB 00                         JMP       $+2
0012  EC                            IN        AL,DX

0013  50                            PUSH      AX
0014  0C 80                         OR        AL,80H                   ; ACKNOWLEDGE INTERRUPT TO LANA
0016  EB 00                         JMP       $+2
0018  EE                            OUT       DX,AL
0019  58                            POP       AX

                                    ; IF LANA IS GETTING AN NCB THEN

001A  F6 06 00A2 08                 TEST      BYTE PTR DS:[LANA_0_STATUS],LANA_GETTING_NCB
001F  74 11                         JZ        HC_SET?

                                    ;    GOT NCB. SET COMPLETE CODE. EXIT

0021  80 26 00A2 F7                 AND       BYTE PTR DS:[LANA_0_STATUS],ALL_BITS-LANA_GETTING_NCB
0026  A8 06                         TEST      AL,CPLT_CODE
0028  74 7F                         JZ        LANA_EXIT
002A  80 0E 00A2 04                 OR        BYTE PTR DS:[LANA_0_STATUS],LANA_HARD_ERR
002F  EB 78 90                      JMP       LANA_EXIT

                                    ; IGNORE INTERRUPTS WHILE PC IS IN CONTROL
                                    ;   (PC IS POLLING SR FOR GO=0)

0032  A8 80           HC_SET?:      TEST      AL,HC                    ; IF PC OWNS INTERFACE THEN
0034  75 73                         JNZ       LANA_EXIT                ; IGNORE THIS INTERRUPT

                                    ; IGNORE UN-ENABLED INTERRUPTS

0036  8A E0                         MOV       AH,AL
0038  BA 0363                       MOV       DX,LANA_0_HIR
003B  EC                            IN        AL,DX
003C  24 41                         AND       AL,TCI+GI
003E  22 C4                         AND       AL,AH
0040  74 67                         JZ        LANA_EXIT

                                    ; INSURE THIS IS A "REAL" INTERRUPT

0042  F6 C4 01                      TEST      AH,GO                    ; LANA MUST BE ASKING PC TO DO A CMD
0045  74 3B                         JZ        INT_BAD
                                    ; MAYBE.  DMA COMPLETE INTERRUPT?

0047  F6 C4 40                      TEST      AH,TC                    ; DMA TERMINAL COUNT REACHED?
004A  74 16                         JZ        LANA_REQUEST

                                    ; YES.  RELEASE DMA

004C  80 26 00A2 DF                 AND       BYTE PTR DS:[LANA_0_STATUS],ALL_BITS-LANA_DMAING  ; LANA ISN'T DMAING ANYMORE
0051  80 26 00A1 7F                 AND       BYTE PTR DS:[LANA_HARDWARE],ALL_BITS-DMA_3_BUSY   ; SO DMA IS NO LONGER BUSY

                                    ; DISABLE DMA COMPLETE INTERRUPTS (IN CASE SOMEONE ELSE USES DMA_3)

0056  BA 0363                       MOV       DX,LANA_0_HIR            ; GET CURRENT PC INTERFACE REGISTER
0059  EC                            IN        AL,DX
005A  24 8F                         AND       AL,ALL_BITS-TCI-IO_METHOD ; TURN OFF "TERMINAL COUNT" INTERRUPT ENABLE
005C  EB 00                         JMP       $+2
005E  EE                            OUT       DX,AL
005F  EB 3F 90                      JMP       CLEAR_GO

                                    ; NOT DMA COMPLETE.  GET THE CODE FOR THE CMD THAT LANA REQUESTS

0062                  LANA_REQUEST:

0062  BA 0361                       MOV       DX,LANA_0_PR             ; POINT TO LANA 0'S PARAMETER REGISTER
0065  EC                            IN        AL,DX                    ; GET THE "CMD CODE"

                                    ; IS IT A "REQUEST DATA TO/FROM LANA" CMD?

0066  3C 44                         CMP       AL,DATA_TO_LANA
0068  74 2C                         JE        DATA_REQ
006A  3C 42                         CMP       AL,DATA_FROM_LANA
006C  74 28                         JE        DATA_REQ

                                    ; NOPE.  MAYBE A "NCB COMPLETE" THEN?

006E  3C 43                         CMP       AL,NCB_CPLT
0070  74 1E                         JE        CMD_CPLT

                                    ; ERROR REPORT FROM LANA

0072  3C 45                         CMP       AL,ERROR_FROM_LANA
0074  75 0C                         JNZ       INT_BAD
0076  EC                            IN        AL,DX
0077  52                            PUSH      DX
0078  BA 0362                       MOV       DX,LANA_0_DR             ;SAVE FATAL ERROR REPORT
007B  EE                            OUT       DX,AL                    ;IN DATA REGISTER FOR LATER VIEWING
007C  5A                            POP       DX
007D  80 0E 00A2 02                 OR        BYTE PTR DS:[LANA_0_STATUS],LANA_HARD_ERR1

                                    ; INVALID INTERRUPT.  CATASTROPHIC ERROR!

0082  E8 0249 R       INT_BAD:      CALL      CATASTROPHIC_ERROR

                                    ; DETERMINE WHETHER TO CLEAR GO

0085  BA 0360                       MOV       DX,LANA_0_SR
0088  EC                            IN        AL,DX                    ; GET THIS LANA'S STATUS PORT VALUE
0089  A8 80                         TEST      AL,HC                    ; DOES LANA OWN INTERFACE?
008B  74 13                         JZ        CLEAR_GO                 ; IF YES, CLEAR_GO
008D  EB 1A 90                      JMP       LANA_EXIT

                                    ; HANDLE "NCB COMPLETE" CODE
```

```
0090  E8 00AD R              CMD_CPLT:  CALL     COMMAND_CPLT
0093  EB 14 90                          JMP      LANA_EXIT               ; CLEARS GO INTERNALLY

                                        ; PROCESS "REQUEST DATA TO/FROM LANA" CMD

0096  E8 0167 R              DATA_REQ:  CALL     LANA_DATA_REQ

                                        ; DON'T CLEAR GO IF USING DMA

0099  F6 06 00A2 20                     TEST     BYTE PTR DS:[LANA_0_STATUS],LANA_DMAING
009E  75 09                             JNZ      LANA_EXIT

                                        ; TELL LANA REQUEST IS COMPLETE

00A0  BA 0360                CLEAR_GO:  MOV      DX,LANA_0_SR
00A3  EC                                IN       AL,DX                   ; GET CURRENT SR PORT VALUE
00A4  24 FE                             AND      AL,ALL_BITS-GO          ; TURN OFF THE GO BIT
00A6  EB 00                             JMP      $+2
00A8  EE                                OUT      DX,AL

                                        ; INTERRUPT RETURN

00A9                         LANA_EXIT: RESTORE  <DS,DX,AX>
00A9  1F                  +                POP      DS
00AA  5A                  +                POP      DX
00AB  58                  +                POP      AX

00AC  CF                                   IRET

00AD                         LANA_0_HNDLR ENDP

                             ;-----------------------------------------------------------------------
                             ; COMMAND_CPLT
                             ;
                             ;    PROGRAM I/O'S THE NCB (OF THE COMPLETED CMD) OVER FROM THE LANA.
                             ; IF THE NCB IS NO-WAIT TYPE, THE POST ROUTINE IS INVOKED.
                             ;
                             ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:
                             ;
                             ;       REGISTER   |  ACCESS   |                USAGE
                             ;      ------------+-----------+----------------------------------------
                             ;       DS         |  CONST    | LO MEMEORY SEGMENT
                             ;          DX      |  DESTROY  | LANA'S PR PORT ADDR
                             ;          AX      |  DESTROY  | INTERNAL
                             ;
                             ; INTERRUPTS SHOULD BE MASKED ON ENTRY.
                             ;
                             ; NOTE: CLEARS GO INTERNALLY.
                             ;
                             ;-----------------------------------------------------------------------

00AD                         COMMAND_CPLT PROC      NEAR

                                        SAVE     <ES,BX,DI,CX,SI>
00AD  06                  +                PUSH     ES
00AE  53                  +                PUSH     BX
00AF  57                  +                PUSH     DI
00B0  51                  +                PUSH     CX
00B1  56                  +                PUSH     SI
                                        ; LOCK INTERFACE & ENABLE INTERRUPTS

00B2  80 0E 00A2 40                     OR       BYTE PTR DS:[LANA_0_STATUS],LANA_LOCKED
00B7  FB                                STI

                                        ; GET NCB'S DESTINATION ADDR & LENGTH

00B8  E8 0260 R                         CALL     ADDR_AND_LEN
00BB  26: C4 7D 34                      LES      DI,ES:[DI].NCB_RESERVE_NCB@  ; GET REAL NCB ES:BX
                                        SAVE     <DI>                         ; SAVE BASE OF NCB FOR LATER
00BF  57                  +                PUSH     DI

                                        ; SETUP INTERFACE FOR PROGRAMMED I/O FROM LANA

00C0  BA 0363                           MOV      DX,LANA_0_HIR
00C3  EC                                IN       AL,DX                        ; GET CURRENT INTERFACE SETUP
00C4  24 C7                             AND      AL,ALL_BITS-IO_METHOD-DD_BIT ; MASK OFF BITS OF INTEREST
00C6  0C 00                             OR       AL,PROGRAMMED_IO+LANA_TO_PC  ; AND SET THEM AS WE WANT IT
00C8  EB 00                             JMP      $+2
00CA  EE                                OUT      DX,AL

                                        ; SETUP TO READ IN THE NCB FROM THE LANA

00CB  BA 0362                           MOV      DX,LANA_0_DR
00CE  BE 0360                           MOV      SI,LANA_0_SR
00D1  B4 20                             MOV      AH,DRF

                                        ; WAIT FOR (THE NEXT) BYTE OF NCB

00D3                         NEXT_NCB_BYTE:
00D3  87 F2                             XCHG     SI,DX
00D5  EC                                IN       AL,DX
00D6  84 C4                             TEST     AL,AH                   ; IS IT READY?
00D8  74 09                             JZ       NCB_NOT_READY           ; IF NOT, JMP TO WAITER

                                        ; GET AND STORE BYTE

00DA                         GET_NCB_BYTE:
00DA  87 D6                             XCHG     DX,SI
00DC  EC                                IN       AL,DX
00DD  AA                                STOSB

                                        ; LOOP IF MORE NCB TO READ IN

00DE  E2 F3                             LOOP     NEXT_NCB_BYTE
00E0  EB 18 90                          JMP      NCB_XFERD

                                        ; BYTE NOT READY.  WAIT UP TO "AWHILE" FOR IT

00E3                         NCB_NOT_READY:
00E3  BB 0000                           MOV      BX,AWHILE

00E6                         NCB_NOT_READY2:
00E6  EC                                IN       AL,DX
00E7  84 C4                             TEST     AL,AH
00E9  75 EF                             JNZ      GET_NCB_BYTE
00EB  4B                                DEC      BX
00EC  75 F8                             JNE      NCB_NOT_READY2

                                        ; TIMEOUT WHILE WAITING.  CATASTROPHIC ERROR!

00EE  E8 0249 R                         CALL     CATASTROPHIC_ERROR
```

```
                                           ; SET NCB RETURN CODE TO NCBSYS_ERR?

                                           RESTORE    <BX>                        ; SAVED FROM DI
00F1  5B                    +                  POP        BX
00F2  26: C6 47 01 40                      MOV        ES:BYTE PTR [BX].NCB_RETCODE,NCBSYS_ERR?
00F7  EB 02 90                             JMP        CLEANUP

                                           ; NCB TRANSFERRED.  GET NCB BASE ADDRESS

00FA                              NCB_XFERD: RESTORE  <BX>                        ; SAVED FROM DI
00FA  5B                    +                  POP        BX

                                           ; MASK INTERRUPTS, UNLOCK INTERFACE, TELL LANA WE'RE DONE

00FB  FA                          CLEANUP: CLI
00FC  80 26 00A2 BF                        AND        BYTE PTR DS:[LANA_0_STATUS],ALL_BITS-LANA_LOCKED
0101  BA 0360                              MOV        DX,LANA_0_SR
0104  EC                                   IN         AL,DX
0105  24 FE                                AND        AL,ALL_BITS-GO
0107  EB 00                                JMP        $+2
0109  EE                                   OUT        DX,AL

                                           ; RESTORE BUFFER@ TO SEG:OFF TYPE ADDR

010A  26: 8B 47 38                         MOV        AX,ES:WORD PTR [BX].NCB_RESERVE_BUFFER@
010E  26: 89 47 04                         MOV        ES:WORD PTR [BX].NCB_BUFFER@,AX
0112  26: 8B 47 3A                         MOV        AX,ES:WORD PTR [BX].NCB_RESERVE_BUFFER@+2
0116  26: 89 47 06                         MOV        ES:WORD PTR [BX].NCB_BUFFER@+2,AX

                                           ; IF NCB FOR MULTIPLE SEND RESTORE SECOND BUFFER@ TO SEG:OFF TYPE ADDR

011A  26: 8A 07                            MOV        AL,ES:[BX].NCB_COMMAND
011D  24 7F                                AND        AL,7FH
011F  3C 17                                CMP        AL,NCBSENDMULTIPLE
0121  75 10                                JNZ        UDT_RCD

0123  26: 8B 47 3C                         MOV        AX,ES:WORD PTR [BX].NCB_RESERVE_BUFFER2@
0127  26: 89 47 0C                         MOV        ES:WORD PTR [BX].NCB_BUFFER@+8,AX
012B  26: 8B 47 3E                         MOV        AX,ES:WORD PTR [BX].NCB_RESERVE_BUFFER2@+2
012F  26: 89 47 0E                         MOV        ES:WORD PTR [BX].NCB_BUFFER@+0AH,AX

                                           ;UPDATE NCB_CMD_CPLT

0133  26: 8A 47 01                UDT_RCD: MOV        AL,ES:[BX].NCB_RETCODE
0137  26: 88 47 31                         MOV        ES:BYTE PTR [BX].NCB_CMD_CPLT,AL


                                           ; DO WE POST THIS NCB?


013B  26: F6 07 80                         TEST       ES:BYTE PTR [BX].NCB_COMMAND,NCBNO_WAIT
013F  75 09                                JNZ        NO_WAIT?
0141  B8 9180                              MOV        AX,DEV_BUSY_POST
0144  CD 15                                INT        15H
0146  FA                                   CLI
0147  EB 18 90                             JMP        CMD_EXIT

                                           ;CHECK FOR POST ADDRESS BEIGN ZERO (NO_WAIT,NO_POST)

014A  26: 83 7F 2C 00             NO_WAIT?: CMP       ES:WORD PTR [BX].NCB_POST@,0000H
014F  75 07                                JNZ        POST
0151  26: 83 7F 2E 00                      CMP        ES:WORD PTR [BX].NCB_POST@+2,0000H
0156  74 09                                JE         CMD_EXIT

                                           ; YES.  GET RETCODE & POST (AS AN INTERRUPT)


0158  26: 8A 47 01                POST:    MOV        AL,ES:[BX].NCB_RETCODE
015C  9C                                   PUSHF
015D  26: FF 5F 2C                         CALL       ES:DWORD PTR [BX].NCB_POST@

                                           ; RETURN

0161                              CMD_EXIT: RESTORE   <SI,CX,DI,BX,ES>
0161  5E                    +                  POP        SI
0162  59                    +                  POP        CX
0163  5F                    +                  POP        DI
0164  5B                    +                  POP        BX
0165  07                    +                  POP        ES
0166  C3                                   RET

0167                              COMMAND_CPLT ENDP


                                  ;-----------------------------------------------------------------------------
                                  ; LANA_DATA_REQ
                                  ;
                                  ;    SETS UP & STARTS A TRANFER OF DATA BETWEEN THE PC AND THE LANA.
                                  ; WHILE DMA IS BUSY, THE PC WILL XFER "CHUNKS" VIA PROGRAMMED I/O.
                                  ; WHEN DMA IS FREE, A DMA XFER OF THE REMAINDER WILL BE SETUP AND STARTED.
                                  ;
                                  ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:
                                  ;
                                  ;       REGISTER   |   ACCESS   |              USAGE
                                  ;      ------------+------------+-----------------------------------------
                                  ;       DS         |   CONST    | LO MEMEORY SEGMENT
                                  ;          DX      |  DESTROY   | LANA'S PR PORT ADDR
                                  ;          AL      |  DESTROY   | CMD CODE OF LANA REQUEST
                                  ;          AH      |  DESTROY   | INTERNAL
                                  ;
                                  ; INTERRUPTS SHOULD BE MASKED ON ENTRY.
                                  ;
                                  ;-----------------------------------------------------------------------------

0167                              LANA_DATA_REQ PROC   NEAR

                                           SAVE        <CX,ES,DI>
0167  51                    +                  PUSH       CX
0168  06                    +                  PUSH       ES
0169  57                    +                  PUSH       DI

                                           ; SAVE CMD CODE FOR LATER

016A  8A E0                                MOV         AH,AL

                                           ; LOCK INTERFACE

016C  80 0E 00A2 40                        OR          BYTE PTR DS:[LANA_0_STATUS],LANA_LOCKED

                                           ; GET DATA'S PC ADDR & LEN

0171  E8 0260 R                            CALL        ADDR_AND_LEN
```

```
                                    ; SETUP FOR PROGRAMMED I/O (MAY LATER CHANGE TO DMA) IN THE CORRECT DIRECTION

0174  BA 0363                       MOV       DX,LANA_0_HIR
0177  EC                            IN        AL,DX                           ; GET CURRENT INTERFACE SETUP
0178  24 C7                         AND       AL,ALL_BITS-IO_METHOD-DD_BIT    ; MASK OFF AREA OF INTEREST
017A  80 FC 44                      CMP       AH,DATA_TO_LANA                 ; IS IT TO OR FROM THE LANA?
017D  75 05                         JNE       FROM_LANA
017F  0C 08                         OR        AL,PROGRAMMED_IO+PC_TO_LANA     ; TO LANA
0181  EB 03 90                      JMP       DIR_SET
0184  0C 00              FROM_LANA: OR        AL,PROGRAMMED_IO+LANA_TO_PC     ; FROM LANA
0186  EE                 DIR_SET:   OUT       DX,AL

                                    ; DMA_3 BUSY?

0187  F6 06 00A1 80      TRY_DMA_3: TEST      BYTE PTR DS:[LANA_HARDWARE],DMA_3_BUSY
018C  74 18                         JZ        GET_DMA_3

                                    ; YEP.  PROGRAM I/O A "CHUNK"

018E  FB                 PROGRAM_IO:STI                                  ; ALLOW INTERRUPTS & HOPE THAT DMA FREES UP
018F  E8 01B9 R                     CALL      PROGRAM_IO_CHUNK
0192  FA                            CLI

                                    ; CATASTROPHIC ERROR HAPPEN?

0193  F6 06 00A2 04                 TEST      BYTE PTR DS:[LANA_0_STATUS],LANA_HARD_ERR
0198  75 16                         JNZ       DATA_EXIT                  ; IF SO, EXIT

                                    ; NOPE.  MORE DATA LEFT TO XFER?

019A  83 F9 00                      CMP       CX,0
019D  74 11                         JZ        DATA_EXIT                  ; IF NOT, EXIT

                                    ; YES.  WORTH TRYING FOR DMA?

019F  83 F9 70                      CMP       CX,CHUNK_SIZE
01A2  73 E3                         JNB       TRY_DMA_3
01A4  EB E8                         JMP       PROGRAM_IO

                                    ; SETUP & START DMA (OF REMAINING DMA)

01A6  BA 0363            GET_DMA_3: MOV       DX,LANA_0_HIR
01A9  50                            PUSH      AX
01AA  B0 01                         MOV       AL,01                      ; INDICATES LANA REQUESTING DMA
01AC  E8 0000 E                     CALL      DMA_START_UP
01AF  58                            POP       AX

                                    ; UNLOCK INTERFACE AND RETURN

01B0  80 26 00A2 BF      DATA_EXIT: AND       BYTE PTR DS:[LANA_0_STATUS],ALL_BITS-LANA_LOCKED
                                    RESTORE   <DI,ES,CX>
01B5  5F                 +              POP       DI
01B6  07                 +              POP       ES
01B7  59                 +              POP       CX
01B8  C3                            RET

01B9                     LANA_DATA_REQ ENDP


                         ;_______________________________________________________________________________
                         ;                                                                               |
                         ; PROGRAM_IO_CHUNK                                                              |
                         ;                                                                               |
                         ;     PROGRAM I/O'S A "CHUNK" OF THE REMAINING DATA BETWEEN THE PC & LANA.      |
                         ;                                                                               |
                         ;     THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                              |
                         ;                                                                               |
                         ;         REGISTER   |   ACCESS  |                  USAGE                       |
                         ;      --------------+-----------+-----------------------------------------     |
                         ;         DS         |   CONST   | LO MEMEORY SEGMENT                           |
                         ;           AH       |   CONST   | CMD CODE OF LANA REQ                         |
                         ;         ES:DI      |    VAR    | START OF "CHUNK" IN MEMORY                   |
                         ;           CX       |    VAR    | LENGTH OF REMAINING DATA                     |
                         ;           DX       |  DESTROY  | INTERNAL                                     |
                         ;                                                                               |
                         ; INTERRUPTS SHOULD BE UNMASKED ON ENTRY.                                       |
                         ;                                                                               |
                         ;_______________________________________________________________________________|

01B9                     PROGRAM_IO_CHUNK PROC NEAR

                                    SAVE      <SI,AX,BX>
01B9  56                 +              PUSH      SI
01BA  50                 +              PUSH      AX
01BB  53                 +              PUSH      BX

                                    ; SETUP CHUNK SIZE, DATA PORT ADDR, UPDATE "REMAINING DATA" SIZE

01BC  BA 0070                       MOV       DX,CHUNK_SIZE              ; THIS_CHUNK := MIN(CHUNK_SIZE,REMAINING_DATA)
01BF  3B CA                         CMP       CX,DX
01C1  73 02                         JNB       GOT_CHUNK
01C3  8B D1                         MOV       DX,CX
01C5  2B CA              GOT_CHUNK: SUB       CX,DX                      ; REMAINING_DATA := REMAINING_DATA - THIS_CHUNK
                                    SAVE      <CX>
01C7  51                 +              PUSH      CX
01C8  8B CA                         MOV       CX,DX
01CA  BA 0362                       MOV       DX,LANA_0_DR

                                    ; DATA TO LANA?

01CD  80 FC 44                      CMP       AH,DATA_TO_LANA
01D0  75 4F                         JNE       FROM_LANA2

                                    ; YEP.  SETUP SOURCE PTR & PORT ADDRS

01D2  8B F7                         MOV       SI,DI
                                    SWAP      ES,DS
01D4  06                 +              PUSH      ES
01D5  1E                 +              PUSH      DS
01D6  07                 +              POP       ES
01D7  1F                 +              POP       DS
01D8  BF 0360                       MOV       DI,LANA_0_SR
01DB  B4 10                         MOV       AH,DRE
01DD  EB 0A 90                      JMP       OUT_TO_LANA2

                                    ; ROOM IN DATA REGISTER?

01E0  87 FA              DATA_ROOM?:XCHG      DI,DX
01E2  EC                            IN        AL,DX
01E3  84 C4                         TEST      AL,AH                      ; ROOM IN DATA REGISTER?
01E5  74 09                         JZ        NO_ROOM

                                    ; STUFF BYTE INTO DATA REGISTER

01E7                     OUT_TO_LANA:
01E7  87 D7                         XCHG      DX,DI
```

```
01E9                            OUT_TO_LANA2:
01E9  AC                                LODSB
01EA  EE                                OUT     DX,AL

                                        ; LOOP IF MORE "CHUNK" TO SEND

01EB  E2 F3                             LOOP    DATA_ROOM?
01ED  EB 15 90                          JMP     LANA_GOT_ALL?

                                        ; WAIT UP TO "AWHILE" FOR DRE

01F0  BB 0000                   NO_ROOM:  MOV   BX,AWHILE
01F3  EC                        ROOM_NOW?: IN   AL,DX
01F4  84 C4                             TEST    AL,AH
01F6  75 EF                             JNZ     OUT_TO_LANA
01F8  4B                                DEC     BX                      ; TIMEOUT EXPIRED?
01F9  75 F8                             JNE     ROOM_NOW?               ; IF NOT, TRY AGAIN
                                        SWAP    ES,DS                   ; RESTORE REGISTERS TO "ON ENTRY" SETUP
01FB  06                       +           PUSH    ES
01FC  1E                       +           PUSH    DS
01FD  07                       +           POP     ES
01FE  1F                       +           POP     DS
01FF  8B FE                             MOV     DI,SI
0201  EB 3E 90                          JMP     IO_TIMEOUT

                                        ; HAS LANA GOT ALL OF THE "CHUNK" YET?

0204                            LANA_GOT_ALL?:
                                        SWAP    ES,DS                   ; RESTORE REGISTERS TO "ON ENTRY" SETUP
0204  06                       +           PUSH    ES
0205  1E                       +           PUSH    DS
0206  07                       +           POP     ES
0207  1F                       +           POP     DS
0208  8B D7                             MOV     DX,DI
020A  8B FE                             MOV     DI,SI
020C  B4 20                             MOV     AH,DRF
020E  EC                                IN      AL,DX
020F  84 C4                             TEST    AL,AH                   ; DATA REGISTER TOTALLY EMPTY?
0211  74 31                             JZ      CHUNK_DONE              ; IF SO, CHUNK IS DONE

                                        ; NO.  WAIT UP TO "AWHILE" FOR LANA TO GET ALL OF IT

0213  BB 0000                           MOV     BX,AWHILE
0216                            GOT_ALL_NOW?:
0216  EC                                IN      AL,DX
0217  84 C4                             TEST    AL,AH                   ; DATA REGISTER TOTALLY EMPTY?
0219  74 29                             JZ      CHUNK_DONE              ; IF SO, CHUNK IS DONE
021B  4B                                DEC     BX                      ; TIMEOUT EXPIRED?
021C  75 F8                             JNE     GOT_ALL_NOW?            ; IF NOT, TRY AGAIN
021E  EB 21 90                          JMP     IO_TIMEOUT

                                        ; SETUP FOR I/O FROM LANA

0221  BE 0360                   FROM_LANA2:MOV  SI,LANA_0_SR
0224  B4 20                             MOV     AH,DRF

                                        ; DATA BYTE WAITING?

0226  87 D6                     ANY_DATA?: XCHG DX,SI
0228  EC                                IN      AL,DX
0229  84 C4                             TEST    AL,AH
022B  74 09                             JZ      NO_DATA
                                        ; YES.  GET IT, STORE IT, LOOP IF MORE DATA REMAINS

022D                            IN_FROM_LANA:
022D  87 F2                             XCHG    SI,DX
022F  EC                                IN      AL,DX
0230  AA                                STOSB
0231  E2 F3                             LOOP    ANY_DATA?
0233  EB 0F 90                          JMP     CHUNK_DONE

                                        ; WAIT UP TO "AWHILE" FOR DATA BYTE FROM LANA

0236  BB 0000                   NO_DATA:  MOV   BX,AWHILE
0239  EC                        DATA_NOW?: IN   AL,DX
023A  84 C4                             TEST    AL,AH                   ; DATA BYTE WAITING NOW?
023C  75 EF                             JNZ     IN_FROM_LANA            ; IF SO, GO GET IT
023E  4B                                DEC     BX                      ; TIMEOUT EXPIRED?
023F  75 F8                             JNE     DATA_NOW?               ; IF NOT, TRY AGAIN

                                        ; TIMEOUT WAITING ON I/O.  CATASTROPHIC ERROR

0241  E8 0249 R                 IO_TIMEOUT:CALL    CATASTROPHIC_ERROR

                                        ; CHUNK DONE.  RETURN

0244                            CHUNK_DONE:RESTORE  <CX>
0244  59                       +           POP     CX
                                        RESTORE <BX,AX,SI>
0245  5B                       +           POP     BX
0246  58                       +           POP     AX
0247  5E                       +           POP     SI
0248  C3                                RET

0249                            PROGRAM_IO_CHUNK ENDP


                                ;_____________________________________________________________________
                                ;                                                                     |
                                ; CATASTROPHIC_ERROR                                                  |
                                ;                                                                     |
                                ;    HANDLES CATASTROPHIC INTERFACE ERRORS.                           |
                                ;                                                                     |
                                ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                     |
                                ;                                                                     |
                                ;       REGISTER   |  ACCESS  |             USAGE                     |
                                ;       -----------+----------+-------------------------------------  |
                                ;       DS         |  CONST   |  LO MEMEORY SEGMENT                   |
                                ;          DX      |  DESTROY |  INTERNAL                             |
                                ;          AX      |  DESTROY |  INTERNAL                             |
                                ;                                                                     |
                                ;_____________________________________________________________________|

0249                            CATASTROPHIC_ERROR PROC NEAR

                                        ; IF LANA OWNS INTERFACE (WITH GO SET), SET CPLT_CODE TO "CANT_CPLT?"

0249  BA 0360                           MOV     DX,LANA_0_SR
024C  EC                                IN      AL,DX
024D  A8 80                             TEST    AL,HC                   ; DOES LANA OWN INTERFACE?
024F  75 09                             JNZ     SET_SPCL                ; IF NOT, DON'T SET CPLT_CODE
0251  A8 01                             TEST    AL,GO                   ; IS LANA WAITING ON THE PC?
0253  74 05                             JZ      SET_SPCL                ; IF NOT, DON'T SET CPLT_CODE
0255  24 F9                             AND     AL,ALL_BITS-CPLT_CODE   ; ELSE, MASK OFF AREA OF INTEREST
```

```
0257  0C 04                           OR      AL,CANT_CPLT?             ; AND SET AS DESIRED
0259  EE                              OUT     DX,AL

                                      ; REPORT A HARDWARE ERROR

025A                          SET_SPCL:

025A  80 0E 00A2 04                   OR      BYTE PTR DS:[LANA_0_STATUS],LANA_HARD_ERR

                                      ; RETURN

025F  C3                              RET

0260                          CATASTROPHIC_ERROR ENDP


                              ;------------------------------------------------------------------
                              ;
                              ; ADDR_AND_LEN
                              ;
                              ;    GETS A 32-BIT ADDRESS AND 16-BIT LENGTH FROM LANA_0'S PR PORT.
                              ; THE 32-BIT ADDRESS IS CONVERTED INTO A SEGMENT:OFFSET PAIR.
                              ;
                              ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:
                              ;
                              ;       REGISTER  |   ACCESS  |              USAGE
                              ;       ----------+-----------+-----------------------------------
                              ;          DX     |   CONST   | LANA'S PR PORT ADDR
                              ;        ES:DI    |  RESULT   | ADDRESS IN SEGMENT:OFFSET FORM
                              ;          CX     |  RESULT   | LENGTH
                              ;
                              ;------------------------------------------------------------------

0260                          ADDR_AND_LEN PROC       NEAR

                                      SAVE    <AX>
0260  50                   +             PUSH    AX

                                      ; GET 32-BIT ADDRESS

0261  EC                              IN      AL,DX
0262  8A C8                           MOV     CL,AL
0264  81 E1 000F                      AND     CX,000FH
0268  8B F9                           MOV     DI,CX                     ; DI GET LOWEST 4 BITS OF ADDR
026A  8A C8                           MOV     CL,AL
026C  EC                              IN      AL,DX
026D  8A E8                           MOV     CH,AL
026F  F8                              CLC
0270  D1 E9                           SHR     CX,1
0272  D1 E9                           SHR     CX,1
0274  D1 E9                           SHR     CX,1
0276  D1 E9                           SHR     CX,1                      ; CX GET NEXT 12 BITS (TOP 4 BITS=0)
0278  EC                              IN      AL,DX
0279  8A E0                           MOV     AH,AL
027B  EC                              IN      AL,DX
027C  D1 E0                           SHL     AX,1
027E  D1 E0                           SHL     AX,1
0280  D1 E0                           SHL     AX,1
0282  D1 E0                           SHL     AX,1
0284  25 F000                         AND     AX,0F000H
0287  03 C1                           ADD     AX,CX                     ; AX BECOMES SEGMENT TYPE ADDR
0289  8E C0                           MOV     ES,AX
                                      ; GET 16-BIT LENGTH AND RETURN

028B  EC                              IN      AL,DX
028C  8A C8                           MOV     CL,AL
028E  EC                              IN      AL,DX
028F  8A E8                           MOV     CH,AL


                                      RESTORE <AX>
0291  58                   +             POP     AX
0292  C3                              RET

0293                          ADDR_AND_LEN ENDP

0293                          NETWORK    ENDS
                                      END
                              TITLE LANA_1 INTERRUPT HANDLER
                                      ;
                                      ; LANA_1.ASM
                                      ;
                                      ;    LANA 1'S INTERRUPT HANDLER.


                              ;------------------------------------------------------------------
                              ;
                              ;       MODULE : LANA_1_HNDLR
                              ;    COMPONENT : NETWORK BIOS
                              ;
                              ;    HANDLES THE LANA_1 (IRQ2 OR IRQ3) INTERRUPT.  VALID REASONS FOR A LANA
                              ; INTERRUPT ARE:
                              ;
                              ;       DMA COMPLETE
                              ;       COMMAND COMPLETE    --> COULD CAUSE A NETBIOS POST
                              ;       LANA REQUEST FOR DATA FROM PC
                              ;       LANA REQUEST TO XFER DATA TO PC
                              ;
                              ;
                              ;    ALL REGISTERS AND FLAGS ARE PRESERVED.
                              ;
                              ;      NETBIOS.LIB CONTAINS THE NETBIOS INTERFACE EQUATES AND STRUCTURES
                              ;      LANAS.INC CONTAINS THE LANA INTERFACE EQUATES AND STRUCTURES
                              ;
                              ;------------------------------------------------------------------

0000                          NETWORK    SEGMENT    PARA PUBLIC     'CODE'

                                      ASSUME  CS:NETWORK
                                      ASSUME  DS:NOTHING
                                      ASSUME  SS:NOTHING
                                      ASSUME  ES:NOTHING

                                      EXTRN   DMA_START_UP    : NEAR    ; STARTS UP A DMA XFER WITH A LANA

                                      PUBLIC  LANA_1_HNDLR    ; NEAR    ; LANA 1'S INTERRUPT HANDLER

                              .LIST

= 00FF                        ALL_BITS        EQU     0FFH              ; AID TO MASKING OFF BITS

= 0020                        THE_8259        EQU     20H               ; 8259 PORT TO SEND EOI CMD TO
= 0020                        EOI             EQU     20H               ; CMD CODE FOR EOI
```

```
= 0000                  AWHILE          EQU     0000H                   ; AMOUNT OF TIME TO WAIT ON INTERFACE
= 0070                  CHUNK_SIZE      EQU     0070H                   ; MAX. SIZE OF A PROGRAM I/O'D "CHUNK"

= 9180                  DEV_BUSY_POST   EQU     9180H


                        ;----------------------------------------------------------------------
                        ; LANA_1_HNDLR
                        ;
                        ;    HANDLES THE LANA_1 (IRQ2 OR IRQ3) INTERRUPT.  VALID REASONS FOR A LANA
                        ; INTERRUPT ARE:
                        ;
                        ;        DMA COMPLETE
                        ;        COMMAND COMPLETE    --> COULD CAUSE A NETBIOS POST
                        ;        LANA REQUEST FOR DATA FROM PC
                        ;        LANA REQUEST TO XFER DATA TO PC
                        ;
                        ;
                        ;    ALL REGISTERS AND FLAGS ARE PRESERVED.
                        ;
                        ;----------------------------------------------------------------------

0000                    LANA_1_HNDLR PROC       NEAR

                                SAVE    <AX,DX,DS>
0000  50                +           PUSH    AX
0001  52                +           PUSH    DX
0002  1E                +           PUSH    DS

                                ; SET UP GLOBAL ASSUMPTIONS

0003  FC                        CLD                             ; ALL STRINGS GO UP
0004  B8 0040                   MOV     AX,LO_MEM_SEG           ; *** CHANGE TO LO_MEM_SEG LATER ***
0007  8E D8                     MOV     DS,AX                   ; DS PTS TO LANA RESERVED MEMORY

                                ; TELL THE 8259 TO START LATCHING THIS INTERRUPT AGAIN

0009  B0 20                     MOV     AL,EOI
000B  E6 20                     OUT     THE_8259,AL

                                ; GET LANA STATUS (JUST ONCE + ONCE MORE IF CLR GO)

000D  BA 0368                   MOV     DX,LANA_1_SR            ; GET THIS LANA'S STATUS PORT ADDR
0010  EB 00                     JMP     $+2
0012  EC                        IN      AL,DX


0013  50                        PUSH    AX
0014  0C 80                     OR      AL,80H                  ; ACKNOWLEDGE INTERRUPT TO LANA
0016  EB 00                     JMP     $+2
0018  EE                        OUT     DX,AL
0019  58                        POP     AX

                                ; IF LANA IS GETTING AN NCB THEN

001A  F6 06 00A3 08             TEST    BYTE PTR DS:[LANA_1_STATUS],LANA_GETTING_NCB
001F  74 11                     JZ      HC_SET?

                                ;    GOT NCB. SET COMPLETE CODE. EXIT

0021  80 26 00A3 F7             AND     BYTE PTR DS:[LANA_1_STATUS],ALL_BITS-LANA_GETTING_NCB
0026  A8 06                     TEST    AL,CPLT_CODE
0028  74 7F                     JZ      LANA_EXIT
002A  80 0E 00A3 04             OR      BYTE PTR DS:[LANA_1_STATUS],LANA_HARD_ERR
002F  EB 78 90                  JMP     LANA_EXIT

                                ; IGNORE INTERRUPTS WHILE PC IS IN CONTROL
                                ;    (PC IS POLLING SR FOR GO=0)

0032  A8 80             HC_SET?: TEST   AL,HC                   ; IF PC OWNS INTERFACE THEN
0034  75 73                     JNZ     LANA_EXIT               ;    IGNORE THIS INTERRUPT

                                ; IGNORE UN-ENABLED INTERRUPTS

0036  8A E0                     MOV     AH,AL
0038  BA 036B                   MOV     DX,LANA_1_HIR
003B  EC                        IN      AL,DX
003C  24 41                     AND     AL,TCI+GI
003E  22 C4                     AND     AL,AH
0040  74 67                     JZ      LANA_EXIT

                                ; INSURE THIS IS A "REAL" INTERRUPT

0042  F6 C4 01                  TEST    AH,GO                   ; LANA MUST BE ASKING PC TO DO A CMD
0045  74 3B                     JZ      INT_BAD

                                ; MAYBE.  DMA COMPLETE INTERRUPT?

0047  F6 C4 40                  TEST    AH,TC                   ; DMA TERMINAL COUNT REACHED?
004A  74 16                     JZ      LANA_REQUEST

                                ; YES.  RELEASE DMA

004C  80 26 00A3 DF             AND     BYTE PTR DS:[LANA_1_STATUS],ALL_BITS-LANA_DMAING  ; LANA ISN'T DMAING ANYMORE
0051  80 26 00A1 7F             AND     BYTE PTR DS:[LANA_HARDWARE],ALL_BITS-DMA_3_BUSY   ; SO DMA IS NO LONGER BUSY

                                ; DISABLE DMA COMPLETE INTERRUPTS (IN CASE SOMEONE ELSE USES DMA_3)

0056  BA 036B                   MOV     DX,LANA_1_HIR           ; GET CURRENT PC INTERFACE REGISTER
0059  EC                        IN      AL,DX
005A  24 8F                     AND     AL,ALL_BITS-TCI-IO_METHOD  ; TURN OFF "TERMINAL COUNT" INTERRUPT ENABLE
005C  EB 00                     JMP     $+2
005E  EE                        OUT     DX,AL
005F  EB 3F 90                  JMP     CLEAR_GO

                                ; NOT DMA COMPLETE.  GET THE CODE FOR THE CMD THAT LANA REQUESTS

0062                    LANA_REQUEST:
0062  BA 0369                   MOV     DX,LANA_1_PR            ; POINT TO LANA 1'S PARAMETER REGISTER
0065  EC                        IN      AL,DX                   ; GET THE "CMD CODE"

                                ; IS IT A "REQUEST DATA TO/FROM LANA" CMD?

0066  3C 44                     CMP     AL,DATA_TO_LANA
0068  74 2C                     JE      DATA_REQ
006A  3C 42                     CMP     AL,DATA_FROM_LANA
006C  74 28                     JE      DATA_REQ

                                ; NOPE.  MAYBE A "NCB COMPLETE" THEN?

006E  3C 43                     CMP     AL,NCB_CPLT
0070  74 1E                     JE      CMD_CPLT

                                ; ERROR REPORT FROM LANA
```

```
0072  3C 45                        CMP      AL,ERROR_FROM_LANA
0074  75 0C                        JNZ      INT_BAD
0076  EC                           IN       AL,DX
0077  52                           PUSH     DX
0078  BA 036A                      MOV      DX,LANA_1_DR
007B  EE                           OUT      DX,AL
007C  5A                           POP      DX
007D  80 0E 00A3 02                OR       BYTE PTR DS:[LANA_1_STATUS],LANA_HARD_ERR1

                                   ; INVALID INTERRUPT.  CATASTROPHIC ERROR!

0082  E8 0249 R          INT_BAD:  CALL     CATASTROPHIC_ERROR

                                   ; DETERMINE WHETHER TO CLEAR GO

0085  BA 0368                      MOV      DX,LANA_1_SR
0088  EC                           IN       AL,DX                   ; GET THIS LANA'S STATUS PORT VALUE
0089  A8 80                        TEST     AL,HC                   ; DOES LANA OWN INTERFACE?
008B  74 13                        JZ       CLEAR_GO                ; IF YES, CLEAR_GO
008D  EB 1A 90                     JMP      LANA_EXIT

                                   ; HANDLE "NCB COMPLETE" CODE

0090  E8 00AD R          CMD_CPLT: CALL     COMMAND_CPLT
0093  EB 14 90                     JMP      LANA_EXIT               ; CLEARS GO INTERNALLY

                                   ; PROCESS "REQUEST DATA TO/FROM LANA" CMD

0096  E8 0167 R          DATA_REQ: CALL     LANA_DATA_REQ

                                   ; DON'T CLEAR GO IF USING DMA

0099  F6 06 00A3 20                TEST     BYTE PTR DS:[LANA_1_STATUS],LANA_DMAING
009E  75 09                        JNZ      LANA_EXIT

                                   ; TELL LANA REQUEST IS COMPLETE

00A0  BA 0368            CLEAR_GO: MOV      DX,LANA_1_SR
00A3  EC                           IN       AL,DX                   ; GET CURRENT SR PORT VALUE
00A4  24 FE                        AND      AL,ALL_BITS-GO          ; TURN OFF THE GO BIT
00A6  EB 00                        JMP      $+2
00A8  EE                           OUT      DX,AL

                                   ; INTERRUPT RETURN

00A9                     LANA_EXIT: RESTORE <DS,DX,AX>
00A9  1F               +              POP      DS
00AA  5A               +              POP      DX
00AB  58               +              POP      AX
00AC  CF                           IRET

00AD                     LANA_1_HNDLR ENDP

                         ;_____________________________________________________________________
                         ;                                                                     |
                         ; COMMAND_CPLT                                                        |
                         ;                                                                     |
                         ;    PROGRAM I/O'S THE NCB (OF THE COMPLETED CMD) OVER FROM THE LANA. |
                         ; IF THE NCB IS NO-WAIT TYPE, THE POST ROUTINE IS INVOKED.            |
                         ;                                                                     |
                         ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                     |
                         ;                                                                     |
                         ;       REGISTER   |   ACCESS   |              USAGE                  |
                         ;       -----------+------------+-------------------------------------|
                         ;       DS         |   CONST    | LO MEMEORY SEGMENT                  |
                         ;          DX      |  DESTROY   | LANA'S PR PORT ADDR                 |
                         ;          AX      |  DESTROY   | INTERNAL                            |
                         ;                                                                     |
                         ; INTERRUPTS SHOULD BE MASKED ON ENTRY.                               |
                         ;                                                                     |
                         ; NOTE: CLEARS GO INTERNALLY.                                         |
                         ;                                                                     |
                         ;_____________________________________________________________________|

00AD                     COMMAND_CPLT PROC  NEAR

                                   SAVE     <ES,BX,DI,CX,SI>
00AD  06               +              PUSH     ES
00AE  53               +              PUSH     BX
00AF  57               +              PUSH     DI
00B0  51               +              PUSH     CX
00B1  56               +              PUSH     SI

                                   ; LOCK INTERFACE & ENABLE INTERRUPTS

00B2  80 0E 00A3 40                OR       BYTE PTR DS:[LANA_1_STATUS],LANA_LOCKED
00B7  FB                           STI

                                   ; GET NCB'S DESTINATION ADDR & LENGTH

00B8  E8 0260 R                    CALL     ADDR_AND_LEN
00BB  26: C4 7D 34                 LES      DI,ES:[DI].NCB_RESERVE_NCB@   ; GET REAL NCB ES:BX
                                   SAVE     <DI>                          ; SAVE BASE OF NCB FOR LATER
00BF  57               +              PUSH     DI

                                   ; SETUP INTERFACE FOR PROGRAMMED I/O FROM LANA

00C0  BA 036B                      MOV      DX,LANA_1_HIR
00C3  EC                           IN       AL,DX                         ; GET CURRENT INTERFACE SETUP
00C4  24 C7                        AND      AL,ALL_BITS-IO_METHOD-DD_BIT  ; MASK OFF BITS OF INTEREST
00C6  0C 00                        OR       AL,PROGRAMMED_IO+LANA_TO_PC   ; AND SET THEM AS WE WANT IT
00C8  EB 00                        JMP      $+2
00CA  EE                           OUT      DX,AL

                                   ; SETUP TO READ IN THE NCB FROM THE LANA

00CB  BA 036A                      MOV      DX,LANA_1_DR
00CE  BE 0368                      MOV      SI,LANA_1_SR
00D1  B4 20                        MOV      AH,DRF

                                   ; WAIT FOR (THE NEXT) BYTE OF NCB

00D3                     NEXT_NCB_BYTE:
00D3  87 F2                        XCHG     SI,DX
00D5  EC                           IN       AL,DX
00D6  84 C4                        TEST     AL,AH                   ; IS IT READY?
00D8  74 09                        JZ       NCB_NOT_READY           ; IF NOT, JMP TO WAITER

                                   ; GET AND STORE BYTE

00DA                     GET_NCB_BYTE:
00DA  87 D6                        XCHG     DX,SI
00DC  EC                           IN       AL,DX
00DD  AA                           STOSB
```

```
                                        ; LOOP IF MORE NCB TO READ IN

)0DE  E2 F3                             LOOP      NEXT_NCB_BYTE
)0E0  EB 18 90                          JMP       NCB_XFERD

                                        ; BYTE NOT READY.  WAIT UP TO "AWHILE" FOR IT

)0E3                            NCB_NOT_READY:
)0E3  BB 0000                           MOV       BX,AWHILE
)0E6                            NCB_NOT_READY2:
)0E6  EC                                IN        AL,DX
)0E7  84 C4                             TEST      AL,AH
)0E9  75 EF                             JNZ       GET_NCB_BYTE
)0EB  4B                                DEC       BX
)0EC  75 F8                             JNE       NCB_NOT_READY2

                                        ; TIMEOUT WHILE WAITING.  CATASTROPHIC ERROR!

00EE  E8 0249 R                         CALL      CATASTROPHIC_ERROR

                                        ; SET NCB RETURN CODE TO NCBSYS_ERR?

                                        RESTORE   <BX>                    ; SAVED FROM DI
00F1  5B                      +            POP       BX
00F2  26: C6 47 01 40                   MOV       ES:BYTE PTR [BX].NCB_RETCODE,NCBSYS_ERR?
00F7  EB 02 90                          JMP       CLEANUP

                                        ; NCB TRANSFERRED.  GET NCB BASE ADDRESS

00FA                            NCB_XFERD: RESTORE  <BX>                  ; SAVED FROM DI
00FA  5B                      +            POP       BX

                                        ; MASK INTERRUPTS, UNLOCK INTERFACE, TELL LANA WE'RE DONE

00FB  FA                        CLEANUP: CLI
00FC  80 26 00A3 BF                     AND       BYTE PTR DS:[LANA_1_STATUS],ALL_BITS-LANA_LOCKED
0101  BA 0368                           MOV       DX,LANA_1_SR
0104  EC                                IN        AL,DX
0105  24 FE                             AND       AL,ALL_BITS-GO
0107  EB 00                             JMP       $+2
0109  EE                                OUT       DX,AL

                                        ; RESTORE BUFFER@ TO SEG:OFF TYPE ADDR

010A  26: 8B 47 38                      MOV       AX,ES:WORD PTR [BX].NCB_RESERVE_BUFFER@
010E  26: 89 47 04                      MOV       ES:WORD PTR [BX].NCB_BUFFER@,AX
0112  26: 8B 47 3A                      MOV       AX,ES:WORD PTR [BX].NCB_RESERVE_BUFFER@+2
0116  26: 89 47 06                      MOV       ES:WORD PTR [BX].NCB_BUFFER@+2,AX

                                        ; IF NCB FOR MULTIPLE SEND RESTORE SECOND BUFFER@ TO SEG:OFF TYPE ADDR

011A  26: 8A 07                         MOV       AL,ES:[BX].NCB_COMMAND
011D  24 7F                             AND       AL,7FH
011F  3C 17                             CMP       AL,NCBSENDMULTIPLE
0121  75 10                             JNZ       UDT_RCD


0123  26: 8B 47 3C                      MOV       AX,ES:WORD PTR [BX].NCB_RESERVE_BUFFER2@
0127  26: 89 47 0C                      MOV       ES:WORD PTR [BX].NCB_BUFFER@+8,AX
012B  26: 8B 47 3E                      MOV       AX,ES:WORD PTR [BX].NCB_RESERVE_BUFFER2@+2
012F  26: 89 47 0E                      MOV       ES:WORD PTR [BX].NCB_BUFFER@+0AH,AX
                                        ;UPDATE NCB_CMD_CPLT

0133  26: 8A 47 01              UDT_RCD:  MOV       AL,ES:[BX].NCB_RETCODE
0137  26: 88 47 31                      MOV       ES:BYTE PTR [BX].NCB_CMD_CPLT,AL

                                        ; DO WE POST THIS NCB?

013B  26: F6 07 80                      TEST      ES:BYTE PTR [BX].NCB_COMMAND,NCBNO_WAIT
013F  75 09                             JNZ       NO_WAIT?
0141  B8 9180                           MOV       AX,DEV_BUSY_POST
0144  CD 15                             INT       15H
0146  FA                                CLI
0147  EB 18 90                          JMP       CMD_EXIT

014A  26: 83 7F 2C 00           NO_WAIT?: CMP       ES:WORD PTR [BX].NCB_POST@,0000H
014F  75 07                             JNZ       POST
0151  26: 83 7F 2E 00                   CMP       ES:WORD PTR [BX].NCB_POST@+2,0000H
0156  74 09                             JE        CMD_EXIT

                                        ; YES.  GET RETCODE & POST (AS AN INTERRUPT)


0158  26: 8A 47 01              POST:     MOV       AL,ES:[BX].NCB_RETCODE
015C  9C                                PUSHF
015D  26: FF 5F 2C                      CALL      ES:DWORD PTR [BX].NCB_POST@

                                        ; RETURN

0161                            CMD_EXIT: RESTORE   <SI,CX,DI,BX,ES>
0161  5E                      +            POP       SI
0162  59                      +            POP       CX
0163  5F                      +            POP       DI
0164  5B                      +            POP       BX
0165  07                      +            POP       ES
0166  C3                                RET

0167                            COMMAND_CPLT ENDP

                                ;___________________________________________________________________________
                                ;
                                ; LANA_DATA_REQ
                                ;
                                ;    SETS UP & STARTS A TRANFER OF DATA BETWEEN THE PC AND THE LANA.
                                ; WHILE DMA IS BUSY, THE PC WILL XFER "CHUNKS" VIA PROGRAMMED I/O.
                                ; WHEN DMA IS FREE, A DMA XFER OF THE REMAINDER WILL BE SETUP AND STARTED.
                                ;
                                ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:
                                ;
                                ;       REGISTER   |   ACCESS  |                 USAGE
                                ;       -----------+-----------+-------------------------------------------
                                ;       DS         |   CONST   | LO MEMEORY SEGMENT
                                ;          DX      |  DESTROY  | LANA'S PR PORT ADDR
                                ;          AL      |  DESTROY  | CMD CODE OF LANA REQUEST
                                ;          AH      |  DESTROY  | INTERNAL
                                ;
                                ; INTERRUPTS SHOULD BE MASKED ON ENTRY.
                                ;
                                ;___________________________________________________________________________

0167                            LANA_DATA_REQ PROC   NEAR

                                        SAVE      <CX,ES,DI>
```

```
0167  51                 +              PUSH     CX
0168  06                 +              PUSH     ES
0169  57                 +              PUSH     DI

                                   ; SAVE CMD CODE FOR LATER

016A  8A E0                        MOV       AH,AL

                                   ; LOCK INTERFACE

016C  80 0E 00A3 40                OR        BYTE PTR DS:[LANA_1_STATUS],LANA_LOCKED

                                   ; GET DATA'S PC ADDR & LEN

0171  E8 0260 R                    CALL      ADDR_AND_LEN

                                   ; SETUP FOR PROGRAMMED I/O (MAY LATER CHANGE TO DMA) IN THE CORRECT DIRECTION

0174  BA 036B                      MOV       DX,LANA_1_HIR
0177  EC                           IN        AL,DX                         ; GET CURRENT INTERFACE SETUP
0178  24 C7                        AND       AL,ALL_BITS-IO_METHOD-DD_BIT  ; MASK OFF AREA OF INTEREST
017A  80 FC 44                     CMP       AH,DATA_TO_LANA               ; IS IT TO OR FROM THE LANA?
017D  75 05                        JNE       FROM_LANA
017F  0C 08                        OR        AL,PROGRAMMED_IO+PC_TO_LANA   ; TO LANA
0181  EB 03 90                     JMP       DIR_SET
0184  0C 00             FROM_LANA: OR        AL,PROGRAMMED_IO+LANA_TO_PC   ; FROM LANA
0186  EE                DIR_SET:   OUT       DX,AL

                                   ; DMA_3 BUSY?

0187  F6 06 00A1 80     TRY_DMA_3: TEST      BYTE PTR DS:[LANA_HARDWARE],DMA_3_BUSY
018C  74 18                        JZ        GET_DMA_3

                                   ; YEP.  PROGRAM I/O A "CHUNK"

018E  FB                PROGRAM_IO:STI                           ; ALLOW INTERRUPTS & HOPE THAT DMA FREES UP
018F  E8 01B9 R                    CALL      PROGRAM_IO_CHUNK
0192  FA                           CLI

                                   ; CATASTROPHIC ERROR HAPPEN?

0193  F6 06 00A3 04                TEST      BYTE PTR DS:[LANA_1_STATUS],LANA_HARD_ERR
0198  75 16                        JNZ       DATA_EXIT             ; IF SO, EXIT

                                   ; NOPE.  MORE DATA LEFT TO XFER?

019A  83 F9 00                     CMP       CX,0
019D  74 11                        JZ        DATA_EXIT             ; IF NOT, EXIT

                                   ; YES.  WORTH TRYING FOR DMA?

019F  83 F9 70                     CMP       CX,CHUNK_SIZE
01A2  73 E3                        JNB       TRY_DMA_3
01A4  EB E8                        JMP       PROGRAM_IO

                                   ; SETUP & START DMA (OF REMAINING DMA)

01A6  BA 036B           GET_DMA_3: MOV       DX,LANA_1_HIR
01A9  50                           PUSH      AX
01AA  B0 02                        MOV       AL,02                 ;INDICATES LANA 2 REQUESTING DMA SERVICE
01AC  E8 0000 E                    CALL      DMA_START_UP
01AF  58                           POP       AX
                                   ; UNLOCK INTERFACE AND RETURN

01B0  80 26 00A3 BF     DATA_EXIT: AND       BYTE PTR DS:[LANA_1_STATUS],ALL_BITS-LANA_LOCKED
                                   RESTORE   <DI,ES,CX>
01B5  5F                 +              POP      DI
01B6  07                 +              POP      ES
01B7  59                 +              POP      CX
01B8  C3                           RET

01B9                    LANA_DATA_REQ ENDP

                        ;--------------------------------------------------------------------------
                        ; PROGRAM_IO_CHUNK
                        ;
                        ;    PROGRAM I/O'S A "CHUNK" OF THE REMAINING DATA BETWEEN THE PC & LANA.
                        ;
                        ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:
                        ;
                        ;       REGISTER    |  ACCESS  |                USAGE
                        ;       ------------+----------+-------------------------------------
                        ;       DS          |  CONST   | LO MEMEORY SEGMENT
                        ;          AH       |  CONST   | CMD CODE OF LANA REQ
                        ;       ES:DI       |   VAR    | START OF "CHUNK" IN MEMORY
                        ;          CX       |   VAR    | LENGTH OF REMAINING DATA
                        ;          DX       | DESTROY  | INTERNAL
                        ;
                        ; INTERRUPTS SHOULD BE UNMASKED ON ENTRY.
                        ;
                        ;--------------------------------------------------------------------------

01B9                    PROGRAM_IO_CHUNK PROC NEAR

                                   SAVE      <SI,AX,BX>
01B9  56                 +              PUSH     SI
01BA  50                 +              PUSH     AX
01BB  53                 +              PUSH     BX

                                   ; SETUP CHUNK SIZE, DATA PORT ADDR, UPDATE "REMAINING DATA" SIZE

01BC  BA 0070                      MOV       DX,CHUNK_SIZE         ; THIS_CHUNK := MIN(CHUNK_SIZE,REMAINING_DATA)
01BF  3B CA                        CMP       CX,DX
01C1  73 02                        JNB       GOT_CHUNK
01C3  8B D1                        MOV       DX,CX
01C5  2B CA             GOT_CHUNK: SUB       CX,DX                 ; REMAINING_DATA := REMAINING_DATA - THIS_CHUNK
                                   SAVE      <CX>
01C7  51                 +              PUSH     CX
01C8  8B CA                        MOV       CX,DX
01CA  BA 036A                      MOV       DX,LANA_1_DR

                                   ; DATA TO LANA?

01CD  80 FC 44                     CMP       AH,DATA_TO_LANA
01D0  75 4F                        JNE       FROM_LANA2

                                   ; YEP.  SETUP SOURCE PTR & PORT ADDRS

01D2  8B F7                        MOV       SI,DI
                                   SWAP      ES,DS
01D4  06                 +              PUSH         ES
01D5  1E                 +              PUSH         DS
01D6  07                 +              POP          ES
```

```
01D7  1F                  +             POP       DS
01D8  BF 0368                     MOV       DI,LANA_1_SR
01DB  B4 10                       MOV       AH,DRE
01DD  EB 0A 90                    JMP       OUT_TO_LANA2

                                  ; ROOM IN DATA REGISTER?

01E0  87 FA                DATA_ROOM?:XCHG   DI,DX
01E2  EC                          IN        AL,DX
01E3  84 C4                       TEST      AL,AH                   ; ROOM IN DATA REGISTER?
01E5  74 09                       JZ        NO_ROOM

                                  ; STUFF BYTE INTO DATA REGISTER

01E7                       OUT_TO_LANA:
01E7  87 D7                       XCHG      DX,DI
01E9                       OUT_TO_LANA2:
01E9  AC                          LODSB
01EA  EE                          OUT       DX,AL

                                  ; LOOP IF MORE "CHUNK" TO SEND

01EB  E2 F3                       LOOP      DATA_ROOM?
01ED  EB 15 90                    JMP       LANA_GOT_ALL?

                                  ; WAIT UP TO "AWHILE" FOR DRE

01F0  BB 0000              NO_ROOM:   MOV   BX,AWHILE
01F3  EC                   ROOM_NOW?: IN    AL,DX
01F4  84 C4                       TEST      AL,AH
01F6  75 EF                       JNZ       OUT_TO_LANA
01F8  4B                          DEC       BX                      ; TIMEOUT EXPIRED?
01F9  75 F8                       JNE       ROOM_NOW?               ; IF NOT, TRY AGAIN
                                  SWAP      ES,DS                   ; RESTORE REGISTERS TO "ON ENTRY" SETUP
01FB  06                  +             PUSH      ES
01FC  1E                  +             PUSH      DS
01FD  07                  +             POP       ES
01FE  1F                  +             POP       DS
01FF  8B FE                       MOV       DI,SI
0201  EB 3E 90                    JMP       IO_TIMEOUT

                                  ; HAS LANA GOT ALL OF THE "CHUNK" YET?

0204                       LANA_GOT_ALL?:
                                  SWAP      ES,DS                   ; RESTORE REGISTERS TO "ON ENTRY" SETUP
0204  06                  +             PUSH      ES
0205  1E                  +             PUSH      DS
0206  07                  +             POP       ES
0207  1F                  +             POP       DS
0208  8B D7                       MOV       DX,DI
020A  8B FE                       MOV       DI,SI
020C  B4 20                       MOV       AH,DRF
020E  EC                          IN        AL,DX
020F  84 C4                       TEST      AL,AH                   ; DATA REGISTER TOTALLY EMPTY?
0211  74 31                       JZ        CHUNK_DONE              ; IF SO, CHUNK IS DONE

                                  ; NO.  WAIT UP TO "AWHILE" FOR LANA TO GET ALL OF IT

0213  BB 0000                     MOV       BX,AWHILE
0216                       GOT_ALL_NOW?:
0216  EC                          IN        AL,DX
0217  84 C4                       TEST      AL,AH                   ; DATA REGISTER TOTALLY EMPTY?
0219  74 29                       JZ        CHUNK_DONE              ; IF SO, CHUNK IS DONE
021B  4B                          DEC       BX                      ; TIMEOUT EXPIRED?
021C  75 F8                       JNE       GOT_ALL_NOW?            ; IF NOT, TRY AGAIN
021E  EB 21 90                    JMP       IO_TIMEOUT

                                  ; SETUP FOR I/O FROM LANA

0221  BE 0368              FROM_LANA2:MOV   SI,LANA_1_SR
0224  B4 20                       MOV       AH,DRF

                                  ; DATA BYTE WAITING?

0226  87 D6                ANY_DATA?: XCHG  DX,SI
0228  EC                          IN        AL,DX
0229  84 C4                       TEST      AL,AH
022B  74 09                       JZ        NO_DATA

                                  ; YES.  GET IT, STORE IT, LOOP IF MORE DATA REMAINS

022D                       IN_FROM_LANA:
022D  87 F2                       XCHG      SI,DX
022F  EC                          IN        AL,DX
0230  AA                          STOSB
0231  E2 F3                       LOOP      ANY_DATA?
0233  EB 0F 90                    JMP       CHUNK_DONE

                                  ; WAIT UP TO "AWHILE" FOR DATA BYTE FROM LANA

0236  BB 0000              NO_DATA:   MOV   BX,AWHILE
0239  EC                   DATA_NOW?: IN    AL,DX
023A  84 C4                       TEST      AL,AH                   ; DATA BYTE WAITING NOW?
023C  75 EF                       JNZ       IN_FROM_LANA            ; IF SO, GO GET IT
023E  4B                          DEC       BX                      ; TIMEOUT EXPIRED?
023F  75 F8                       JNE       DATA_NOW?               ; IF NOT, TRY AGAIN

                                  ; TIMEOUT WAITING ON I/O.  CATASTROPHIC ERROR

0241  E8 0249 R            IO_TIMEOUT:CALL  CATASTROPHIC_ERROR

                                  ; CHUNK DONE.  RETURN

0244                       CHUNK_DONE:RESTORE  <CX>
0244  59                  +             POP       CX
                                  RESTORE   <BX,AX,SI>
0245  5B                  +             POP       BX
0246  58                  +             POP       AX
0247  5E                  +             POP       SI
0248  C3                          RET

0249                       PROGRAM_IO_CHUNK ENDP

                           ;________________________________________________________________
                           ; CATASTROPHIC_ERROR                                             |
                           ;                                                                |
                           ;    HANDLES CATASTROPHIC INTERFACE ERRORS.                      |
                           ;                                                                |
                           ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                |
                           ;                                                                |
                           ;       REGISTER   |  ACCESS  |              USAGE               |
                           ;       -----------+----------+--------------------------------- |
                           ;       DS         |  CONST   | LO MEMEORY SEGMENT               |
                           ;          DX      | DESTROY  | INTERNAL                         |
```

```
                                        ;            AX       |  DESTROY  |  INTERNAL                                  |
                                        ;                                                                              |
                                        ;______________________________________________________________________________|

0249                                    CATASTROPHIC_ERROR PROC NEAR

                                                  ; IF LANA OWNS INTERFACE (WITH GO SET), SET CPLT_CODE TO "CANT_CPLT?"

0249  BA 0368                                     MOV      DX,LANA_1_SR
024C  EC                                          IN       AL,DX
024D  A8 80                                       TEST     AL,HC                       ; DOES LANA OWN INTERFACE?
024F  75 09                                       JNZ      SET_SPCL                    ; IF NOT, DON'T SET CPLT_CODE
0251  A8 01                                       TEST     AL,GO                       ; IS LANA WAITING ON THE PC?
0253  74 05                                       JZ       SET_SPCL                    ; IF NOT, DON'T SET CPLT_CODE
0255  24 F9                                       AND      AL,ALL_BITS-CPLT_CODE       ; ELSE, MASK OFF AREA OF INTEREST
0257  0C 04                                       OR       AL,CANT_CPLT?               ; AND SET AS DESIRED
0259  EE                                          OUT      DX,AL

                                                  ; SET LANA'S LANA_HARD_ERR

025A                                    SET_SPCL:

025A  80 0E 00A3 04                               OR       BYTE PTR DS:[LANA_1_STATUS],LANA_HARD_ERR

                                                  ; RETURN

025F  C3                                          RET

0260                                    CATASTROPHIC_ERROR ENDP

                                        ;______________________________________________________________________________
                                        ;                                                                              |
                                        ; ADDR_AND_LEN                                                                 |
                                        ;                                                                              |
                                        ;    GETS A 32-BIT ADDRESS AND 16-BIT LENGTH FROM LANA_1'S PR PORT.            |
                                        ; THE 32-BIT ADDRESS IS CONVERTED INTO A SEGMENT:OFFSET PAIR.                  |
                                        ;                                                                              |
                                        ;    THE FOLLWING CONVENTIONS SHOULD BE FOLLOWED:                              |
                                        ;                                                                              |
                                        ;       REGISTER  |  ACCESS  |                USAGE                            |
                                        ;       ----------+----------+-------------------------------------------     |
                                        ;          DX     |  CONST   | LANA'S PR PORT ADDR                             |
                                        ;        ES:DI    |  RESULT  | ADDRESS IN SEGMENT:OFFSET FORM                  |
                                        ;          CX     |  RESULT  | LENGTH                                          |
                                        ;                                                                              |
                                        ;______________________________________________________________________________|

0260                                    ADDR_AND_LEN PROC      NEAR

                                                  SAVE     <AX>
0260  50                              +              PUSH     AX

                                                  ; GET 32-BIT ADDRESS

0261  EC                                          IN       AL,DX
0262  8A C8                                       MOV      CL,AL
0264  81 E1 000F                                  AND      CX,000FH
0268  8B F9                                       MOV      DI,CX                       ; DI GET LOWEST 4 BITS OF ADDR
026A  8A C8                                       MOV      CL,AL
026C  EC                                          IN       AL,DX
026D  8A E8                                       MOV      CH,AL
026F  F8                                          CLC
0270  D1 E9                                       SHR      CX,1
0272  D1 E9                                       SHR      CX,1
0274  D1 E9                                       SHR      CX,1
0276  D1 E9                                       SHR      CX,1                        ; CX GET NEXT 12 BITS (TOP 4 BITS=0)
0278  EC                                          IN       AL,DX
0279  8A E0                                       MOV      AH,AL
027B  EC                                          IN       AL,DX
027C  D1 E0                                       SHL      AX,1
027E  D1 E0                                       SHL      AX,1
0280  D1 E0                                       SHL      AX,1
0282  D1 E0                                       SHL      AX,1
0284  25 F000                                     AND      AX,0F000H
0287  03 C1                                       ADD      AX,CX                       ; AX BECOMES SEGMENT TYPE ADDR
0289  8E C0                                       MOV      ES,AX

                                                  ; GET 16-BIT LENGTH AND RETURN

028B  EC                                          IN       AL,DX
028C  8A C8                                       MOV      CL,AL
028E  EC                                          IN       AL,DX
028F  8A E8                                       MOV      CH,AL

                                                  RESTORE  <AX>
0291  58                              +              POP      AX
0292  C3                                          RET

0293                                    ADDR_AND_LEN ENDP

0293                                    NETWORK   ENDS
                                                  END
```

Appendix E. Multitasking Considerations

If you use the wait option for the commands in a multitasking environment, a "hook" is provided for the multitasking program using interrupt 15H. When either a busy or wait loop occurs in NET BIOS, a "hook" is provided for the program to break out of the loop. To distinguish individual calls, look at the ES:BX register which points to the NCB. The "hook" is also used when NET BIOS is servicing an interrupt, that in turn causes a corresponding wait loop, providing a means to break out of the loop. The steps necessary to service interrupt 15H are as follows:

When programming in the multitasking environment, the program has the responsibility to check the AX register for the following function codes:

AH contains:	AL contains:
90H	80H
91H	80H

Figure E-1 AX Register Function Codes

The program must pass all other functions through to the previous user of interrupt 15H. This can be accomplished by either a JMP or a CALL. With either a 90H or 91H function code in the AH register, the program performs the necessary processing and returns using an IRET instruction. An 80H in the AL register indicates that NET BIOS issued the interrupt.

9080H This function code is in the AX register whenever NET BIOS is about to enter either a busy or a wait loop. NET BIOS also issues an interrupt 15H at this time to signal the program of the loop. When this occurs, the program saves the task status and dispatches another task. This allows overlapping execution of tasks when the hardware is busy.

9180H This function code is in the AX register whenever NET BIOS has set an interrupt flag for a corresponding busy loop. NET BIOS also issues an interrupt 15H at this time. This code is used to signal a POST condition and the program sets the task status to "ready to run" before returning.

Glossary

active circuit. A circuit or device that requires electrical power to operate.

address. A number specifying a particular user device attachment point.

alias. An alternate name that you can be known by on the network.

allocations. The assignments of frequencies by the FCC for various communications uses (for example; television, radio, land-mobile, defense, microwave, etc.) The assigned frequencies are to achieve a fair division of the available spectrum and to minimize interference among users.

amplifier. A device used to boost the strength (dB level) of an electronic signal. Amplifiers are spaced at intervals throughout a cable system to rebuild the strength of TV or data signals that weaken as they pass through the cable network. Midsplit configurations use a forward and a reverse amplifier in the same enclosure to boost signals in both directions.

balancing (signal). A method of equalizing the attenuation that a particular signal encounters through the network (forward direction) so that the signal level is essentially the same at all outlets. Balancing also produces near equal inputs to the frequency translator from a fixed level transmitter (reverse), no matter where the transmitter is attached to the network.

bandwidth. A measure of spectrum (frequency) use or capacity. For instance, a voice transmission by telephone requires a bandwidth of about 3000 cycles per second (3 kHz). A TV channel occupies a bandwidth of 6 million cycles per second (6 MHz).

BIOS. Basic Input Output System.

branch. An intermediate cable distribution line in a broadband coaxial network that either feeds or is fed from a main trunk. Also referred to as a feeder.

bridge. A specialized device containing programs and network attachments. It is used to route messages between the same network, on a different broadband network or both.

broadband. A general term used to describe wide bandwidth equipment or systems that can carry a large proportion of the electromagnetic spectrum. A broadband communications system can accommodate all broadcast and many other services.

cable kit. An 8-port splitter device used to connect the Personal Computers to the network.

cable loss. The amount of rf signal attenuation by coaxial cable transmission. The amount of cable attenuation is a function of frequency and cable distance. High frequencies have a greater loss than low frequencies and follow a logarithmic function. Cable losses are usually calculated for the highest frequency carried on the cable.

Cable powering. Supplying operating power to active CATV equipment by using the coaxial cable to carry this power along with the information signal.

cable tilt. A reduction in the level of an RF sweep signal passing through a cable as it sweeps from low to high frequency. This "tilt" is caused by the increase in cable attenuation as the frequency increases. A specific fixed length of cable and a fixed frequency range produces a fixed amount of tilt.

Cable TV. Previously called Community Antenna Television (CATV). A communication system that distributes broadcast programs simultaneously via a coaxial cable.

carrier sense multiple access with collision detection (CSMA/CD). A technique by which many independent nodes can share a common broadcast communication channel without requiring a central transmission allocation authority.

CATV. See Cable TV.

composite video signal. The complete video signal. For monochrome, it consists of the picture signal, blanking and synchronizing signals. For color, additional color synchronizing signals and color picture information are added.

coaxial cable. Coaxial means that two conductors and the dialectic share the same axis - the center of the cable. One conductor is the center wire, while the other is the shield and is referenced to ground. The shield and center conductor are separated by an insulating dialectic made of polyethylene.

CRC. See cyclic redundancy check.

cross modulation. A form of signal distortion in which modulation from one or more RF carrier(s) is imposed on another carrier.

CSMA/CD. See carrier sense multiple access with collision detection

cyclic redundancy check. A numeric value derived from the bits in a message that is used to check a message for any bit errors in transmission.

datagram. A particular type of information encapsulation at the network layer of the adapter protocol. No explicit acknowledgment for the information is sent by the receiver. Instead, transmission relies on the "best effort" of the link layer.

data rate. The rate at which data is transferred within a processor and between a processor and an external device. This rate is usually expressed in units of bits per second (bps).

dB. An abbreviation for decibel, used as a relative unit of measure between two signals on a logarithmic basis. dB is an expression of a ratio between an input level and an output level.

dBmV. An abbreviation for decibel millivolt. The level at any point in a system expressed in dB's above or below a 1 millivolt/75 ohm standard is the level in decibel millivolts (dBmV). Zero dBmV is equal to 1 millivolt across 75 ohms.

default. The default value of a setting is the original one, which is in effect until other instructions are entered.

directional coupler. A high quality tapping device providing isolation between a single tap outlet drop line and external devices (can be more than one).

distribution amplifier. An amplifier used to increase rf signal levels to overcome cable and flat loss for user distribution.

DMA. Direct Memory Access.

drop cable. A flexible coaxial cable that extends from a tap on the coaxial network. The end of the drop cable has the network outlet connector, which is used to attach an external device. Also referred to as a drop line.

drop-line device. Any external device attached to the coaxial network through a drop cable, for example a TV set, audio modulator, or adapter.

echo. See reflections.

equalization. A means of modifying the frequency response of an amplifier or network, thereby resulting in a flat overall response. It is slope compensation done by a module within an amplifier enclosure.

F connector. A type of connector used by the CATV industry to connect a coaxial cable to equipment.

FDM. Frequency Division Multiplex. See frequency division multiplexing.

feeder (cable). Same as a branch.

filter. A circuit that selects one or more components of a signal depending on their frequency. Used in trunk and branch lines for special cable services such as two-way operation.

flat loss. Equal loss at all frequencies, such as that caused by attenuators.

flooded cable. A special CATV cable containing a corrosion-resistant gel between the outer aluminum sheath and the outer jacket. The gel flows into imperfections in the aluminum to prevent corrosion in high moisture areas.

forward direction. The direction of signal flow away from the frequency translator.

frequency. The number of times an electromagnetic signal repeats an identical cycle in a unit of time, usually one second. One Hertz (Hz) is one cycle per second. A kHz (Kilohertz) is one thousand cycles per second; a MHz (Megahertz) is one million cycles per second; a GHz (Gigahertz) is one billion cycles per second.

frequency division multiplexing. A method of dividing a communication channel bandwidth among several subchannels with different carrier frequencies. Each subchannel can carry separate data signals.

frequency response. The change of gain with frequency.

frequency translator. In a mid-split configuration, an active electronic circuit in the headend that picks up information signals on one 6 MHz channel, coming in from the reverse direction—converts them to another 6 MHz channel above the mid-split frequency and sends them out in the forward direction.

FSK. Frequency Shift Keying.

full duplex. A connection on the network that allows transmissions in both directions at the same time.

gateway. A protocol-translating interface between an adapter (and its protocols) and an external network that uses a distinctly different protocol suite.

harmonic distortion. Form of interference involving the generation of harmonics according to the frequency relationship f = (n)f for each frequency present, where n is a whole number equal to 2^1 or more.

headend. The location of the frequency translator or an electronic control center, generally located at the antenna site of a CATV system, usually including antennas, preamplifiers, frequency converters, demodulators, modulators and other related equipment that amplify, filter and convert incoming broadcast TV signals to cable system channels. See also frequency translator.

high frequencies. Frequencies from 160MHz to 400MHz allocated for the forward direction in a mid-split system.

host concept. Many protocols such as IBM's SNA for example, employ some large data processing facility as part of the network.

hub. The same as a headend for bidirectional networks, except that it is more centrally located within the network.

insertion loss. Additional loss in a system when a device such as a directional coupler is inserted; equal to the difference in signal level between input and output of such a device.

isolation loss. The amount of signal attenuation in a passive device from input port to tap outlet port.

LAN. Local Area Network.

LANA. Local Area Network Adapter.

local session number. The number assigned to each session established by an adapter. Each session receives a unique number that distinguishes it from any other active sessions.

low frequencies. Frequencies from 5MHz to 116MHz allocated for the return direction in a mid-split system.

LSN. See local session number.

main trunk. The major link(s) from the headend (or hub) to downstream branches.

message. A message is a logical partition of the user device's data stream to and from the adapter.

mid-band. The part of the frequency band that lies between television channels 6 and 7, reserved by the FCC for air, maritime and land mobile units, FM radio and aeronautical and maritime navigation. Mid-band frequencies, 108 to 174 MHz, can also be used to provide additional channels on cable television systems.

mid-split. A method of frequency division that allows two-way traffic on a single cable. Incoming signals go to the frequency translator between 5-116MHz; outgoing signals go from the frequency translator between 168-400MHz. No signals are present between 116-162MHz.

multitap. A passive distribution component composed of a directional coupler and a splitter with two or more output connections.

node. Consists of a personal computer, an adapter with a cable and other adapters to the Personal Computer (such as; disk drives, printers, and plotters). Along with the Personal Computer hardware, the necessary software must be available.

noise. The word "noise" is a carry-over from audio practice. Refers to random spurts of electrical energy or interference.

noise figure. A measure of the amount of noise in dB generated at the input of an amplifier as compared with the noise generated by a 75-ohm resistor.

packet. A unit of the protocol used by the transport layer. The packet contains header control information, as well as user data.

parity. The checksum of each data byte transmitted or received. Each 1 bit is counted in a byte. The number of odd or even 1 bits in the byte is the parity. Parity may be even, odd, or none.

passive circuit. A circuit or device that does not require electrical power to operate.

point-to-point. A connection between two and only two nodes on a network.

PROM. Programmable Read Only Memory.

protocol. A procedure for ordering the exchange of formatted information packets between correspondents. Protocols are "interpreted" by hardware and software within the adapter. See the protocol section in Chapter 2.

RAM. Random Access Memory.

receiver isolation. The attenuation between any two receivers that are connected to the system.

rf. Radio frequency.

rf modem. A modulator-demodulator device that codes or decodes a digital information signal. The modulator part of the rf modem codes the digital information onto an analog signal by varying the frequency of the carrier signal. The demodulator part extracts the digital information from a modulated carrier signal.

reflections. Signal waves reflected from components within the network, which are the result of impedance or mismatches in the transmission coax medium. Also called echoes.

return loss. Reflection coefficient expressed in dB.

return path. See reverse direction.

reverse direction. The direction of signal flow toward the frequency translator.

reverse path. See reverse direction.

ROM. Read Only Memory.

session. The data transport connection resulting from a call between two user devices.

signal ingress. This is a signal or signals that enter into the cable or cable system from an outside source, such as an RF transmitting tower (AM or FM).

signal level. The root-mean-square (rms) voltage measured during the peak of the RF signal. It is usually expressed in microvolts referred to an impedance of 75 ohms, or in dBmV.

signal-to-noise ratio. The relative power of the signal to the noise on the cable.

slope. The difference between the signal levels at the highest frequency and the lowest frequency in a network. Slope is sometimes referred to as spectrum tilt.

slope compensation. The action of a slope-compensated gain control. The gain of an amplifier and the slope of the amplifiers equalization circuit are simultaneously changed to provide the correct cable equalization for different lengths of cable. This is normally specified in terms of cable loss.

splitter. A passive, 5 MHz–300 MHz or 800 MHz bandpass device. The device is coupled in-line to a main trunk or branch for splitting the power and the information signal two or more ways on a coaxial network. Splitters always pass through 60Hz power to the network, if used.

subsplit. A method of frequency division that allows two-way traffic on a single cable. Incoming signals go to the frequency translator between 5-30MHz; outgoing signals go from the frequency translator between 54-400MHz. No signals occupy 30-54MHz.

tap. A passive 5 MHz–300 MHz box-like device, normally installed in line with a broadband branch cable. Passive circuits tap off only the information signal to its small Type F outlet ports.

tap outlet. A Type F connector port on a tap used to attach a drop cable. The information signal is carried through this port. The number of outlets on a branch line tap normally varies from 2 to 8.

TDM. See time division multiplexing.

terminator. A 75-ohm resistive connector used to terminate the end of a cable or an unused tap. The device is used to minimize cable reflections.

tilt compensation. See slope compensation.

time division multiplexing. A method of sharing a communication channel among several users by allowing each to use the channel for a given period of time in a defined, repeated sequence.

trunk line. See main trunk.

unity gain. A standard design parameter used in CATV network amplifiers. The amplifier is designed to compensate for cable signal loss and flat loss. It also implies that the output of any amplifier is equal to or less than the output of the previous cascaded amplifier (forward or reverse) in the network.

virtual connection. A connection between two nodes on the network that is established using the transport layer and provides reliable data transfer between the nodes.

Bibliography

The following is a list of related publications.

- Intel 82586 Reference Manual
- Intel iAPX188 Data Sheets
- Sytek Serial Interface Controller (SIC) Data Sheet
- The *IBM Macro Assembler*

INDEX

Special Characters

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

IBM

The Personal Computer
Hardware Reference Library

Reader's Comment Form

Technical Reference, PC Network 6322916

Your comments assist us in improving the usefulness of our publication; they are an important part of the input used for revisions.

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Please do not use this form for technical questions regarding the IBM Personal Computer or programs for the IBM Personal Computer, or for requests for additional publications; this only delays the response. Instead, direct your inquiries or request to your authorized IBM Personal Computer dealer.

Comments:

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 321 BOCA RATON, FLORIDA 33432

POSTAGE WILL BE PAID BY ADDRESSEE

IBM PERSONAL COMPUTER
SALES & SERVICE
P.O. BOX 1328-C
BOCA RATON, FLORIDA 33432

Fold here

Please do not staple

Tape

Continued from inside front cover

SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

IBM does not warrant that the functions contained in the program will meet your requirements or that the operation of the program will be uninterrupted or error free.

However, IBM warrants the diskette(s) or cassette(s) on which the program is furnished, to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery to you as evidenced by a copy of your receipt.

LIMITATIONS OF REMEDIES

IBM's entire liability and your exclusive remedy shall be:

1. the replacement of any diskette(s) or cassette(s) not meeting IBM's "Limited Warranty" and which is returned to IBM or an authorized IBM PERSONAL COMPUTER dealer with a copy of your receipt, or
2. if IBM or the dealer is unable to deliver a replacement diskette(s) or cassette(s) which is free of defects in materials or workmanship, you may terminate this Agreement by returning the program and your money will be refunded.

IN NO EVENT WILL IBM BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH PROGRAM EVEN IF IBM OR AN AUTHORIZED IBM PERSONAL COMPUTER DEALER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

GENERAL

You may not sublicense, assign or transfer the license or the program except as expressly provided in this Agreement. Any attempt otherwise to sublicense, assign or transfer any of the rights, duties or obligations hereunder is void.

This Agreement will be governed by the laws of the State of Florida.

Should you have any questions concerning this Agreement, you may contact IBM by writing to IBM Personal Computer, Sales and Service, P.O. Box 1328-W, Boca Raton, Florida 33432.

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU FURTHER AGREE THAT IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN US WHICH SUPERSEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN US RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

International Business Machines Corporation

P.O. Box 1328-C
Boca Raton, Florida 33432

6322916

Printed in the United States of America