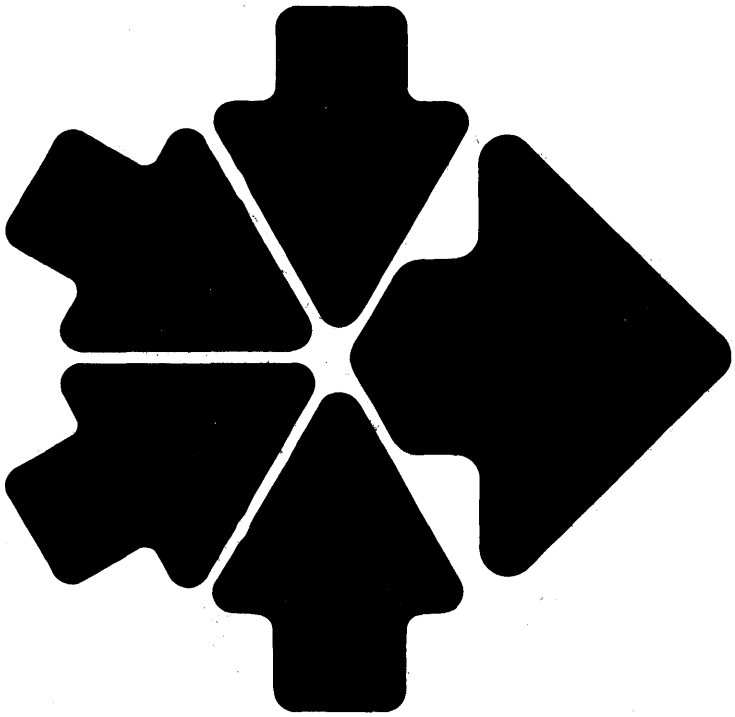
INTERPHASE

corporation

L N C 5 1 8 0

LOCAL AREA NETWORK CONTROLLER

USER's GUIDE



We Stay Ahead Of Our Competition . . .

So You Can Stay Ahead Of Yours.

* Note: [illegible] LNC [illegible]

L N C   5 1 8 0

LOCAL AREA NETWORK CONTROLLER

USER's GUIDE

OCTOBER, 1984

~~SEPTEMBER~~, 1983 --- VERSION 1.2

INFORMATION IN THIS USER DOCUMENT SUPERSEDES ANY PRELIMINARY SPECIFICATION AND DATA SHEETS, AND IS BELIEVED TO BE ACCURATE AND RELIABLE. INTERPHASE RESERVES THE RIGHT TO MAKE PRODUCT IMPROVEMENTS WITHOUT PRIOR NOTICE. SUCH IMPROVEMENTS MAY INCLUDE EXPANDED COMMAND AND ERROR CODES.

TABLE OF CONTENTS

TABLE OF CONTENTS
PAGE 2

THE LNC 5180 -- AN OVERVIEW

The LNC 5180 is a member of the Interphase family of intelligent Local Area Network Controllers. It is an easy to use, yet powerful and flexible single board controller. Its purpose is to allow the transfer of data in system memory of one MULTIBUS system (or network node) to system memory of another MULTIBUS system (or network node), without CPU intervention, at a rate of up to 2 Mbits/sec. LNC is resident in each end of the transaction and handles all protocol control, data packet formation, recovery from transmission errors, DMA into and out of MULTIBUS system memory, and notification of the CPU that the transaction is complete.

The LNC 5180 is a very intelligent, bipolar microprocessor based, firmware driven controller capable of assuming a number of different personalities or operating modes. It communicates with other LNCs (or other devices) on a network using an IBM defined bit stream protocol known as SDLC. The LNC can be a network MASTER (Primary station) or a Slave (Secondary station) in a MASTER/SLAVE type network typical of SDLC, and can send data to or request data from any other LNC attached to the network. The transaction is totally automatic after the initiating CPU builds a simple IOPB (I/O Parameter Block) in system memory and queues up the transaction. The transaction can include sending up to 65,536 bytes of data which will automatically be broken into multiple smaller packets of data. The receipt of each packet is positively acknowledged and any packets received in error are automatically retransmitted. sliding window protocol?

The term MASTER refers only to the LNC that controls which node can start a transaction at a given time. Any node can talk to any other node.

In addition to automatic transaction queueing, automatic error recovery, and automatic polling of Slaves by the network Master, the LNC even allows for dynamic buffer allocation using an optional feature called Memory Management.

Due to the firmware nature of the LNC, it is also adaptable to use in applications that do not use its normal MASTER/SLAVE modes. A number of higher level protocols such as HDLC, X.25, IBM, SNA, and others use an SLDC style bit stream and the SLDC facilities to begin and end data packets and to frame up information inside the packet. By the use of custom firmware the LNC 5180 is capable of emulating most higher level protocols so long as SLDC is the basis of the bit stream.

NETWORK ISO MODEL

The LNC 5180 conforms to the standard "ISO-MODEL" for layered networks and includes layers 2 through 4 and, with use of certain options and in conjunction with the operating system, also includes layer 5.

LAYER	ISO	LNC 5180	ETHERNET CONTROLLERS
7	APPLICATION	NO	NO
6	PRESENTATION	NO	NO
5	SESSION	YES (OPTIONAL)	NO
4	TRANSPORT	YES	NO
3	NETWORK	YES	NO
2	DATALINK	YES	YES
1	PHYSICAL	NO	NO

For comparison with other single board LNC controllers the above table shows that the LNC 5180 includes 3 more layers of function, operating at bipolar processor speeds, than any known ETHERNET controller. All known ETHERNET controllers include only the DATALINK layer and require all of the higher layers to be implemented in system software, typically operating at MOS processor speeds and requiring substantial memory space and development time.

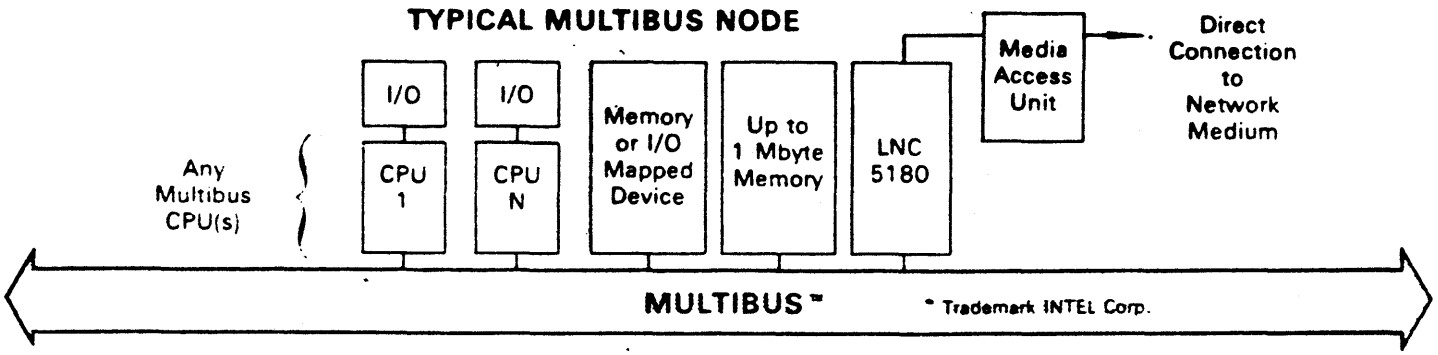
The Physical layer is the actual physical hookup (modem, modulator, transceiver, transporter, etc.) to the network transmission media and has been specifically excluded from the LNC 5180 to allow for useability in a wide range of network types. A variety of physical layer devices are available from Interphase and others.

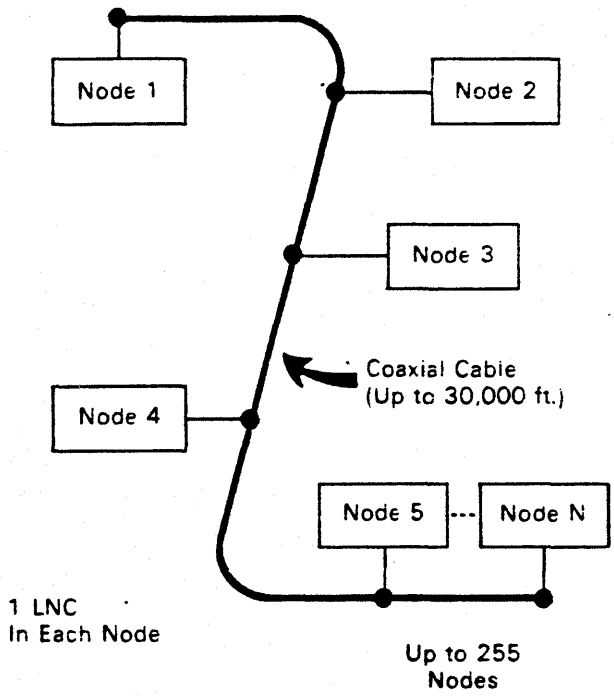
MEDIA AND TOPOLOGY INDEPENDENCE

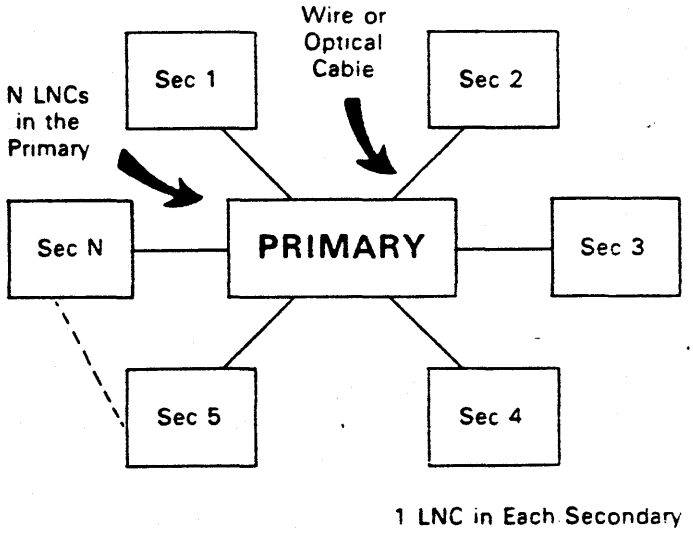
Since the LNC 5180 excludes the Physical Layer, or MEDIA ACCESS UNIT (MAU), it can be attached to a wide variety of network media such as baseband coax, broadband coax, optical fiber, microwave, T1 carrier (1.544 Mb), Bell DDS (56 Kb), wide band service and others. The data rate of the LNC 5180 is largely dependent on the media and MAU and varies from 0-2 Mbits/sec. The interface is, of course, well defined, and is RS422 compatible. This allows direct connection to standard modems in some circumstances.

As shown below the LNC 5180 can be used in a wide variety of network topologies including a multidrop bus, a STAR (point to point) and any other style of physical connection.

In addition it is entirely possible to implement in system software, or with custom LNC Firmware, a RING, LOOP, token passing or other logical network control protocol.







AN INTRODUCTION TO USING THE LNC 5180

The LNC 5180 is very simple to use from the programming point of view, yet the LNC offers a wide variety of features, options, and modes of operation. In many applications only a few of the features may be used, while others may use all of them. This section attempts to show the rudimentary operating features without describing any of the options and exotic features which might otherwise confuse the issue. Later sections describe use of other features in a building block fashion. If something in this section seems to be "left out" you are probably right. Later sections should fill in the gaps.

The following procedures are a short form of how to operate the LNC 5180:

1. INITIALIZATION

On power up the LNC must be told its identity. An "INITIALIZATION BLOCK" is built in system memory. This contains such information as the network node address, operating mode (MASTER, SLAVE, ETC.), maximum packet size, number of retries allowed, a pointer to the "CONTROL BLOCK" and certain other data that does not vary from transaction to transaction. After the User builds the "INITIALIZATION BLOCK" he simply writes the memory address of the "INITIALIZATION BLOCK" into a hardware write only register and turns on a bit in a hardware write only register. The LNC automatically "INITIALIZES" itself.

2. SEND A BLOCK OF DATA

Suppose the User wishes to send a block of data that is 350 bytes long and starts at his address 1000H, to a remote system memory starting at location 2400H. Further, his node address is 01H and the remote node address is 47H.

The user first builds an IOPB (I/O Parameter Block) similar to the one below in system memory. Remember everything fancy is left out for now.

SIMPLIFIED IOPB

07 ~~08~~	(SEND DATA)	(COMMAND CODE)
00		(STATUS) (LNC FILLS THIS IN LATER)
00		(ERROR CODE) (LNC FILLS THIS IN LATER)
00	(LSB)	(LOCAL BUFFER POINTER)
10	(MSB)	
00	(XSB)	
47		(DESTINATION NODE ADDRESS)
5E	(LSB)	(TRANSACTION BYTE COUNT)
01	(MSB)	
00	(LSB)	(REMOTE BUFFER POINTER)
24	(MSB)	
00	(XSB)	

The user then "QUEUES UP" the transaction by writing the address of the IOPB and a "GO" bit into the queue control byte of the "CONTROL BLOCK". The transaction is then automatically completed by the local LNC and the one on the other end.

No other LNCs that may be attached to the network are involved. Upon completion the local CPU is interrupted and can read the "STATUS" byte of the IOPB. If STATUS=02 the transaction was completed successfully. If there was an unrecoverable error, STATUS=03 and the ERROR CODE has an entry descriptive of the cause of the error. If interrupts are not used the User can simply observe the STATUS byte for completion. While the LNC is working on the job it will put a 01 into STATUS. On completion it will put either a 02 or 03 into STATUS.

3. RECEIVING A BLOCK OF DATA (ASYCHRONOUSLY)

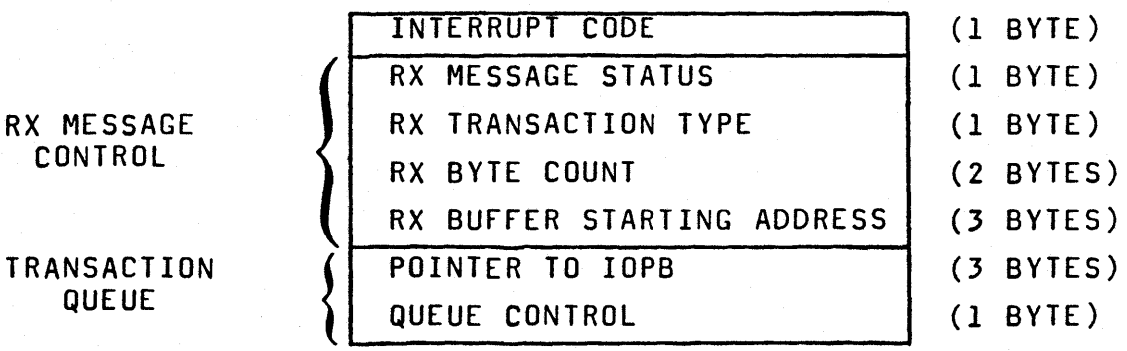
If you are on the receiving end of a Send Data Block transaction you will receive an interrupt. In the "CONTROL BLOCK" you will find information as to the type of transaction that took place, a starting block address and byte count, and a Receive Message status byte.
Before the next message can be received an entry in the CONTROL BLOCK must be cleared.

4. REQUESTING A DATA BLOCK

Most, if not all, competitive products allow only for the sending of a message. If data from the remote node is desired, a message is sent which must be interpreted by the software in the remote and a return message generated. This approach can certainly be used in a LNC 5180 based system, but many times it may be more convenient, and certainly more timely and efficient, for the transaction to take place automatically. With the LNC 5180 a User can request data in exactly the same way as he sends data. The only difference is that the command code in the IOPB is an ~~81~~ 09H instead of a ~~40~~ 07H.

5. SIMPLIFIED CONTROL BLOCK

The above transaction referred to the "CONTROL BLOCK" The CONTROL BLOCK is a block of system memory that is pointed to by the Initialization Block and stays fixed thereafter. It contains interrupt information, a receive message control section and a transaction queue section similar to that shown below (with all the fancy stuff left out).



ADVANCED CONCEPTS AND FEATURES

The LNC 5180, in addition to its rudimentary functions described earlier, has a number of high performance features and options described in this section. They are intended to be independent features and can be used individually or in combination as the needs of your system dictate. This section primarily deals with the advanced functions on a conceptual basis and leaves the details of use to the later sections.

TRANSACTION QUEUE

A "Transaction" is any function that the User causes the LNC to do, such as SEND DATA, REQUEST DATA, REQUEST STATUS, ETC. In a single task environment the User may simply determine what is to be done, tell the LNC to do it, and wait for a completion of that transaction before going to the next. Any Multi-User or Multi-Task environment, however, requires some way to stack up transactions for the communications function in the system to handle as fast as it can. This commonly requires a "QUEUEING" function to be handled by the Operating System, a sometimes very complicated and usually very inefficient part of the overall system software.

The LNC 5180 has a feature called the "TRANSACTION QUEUE" so that the operating software need not worry about this bothersome task even in a multiprocessor, multitask, or multiuser system. The number of Q-Slots are set by the Initialize procedure to be anything from 0 to 255 slots. ~~A selection of 0 turns off all Queueing functions and establishes a single "High Priority" Q-Slot.~~ A selection of N (1-255) slots turns on the Queueing function and establishes N slots ~~in addition to~~ including the High Priority Q-Slot. The $\frac{N}{(N-1)}$ normal priority Q-Slots are handled by the LNC 5180

A selection of Ø turns off all Queueing function and the node becomes a "receive only" device. It can not execute host level commands when initialized in this condition, however, it can receive ~~network traffic~~ LNC-8 data and properly report the receipt of messages.

in a round robin fashion described in a later section. Entries in the High Priority Q-Slot are handled on the very next transaction done by the LNC.

The Transaction Queue table is a section of the Control Block and includes three control bytes, a 4 byte High Priority Q-Slot, and (N-1) 4 byte normal priority transaction Q-Slots.

DATA REQUESTS

Most local area network products only send data messages. If the message is a request for data, that request is handled by the receiving station software. The LNC 5180 allows for a request for data to be handled directly by the receiving station. Two types of data requests are provided. The "REQUEST IMMEDIATE DATA" command causes the receiving station to immediatley fetch the data, format it and send it to the requesting station, all as part of the current transaction. Since no other network activity is happening during this transaction, this type of data request can be fairly inefficient and should be used advisedly with respect to the overall network needs, however, it is very effective in many fairly simple systems and is optimum if the need is for an instantaneous update of remote data in "real time".

The other type of data request, the "REQUEST DELAYED DATA" command is more elegant and does not incur any network overhead (wasted time) while waiting for a reply.

The "REQUEST DELAYED DATA" command causes the receiving station LNC to automatically queue up a SEND DATA command in one of several internal queues (not using the CONTROL BLOCK). The next time the station is allowed to start a transaction the data is sent back to the original requesting station, thereby satisfying the request.

MAILBOX

In many applications a User or a network node might routinely receive messages from a great many other nodes or Users within one or more network nodes. In those cases it is convenient for the User to have a "POST OFFICE" consisting of an array of pigeon holes or "MAILBOXES" with one "MAILBOX" for each potential source from which a transaction may be received.

Upon initialization the LNC is told the beginning memory address of the mailbox array which is always 256 bytes long. When a subsequent message is received the LNC observes the mailbox number, which was specified by the originating device in its IOPB, and puts a ØFFH into the corresponding mailbox before notifying the CPU. This can be used in any way the User software wishes. A common usage in cases where a memory image is supposed to be cyclically updated by another device on the network is to cyclically decrement all mailboxes (under a real time clock timer for example) and to spot network devices who have not provided the expected updates within a specified period of time.

The Mailbox feature may be disabled by initializing the MAILBOX STARTING ADDRESS to start at 0.

MEMORY MANAGEMENT

Although all discussions so far have assumed that a User will know the absolute memory addresses to send data to in a remote device, that is not always the case. Many times User software in the remote device will dynamically allocate buffer space for incoming messages. In other cases the operating system may routinely translate logical addresses to physical addresses.

In order for the LNC 5180 to do a DMA of data in the cases mentioned above, "MEMORY MANAGEMENT" is used. After receipt of the message, or the first packet of a long message, but before data is actually DMAed into memory the receiving station CPU is interrupted and allowed to modify the absolute addresses traveling with the message (in the header). A complete description of the process is presented in later sections (under Interrupt Handling). MEMORY MANAGEMENT can be "allowed" on a transaction by transaction basis or forced on all transactions based on initialization options.

POLLING

In a MASTER/SLAVE oriented system, the Master is in control of who starts a transaction on the network, and it is one of the Master's jobs to ask each Slave if it has a transaction it wishes to perform. Each Slave is given a slot of time to do its queued up transactions. This function is called polling, and the Master must poll each Slave often enough to satisfy the needs of the system. Most competitive products require the CPU software to cause the polling function to take place, and the LNC 5180 has a manual Poll command to allow this mode of operation.

In addition, however, the LNC 5180 has an automatic polling feature that can be turned ON and OFF under CPU command. The list of network nodes or devices to be polled (the "POLLING LIST") is under control of, and can be modified by, the CPU.

LINKED TRANSACTIONS

Multiple sequential transactions can be linked to one another by linking together their IOPBs. Only one Q-SLOT is used and once the first transaction is started all linked transactions are automatically completed (unless an unrecoverable error condition occurs) before the CPU is interrupted and notified of completion. This should not be confused with the QUEUEING function which treats all transactions independently.

DETAILED TECHNICAL DESCRIPTION

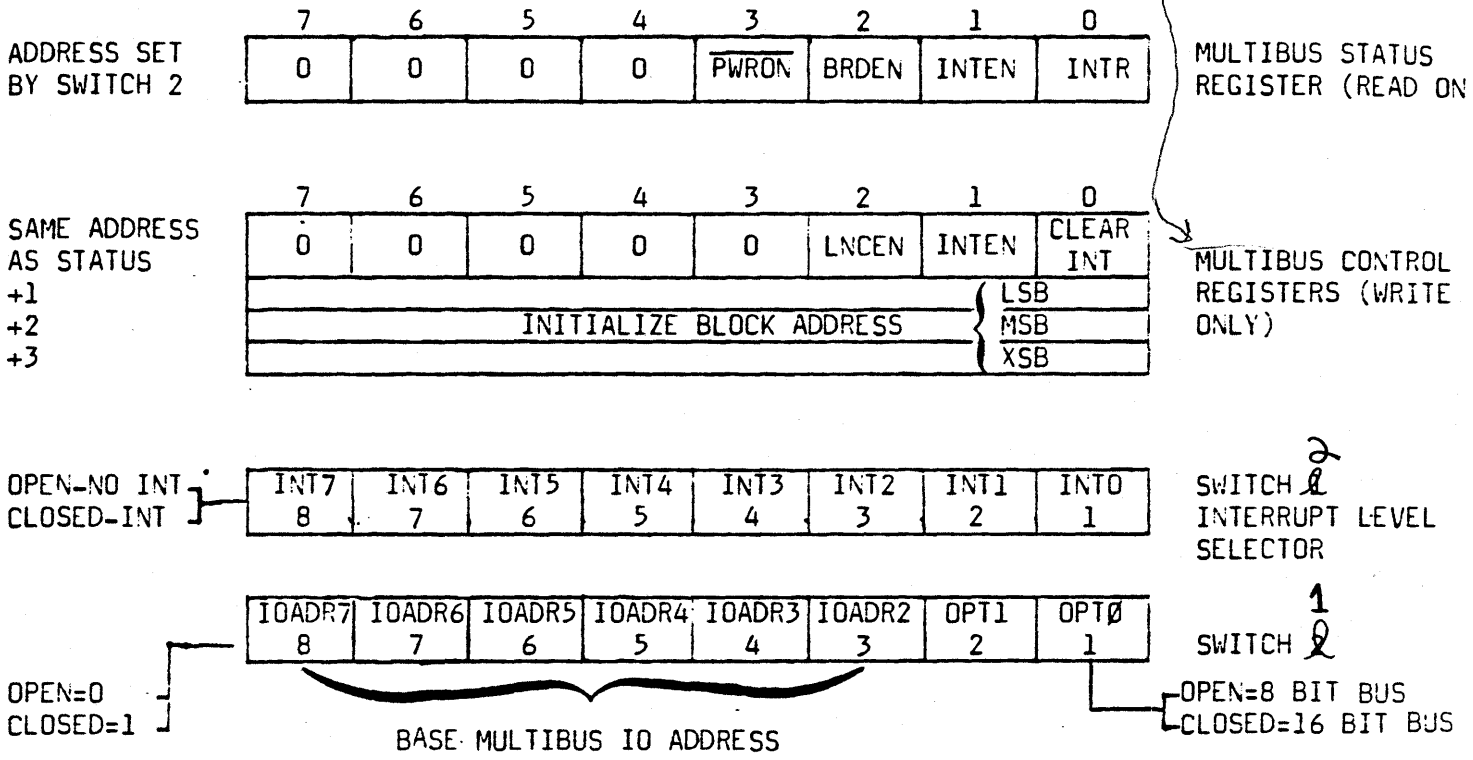
This section describes the hardware registers, INITIALIZE BLOCK, IOPB, CONTROL BLOCK, and other elements of the LNC 5180 in detail. While the LNC 5180 is simple to use, the extreme flexibility of the product makes these structures look complicated and confusing to the uninitiated. It is suggested that you first become familiar with the basic concepts described in earlier sections of this manual, if you have not yet done so, before reading this section. These sections refer extensively to Figures which are also duplicated in Appendix A for your convenience.

MULTIBUS I/O REGISTERS AND HARDWARE SWITCHES

The LNC 5180 has four Multibus Write Only Registers and one Multibus Ready Only register as shown in Figure 1. These registers are accessed via Multibus I/O Write and Read Commands (INPUT, OUTPUT, ETC.) by the CPU. The I/O address can be set via DIP switches (Switch 1 : positions 3-8) as shown in Figure 1. SWITCH 1 position 1 sets the Multibus data bus width as either 8 or 16 bits. Figure 1 also shows that DIP SWITCH 2 (positions 1-8) selects the Multibus hardware interrupt level (INT0-INT7) to be generated when an interrupt is generated. Only one position (or no positions) of SWITCH 2 should be closed.

The Write-Only registers consist of one control register and three address registers (see Figure 1). After power up the INITIALIZE BLOCK and CONTROL BLOCK sections of memory should be set up. Then the three address registers should be set up, via Multibus OUTPUT (or equivalent) commands, to point to the first byte of the INITIALIZE BLOCK.

Finally! a board that uses W.O.M.'s!



F I G U R E 1

Control Register:
After the address registers are set up the User can write a 1 to the LNCEN (bit 2) bit of the Control Register to enable the board. This "wakes up" the board and causes it to do the initialize function per the INITIALIZE BLOCK.

So long as the LNC is intended to be operational, the LNCEN bit should be left on. Turning the LNCEN bit OFF, for example, in case the CPU determines that the LNCEN has failed, disables the LNC (and its interrupts) and will require a reinitialize procedure to start it back up. Once it is turned ON the LNC 5180 operates out of the CONTROL BLOCK.

The other two bits in the Control Register ENABLE and CLEAR the interrupt. On power up all interrupts are disabled and the interrupt is cleared. To enable interrupts write a 1 into bit 1 of the Control Register. To clear an interrupt write a 1 into bit 0 of the Control Register. All the bits operate independently. Be careful that the LNCEN bit is a 1 when clearing and/or enabling interrupts.

Status Register:
The single read only register (accessed with an INPUT or equivalent) has four bits as shown in Figure 1. Bit 3 says that power is applied to the board if it is a 0. If Bit 2 is a 1 the board has been enabled to operate via the LNCEN bit. Bit 1 is a 1 if interrupts are enabled and 0 if not enabled. If Bit 0 is a 1 the LNC has generated an interrupt (if enabled). If interrupts are not enabled, or if no interrupt level select switch has been closed, no interrupt will be generated, however, the INTR bit will still show interrupt (or completion) status. Bits 4-7 of the Status Register should be ignored.

INITIALIZE BLOCK

After power up, but before turning ON the LNCEN bit, the INITIALIZE BLOCK and CONTROL BLOCKS must be built in System memory. The format of the INITIALIZE BLOCK is shown in Figure 2, and the elements are described below. The INITIALIZE BLOCK is used only during the initialize procedures and may be overwritten in subsequent operations.

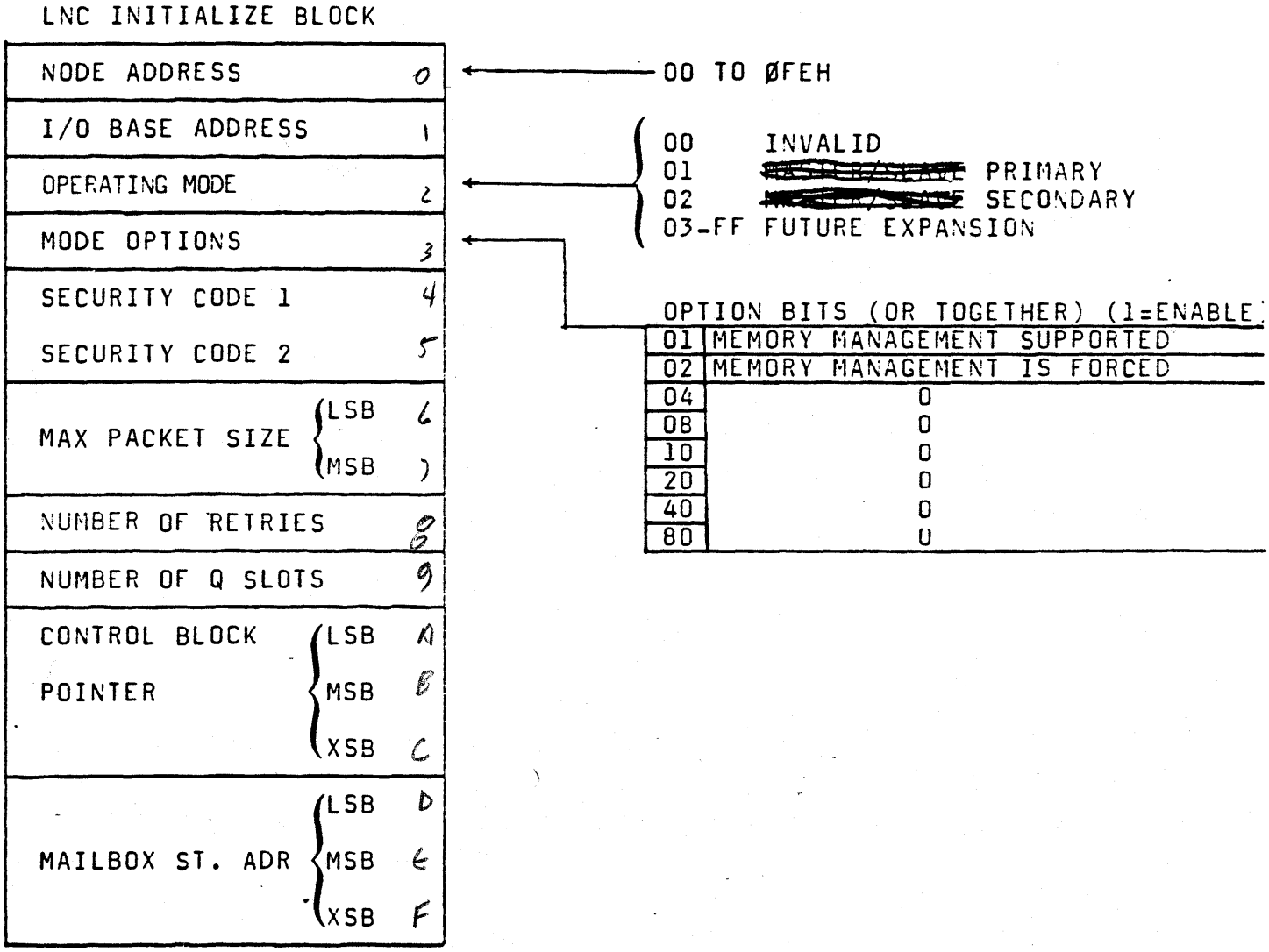


FIGURE 2

NODE ADDRESS:

The network address (or node address) of the LNC 5180 on the network. May be any number from 0 to 254 (0FEH). Be sure that no two LNCs on the network have the same node address. (1 Byte)

I/O BASE ADDRESS:

The Multibus I/O ADDRESS of the LNC 5180 Control and Status Register must match the address set into hardware DIP SWITCH 1. (1 Byte)

OPERATING MODE:

The operating mode of the LNC 5180:

00 = Invalid
01 = Primary on a Master/Slave network
02 = Secondary on a Master/Slave network
03-FF = Reserved for future modes

Only one LNC in a network may be initialized as a Primary (or master) at any one time. (1 Byte)

MODE OPTIONS:

A bit mapped set of options or variations of the operating node. All bits set independent options which may coexist.

Bit 0 = 0 Memory Management is not supported
Bit 0 = 1 Memory Management is supported
Bit 1 = 0 Memory Management is not required of all transactions
Bit 1 = 1 Memory Management is forced on all transactions

All other Bits are reserved for future options. (1 Byte)

SECURITY CODE 1
SECURITY CODE 2:

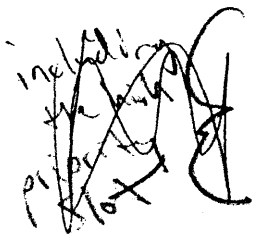
If both CODE 1 and CODE 2 bytes are 0 no security code is used. If either or both bytes are non-zero no incoming messages will be accepted unless the Security Code bytes traveling with the message match the initialized values. (2 Bytes)

MAX PACKET SIZE:

A number from 1 to 2048 that defines the maximum number of data bytes to be transferred in a data packet. Larger transactions are automatically broken into packets by the LNC 5180. A normal MAX PACKET SIZE is ~~about 256~~ 1024 bytes although this may vary based on the nature of the network, its loading characteristics, and normal bit error rate. Various control fields and transparent information fields are automatically appended to each packet by the LNC 5180. (2 Bytes)

NUMBER OF RETRIES: When the sending station does not receive an ACK (packet Acknowledgement), ~~When errors are detected in received data blocks the receiving station sends a NACK (negative Acknowledgement) message.~~ the sending station will automatically send the data again. This byte is the maximum number of retries before the sending station gives up and reports an error. A normal number is 3-5. (1 Byte)

NUMBER OF Q-SLOTS:



A number from 0-254 defining the number of transaction queue slots, ~~in addition to~~ including the high priority slot, to be provided. This determines the actual size of the CONTROL BLOCK. A value of 0 turns OFF the entire Queueing function. Most systems will need less than 16 slots. (1 Byte)

CONTROL BLOCK POINTER:

A 3 byte (24 bit) pointer to the first byte of the CONTROL BLOCK. (3 Bytes)

MAILBOX STARTING ADDRESS:

A 3 byte (24 bit) pointer to the first mailbox of the mailbox array which is 256 bytes long. If the field = 0 there is no mailbox array. (3 Bytes)

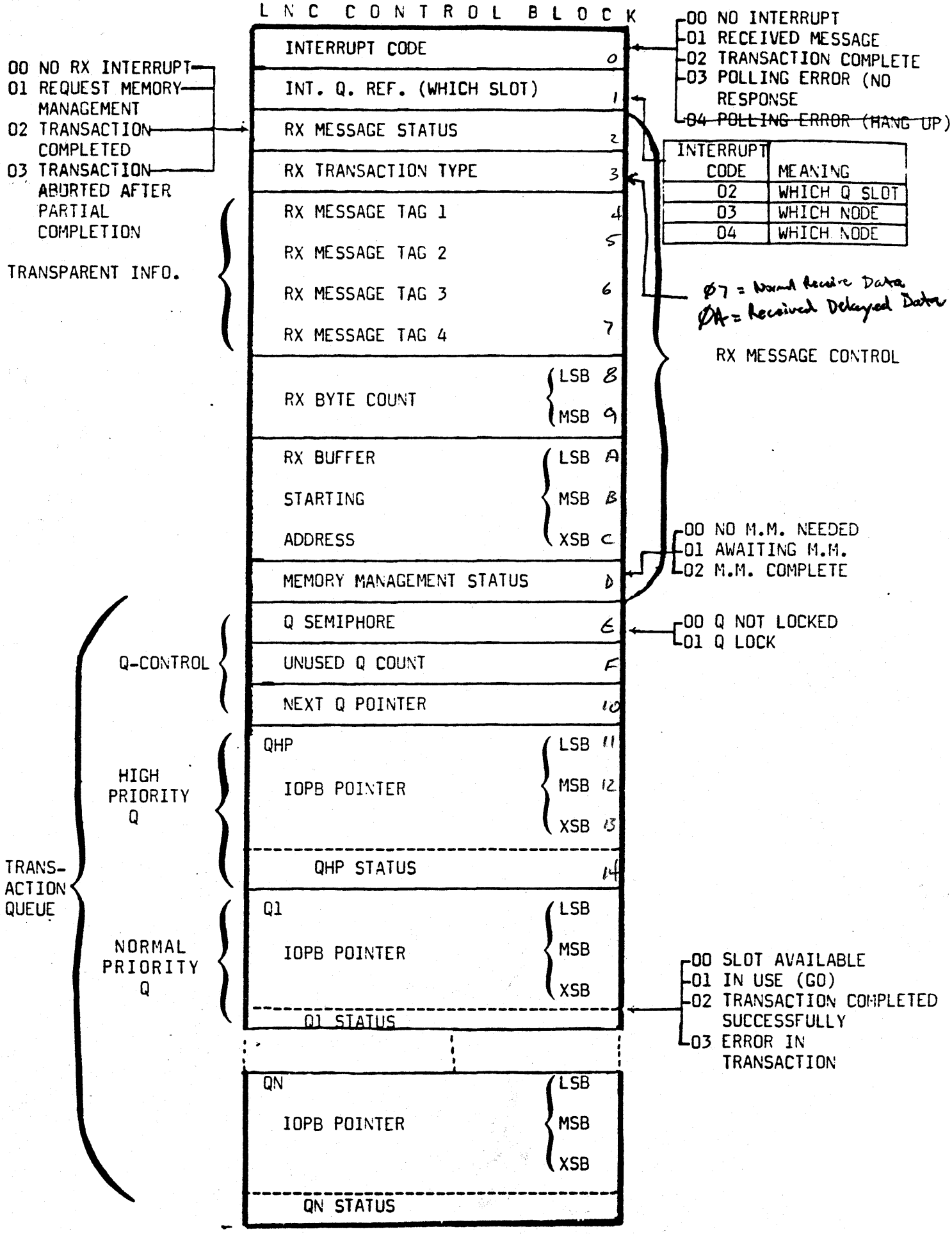

LNC CONTROL BLOCK
INTERRUPT CODE 0
INT. Q. REF. (WHICH SLOT) 1
RX MESSAGE STATUS 2
RX TRANSACTION TYPE 3
RX MESSAGE TAG 1 4
RX MESSAGE TAG 2 5
RX MESSAGE TAG 3 6
RX MESSAGE TAG 4 7
RX BYTE COUNT LSB 8 MSB 9
RX BUFFER STARTING ADDRESS LSB A MSB B XSB C
MEMORY MANAGEMENT STATUS D
Q SEMIPHORE E
UNUSED Q COUNT F
NEXT Q POINTER 10
QHP IOPB POINTER LSB 11 MSB 12 XSB 13
QHP STATUS 14
Q1 IOPB POINTER LSB MSB XSB
Q1 STATUS
QN IOPB POINTER LSB MSB XSB
QN STATUS
00 NO INTERRUPT
01 RECEIVED MESSAGE
02 TRANSACTION COMPLETE
03 POLLING ERROR (NO RESPONSE
04 POLLING ERROR (HANG UP)
INTERRUPT CODE | MEANING
02 | WHICH Q SLOT
03 | WHICH NODE
04 | WHICH NODE
00 NO RX INTERRUPT
01 REQUEST MEMORY MANAGEMENT
02 TRANSACTION COMPLETED
03 TRANSACTION ABORTED AFTER PARTIAL COMPLETION
TRANSPARENT INFO.
Ø7 = Normal Receive Data
ØA = Received Delayed Data
RX MESSAGE CONTROL
00 NO M.M. NEEDED
01 AWAITING M.M.
02 M.M. COMPLETE
00 Q NOT LOCKED
01 Q LOCK
Q-CONTROL
HIGH PRIORITY Q
NORMAL PRIORITY Q
TRANS-ACTION QUEUE
00 SLOT AVAILABLE
01 IN USE (GO)
02 TRANSACTION COMPLETED SUCCESSFULLY
03 ERROR IN TRANSACTION

FIGURE 3

CONTROL BLOCK

The Control Block is a memory resident control structure permanently located at initialization time. It is shown in Figure 3 and contains several subsections to handle interrupts, a Receive Message control Section and Transaction Queue. The elements of each section are described below:

INTERRUPT CODE:

When an interrupt is received on the selected level the handler should refer to this byte to get more information about the source of the interrupt. It is valid until the CPU sets it to 00, thereby allowing the next interrupt. {Set by LNC and Cleared by the CPU}

CODE = 00 Interrupt is cleared (set by CPU)
= 01 A message has been received
= 02 A transaction has been completed
= 03 There was a polling error (No Response)
~~= 04 A "polled" device refuses to relinquish control~~
= 04 A transmit error has occurred on a "Poll".

INTERRUPT REFERENCE:

The definition of this byte has multiple meanings based on the value of the INTERRUPT CODE.
If CODE = 02 This byte says which Q-Slot has a transaction completion.
IF CODE = 03 This byte says which polled deivce did not respond
~~IF CODE = 04 This byte says which device refuses to relinquish control.~~
IF CODE = ANY OTHER VALUE This byte has no meaning. {Set by LNC}

RX MESSAGE STATUS:

IF CODE = 01 This byte contains more information about the receive message interrupt and the action to be taken by the CPU. If this byte is:

01 = The received message is requesting Memory Management. Follow procedure described in a later section. {Set by LNC}
02 = A complete message has been received. {Set by LNC}
03 = A transaction has been aborted after partial completion. {Set by LNC}

RX TRANSACTION TYPE:

This byte identifies the nature of the received transaction or message and is simply the COMMAND CODE set by the sending device. {Set by LNC}

RX MESSAGE TAG 1, 2, 3, & 4:

These four bytes reflect the four bytes put into the IOPB by the originating station and are totally transparent to the LNC. They may be used in any way the User sees fit and may include information to be used by the interrupt handler or memory manager. They may also be used by the operating system to correlate incoming messages to requests sent out previously. {Set by LNC}

RX BYTE COUNT:

This 2 byte field contains the size of the data field of a received message already in memory. {Set by LNC}

RX BUFFER STARTING ADDRESS:

This 3 byte field contains the starting address of the data field of a received message already in memory. {Set by LNC and modifiable by CPU under Memory Management}

MEMORY MANAGEMENT STATUS:

Indicates the current status of the Memory Management function for this incoming transaction. If the contents are:

00 = No Memory Management is needed {Set by LNC}
01 = The LNC is waiting for the CPU to apply Memory Management to the RX Buffer Starting Address field {Set by LNC}
02 = The CPU has applied memory management and the LNC may proceed with DMA {Set by CPU}

Q-SEMIPHORE:

This byte is a software semiphore to allow multiple CPUs (or other MULTIBUS MASTERS) to have access to the transaction queue slots. Exact useage of this device is somewhat determined by the "Bus-Locking" characteristic of the CPU.

00 = Q is not "locked" and may be manipulated by the CPU
01 = Q is "locked" by another CPU and must not be manipulated. {Set by device attempting to queue up a transaction -- must be set back to 00 by that device after manipulation.}

UNUSED Q COUNT:

This byte is manipulated by the device queueing up a transaction. It should always reflect the number of normal priority Q-Slots that have not been assigned to far. If it ever gets to 0, the Queue is full. It should be decremented when a transaction is queued up and incremented when that Q-Slot is freed up after completion of the transaction. This byte is not used if the queueing feature ~~(normal psiority)~~ is not used. The High Priority Q-Slot operates independently of this byte.

NEXT Q POINTER:

This byte identifies the next available normal priority Q-Slot and is manipulated by the device queueing up a transaction for use by the next device. This procedure is explained more completely in later sections. If the UNUSED Q COUNT = 0, this byte is, of course, not valid.

Q-HP (HIGH PRIORITY):

This 4 byte field is the "High Priority Q-Slot" and exists whether or not any normal priority Q-Slots exist. If 1 or more normal Q-Slots do exist, this slot will be satisfied before any of the normal priority Q-Slots, except for the one currently being worked on.

The first 3 bytes point to the IOPB of the queued transaction. The fourth byte is the Q-Status byte. It can have the following values:

00 =	This slot is available	{Set by CPU}
01 =	This slot is in use (GO)	{Set by CPU}
02 =	The Transaction was completed successfully	{Set by LNC}
03 =	An unrecoverable error was encountered.	{Set by LNC}

Q-N (NORMAL Q SLOTS):

From 0 to (N-1) normal priority 4 byte Q-Slots with the same format as the Q-HP slot can be supported. All of these slots have equal priority and the LNC services them in a round robin fashion. The value of N is set at Initialization time and can be from 1 to 255.

NOTE: The LNC does not automatically free up Q-Slots upon completion of the transaction. It is up to the CPU to write a 00 to the Q-STATUS byte in the Slot to make it available for subsequent use.

IOPB

The I/O parameter block is the structure that tells the LNC what function is to be performed. It must be built in LNC accessible memory and pointed to by the Q-Slot in the CONTROL BLOCK. Figure 4 shows the format of the IOPB.

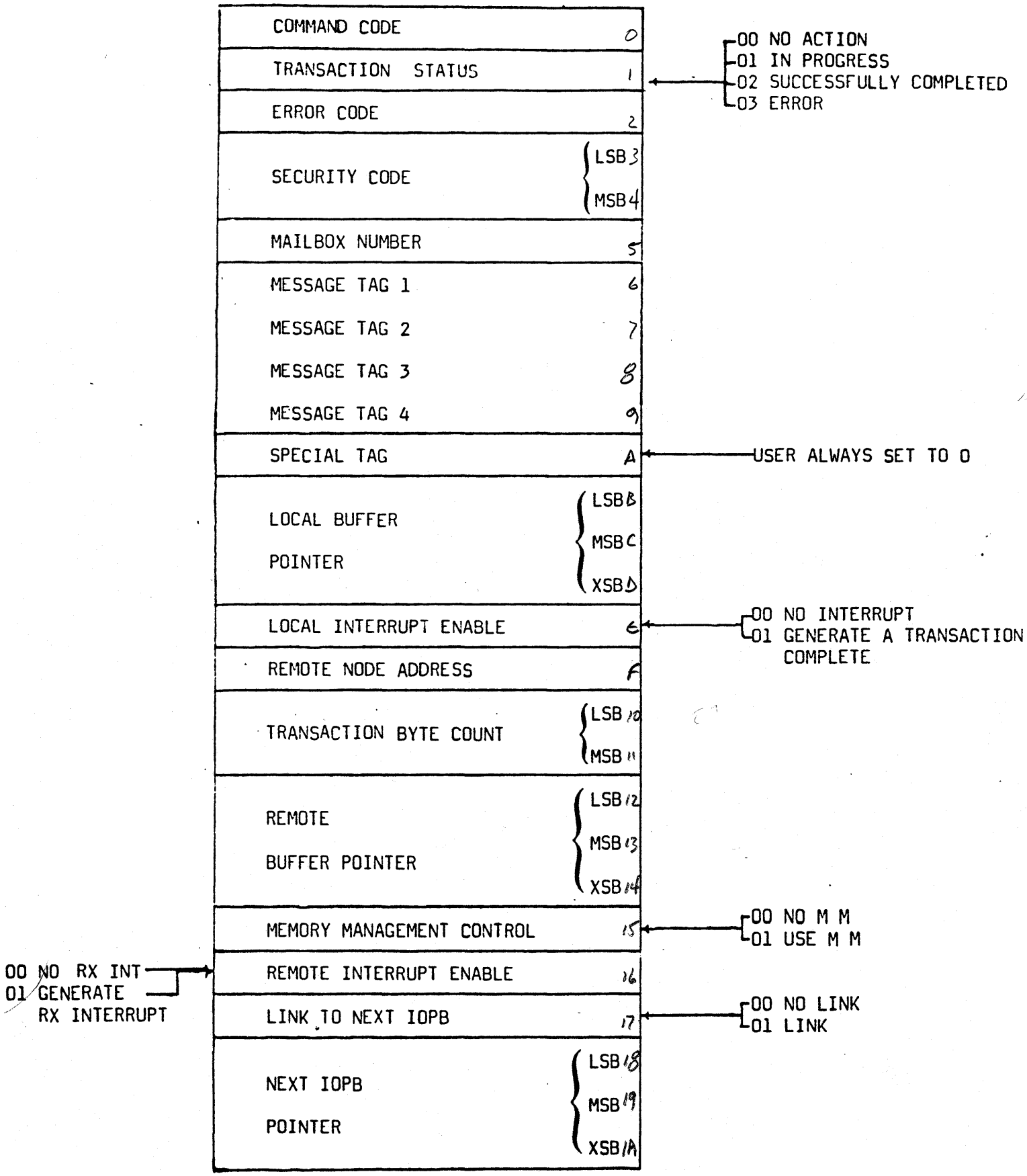


FIGURE 4

COMMAND CODE:

This byte contains the Command Code of the transaction. A table of command codes can be found in Figure 5. For example, a SEND DATA command will have a 80H in this byte.

TRANSACTION STATUS:

This byte is initialized to 0 by the CPU and is subsequently updated by the LNC to reflect the current status of the transaction.

00 = No Action has been taken.
01 = Transaction in Progress
02 = Transaction was successfully completed
03 = Error

ERROR CODE:

If the TRANSACTION STATUS byte indicates an error, this byte indicates the nature of the error. A table of error codes is found in Figure 6. If TRANSACTION STATUS indicates a successful transaction this byte contains the number of retries required to complete it.

SECURITY CODE:

This 2 byte field must match the security code of the receiving station for a SEND DATA transaction to be completed. This optional feature can be disabled by the receiving station by setting the code to 0 during initialization.

MAILBOX NUMBER:

This one byte field is the number of the mailbox in the receiving station that is to be set to FFH when the message is received.

MESSAGE TAG 1, 2, 3, & 4:

This 4 byte field is totally transparent to the LNC and is sent along with the message to the receiving station. It is then put into the receiving station RX Message Control section of the CONTROL BLOCK for use as described by the CPU software.

SPECIAL TAG:

This byte is for internal use by the LNC only. The User should set it to 0.

LOCAL BUFFER POINTER:

This 3 byte field points to the beginning memory address of the local data to be sent in a SEND transaction or the local receive buffer in a REQUEST transaction.

LOCAL INTERRUPT ENABLE:

If this one byte field is:
00 = Generate NO local interrupt on completion
01 = Generate a local interrupt on completion

REMOTE NODE ADDRESS:

The network or node address of the remote end of the transaction. Can be from 00 to 254. An address of 255 (FFH) addresses all attached devices and is only used by the LNC in special cases.

TRANSACTION BYTE COUNT:

This 2 byte field specifies the size of the data block to be moved.

REMOTE BUFFER POINTER:

This 3 byte field points to the beginning memory address in the remote device, of the data to be received and stored in a SEND transaction, or to be fetched and transmitted in a REQUEST transaction.

MEMORY MANAGMENT CONTROL:

This byte controls the usage of MEMORY MANAGEMENT on the remote end.
00 = Use no Memory Management
01 = Request Memory Management

REMOTE INTERRUPT ENABLE:

Determines whether or not the remote end is to be interrupted on receipt of a message.
00 = No interrupt
01 = Generate an interrupt

LINK TO NEXT IOPB:

If this byte is 01, the IOPB pointed to by the next field is to be automatically appended to this transaction. No completion interrupt is to be generated at this time. If this byte is 00, the transaction is over when the current operation is complete.

NEXT IOPB POINTER:

This 3 byte field points to the next IOPB to be considered part of this transaction.

LNC 5180 COMMANDS

Figure 5 is a table of all commands that the LNC 5180 currently supports. The command code is to be written into the first byte of the IOPB. Figure 6 is a table of Error Codes that may be returned in the Error Code Byte of the IOPB upon failure of a given command. Following is a description of the operation of each command. Any options that may not be supported by the transaction are delineated.

L N C C O M M A N D C O D E S

HEX CODE	COMMAND
00	NO OP
01	REINITIALIZE
02	LOCAL SELF TEST
03	TURN ON POLLING
04	TURN OFF POLLING
05	SET POLLING CONTROL TABLE
06	SEND POLL (MANUAL POLLING)
07	SEND DATA
08	SEND NO OP
09	REQUEST IMMEDIATE DATA
0A	REQUEST DELAYED DATA
0B	REQUEST LOCAL ~~STATUS~~ Statistics
0C	REQUEST REMOTE ~~STATUS~~ Statistics

Delete

F I G U R E 5

E R R O R C O D E S

CODE(#)	ERROR TYPE
01	INVALID PARAMETER IN IOPB
02	NO RESPONSE FROM DESTINATION (TIMEOUT)
03	TOO MANY DATA RETRIES
04	DID NOT RECEIVE REQUESTED IMMEDIATE DATA/STATUS
10	LOCAL BUS TIMEOUT
~~11~~ 0B	CANNOT READ BUS DATA
~~12~~ 0C	CANNOT WRITE BUS DATA
20	REMOTE Q WAS FULL ON REQUEST DELAYED DATA
21	TRANSACTION REJECTED BY DESTINATION
	FUTURE ERROR CODES MAY BE ADDED

F I G U R E 6

17 Security Error Detected on Received Packet

NO OPERATION (00)

This command can be used to check basic operation of the LNC since it will show completion status in the IOPB and CONTROL BLOCK and will cause an interrupt if so instructed in the IOPB.

SEND DATA (~~40~~ 07H)

Send a single contiguous block of data starting at the specified location in local system memory to the specified memory locations in the specified remote node. All IOPB options are supported.

REQUEST IMMEDIATE DATA (~~81H~~ 09H)

Read the contents of a block of memory in a remote node as a part of this transaction, i.e., wait for the response before ending the transaction. Does not interrupt the remote device unless Memory Management is applied. No mailboxes are effected. The Security Code features does apply if used. This transaction does incur some network overhead while waiting for the response to be fetched and formatted.

REQUEST DELAYED DATA (~~82H~~ 0AH)

Request that a remote device send a specified block of data to this station as soon as it can. Causes a transaction to be put into the internal queue of the remote station LNC. The specified MESSAGE TAG(s) will be returned with the data and presented upon receipt of the return message, The RX TRANSACTION TYPE, upon receipt of the return message, will be a ~~82H~~ 0AH The Mailbox Number refers to the one to be set upon receipt of the return message. Memory Management refers to the remote device only (the Local Buffer pointer is an absolute value). The remote device is not interrupted (except for possible Memory Management) or notified of the operation, and no remote Mailbox is set. The Security Code feature in the remote device, however, does apply. This transaction reports completion (interrupts) as soon as the remote device accepts the queued transaction. The later receipt of the returned data appears to be a separate operation except as stated above. It is suggested that the MESSAGE TAGS (1, 2, 3, 4,) will be used to correlate the return with the request. If the returned data operation is to set a local interrupt on receipt (as would be expected) set the "REMOTE INTERRUPT ENABLE" byte in the IOPB to an 01.

REQUEST LOCAL ~~STATUS~~ Statistics (~~83H~~ 0BH)

The LNC 5180 keeps various operation status and statistics in an internal table of the format shown in Figure 8. This command sends the contents of this table into the local memory buffer pointed to in the IOPB. The user may chose to see only a part of the Status table by setting a TRANSACTION BYTE COUNT to less than the total table size, but it always

starts at the first byte of the table. A local interrupt may be set on completion but no other options apply. except Linked IOPBs.

Insert → SEE END OF APPENDIX A

REQUEST REMOTE ~~STATUS~~ Statistics (~~84~~ 0C H)

Same as REQUEST LOCAL STATUS except that the Status Table in the specified remote device is reported. Operates in the same fashion as the REQUEST IMMEDIATE DATA transaction.

SEND NO-OP (~~80~~ 08 H)

Send a NO-OP message to the specified remote devices LNC. This is basically a "Sanity Check" on the remote device. No interrupt or any other operation is performed on the remote except for the LNC's acknowledgement of receipt of the message. The local CPU is interrupted (if enabled) just like any other transaction.

REINITIALIZE (01)

This command sends the LNC back through the initialize sequence without requiring manipulation of the Multibus Control Register bits. The statistical information and polling control tables are not cleared as they would be with the LNCEN manipulation. The local buffer pointer must point to the new Initialization Block. The transaction is completed like any other transaction, except for the possible relocation of the CONTROL BLOCK.

LOCAL SELF TEST (02)

The LNC performs a series of local self tests and reports the results into local memory much like the REQUEST LOCAL STATUS COMMAND. The format of the report is found in Figure 10. The maximum report length can be set by the TRANSACTION BYTE COUNT.

Insert → SEE END OF APPENDIX A

TURN ON POLLING (03)

Turns on the automatic polling features (only in MASTER STATIONS). A SET POLLING CONTROL TABLE command must have been successfully completed before this command can be accepted.

TURN OFF POLLING (04)

Turns off the automatic polling features. Does not effect the POLLING CONTROL TABLE.

SET POLLING CONTROL TABLE (05)

This command sets the poll slot time duration and order of polling using a format shown in Figure 7. The local Buffer Pointer must point to the Polling Control Table. The transaction Byte Count must be equal to the total number of bytes in the Table or the command will be rejected as a special validity check on this sensitive procedure. Note that you can vary the frequency of polling for certain node addresses by simply repeating their number more often than other devices. This effectively allows some nodes to have higher priority than others. Note that even if automatic polling is not used, for first part of this table must be set up to allow for Manual Poll Command to be used. This command does not apply to Secondary (Slave) stations.

SEND POLL COMMAND (06)

Send a Manual Poll to the specified node. This allows the specified node to transact business for up to a maximum time period as determined by the Polling Control Table which must have preceded this command. Can be used instead of or in combination with automatic polling.

INTERRUPT HANDLING

Although some simple applications can operate strictly under status in the Control Block and the IOPB, most systems will use an interrupt handler to respond to events as they happen. Recall that interrupts may be enabled, disabled, and cleared using the write only Multibus Control Register, and that the Multibus interrupt level is set via dip switches on the LNC. Also, the status of Interrupt enable and the INTR bit that verifies that the LNC is generating an interrupt condition (whether or not the interrupt is enabled) are available in the MULTIBUS STATUS REGISTER as shown in Figure 1.

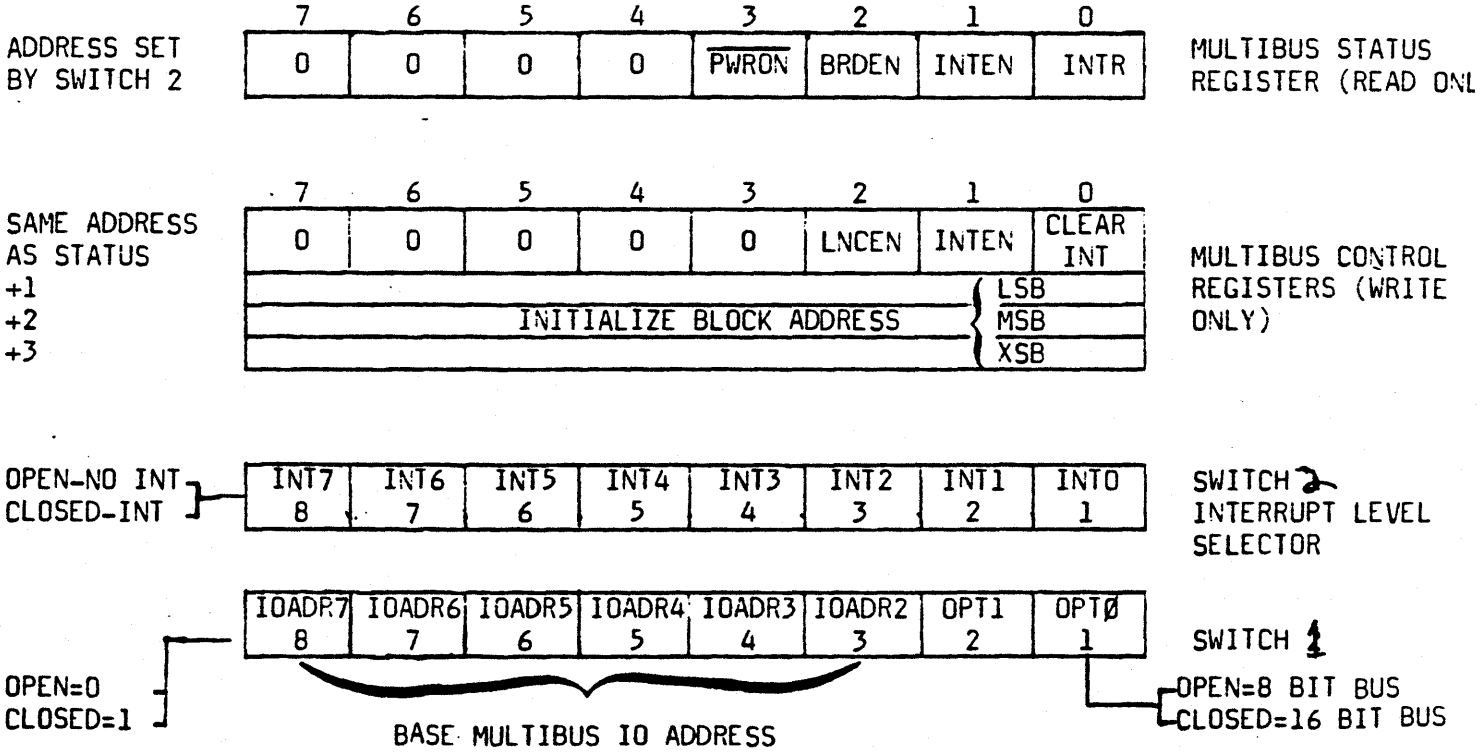


FIGURE 1

Four general sources of interrupt exist:

1. A transaction is completed
2. An incoming message is received
3. The Memory Management function is requested on an incoming message
4. There was a polling error ~~(2 types)~~

To determine the source of a given interrupt read the INTERRUPT CODE byte of the CONTROL BLOCK. Appropriate action should then be taken to handle the interrupt. After that action is taken, the INTERRUPT CODE must be cleared to 00 before the LNC can present another interrupt.

* * * * * * * * * * * * * * * *

SPECIAL NOTE: CLEAR THE HARDWARE INTERRUPT (using the Multibus Control Register) BEFORE THE INTERRUPT CODE IS SET TO 00, or you may miss the next hardware interrupt (and its INTR status bit). This is true so long as you are using interrupts and/or the INTR hardware register bit at all. If you are running strictly under Status and no interrupt level is selected (with on board dip switches) you do not have to clear the hardware interrupt at all, but still must clear the INTERRUPT CODE byte.

* * * * * * * * * * * * * * * *

Refer to Figure 3, the Control Block format, while reading the following sections.

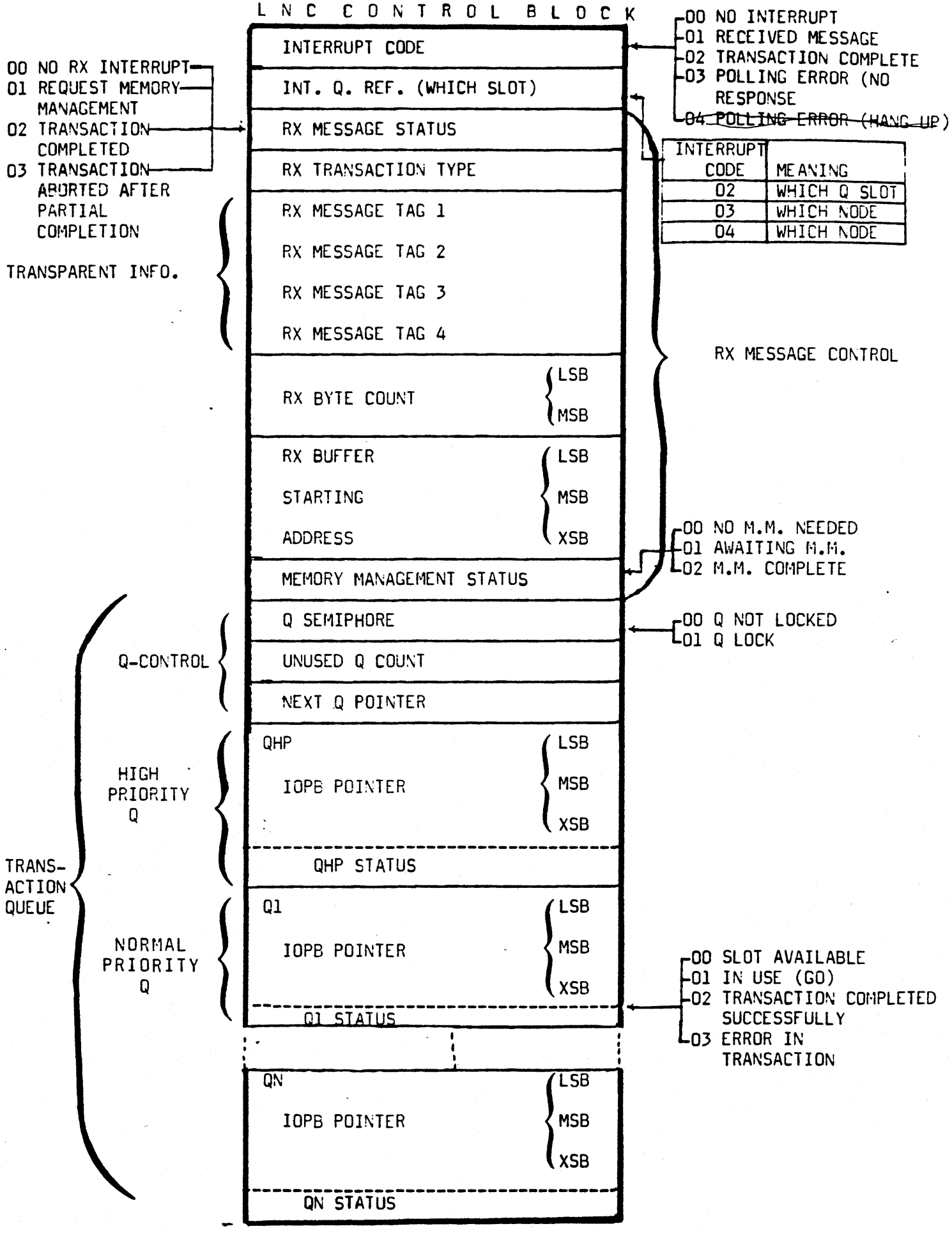


FIGURE 3

LNC-32

TRANSACTION COMPLETE INTERRUPT

If the INTERRUPT CODE byte of the CONTROL BLOCK is a 02 the interrupt is a result of the completion of a queued transaction. The INTERRUPT Q REFERENCE byte refers to the Q-Slot that was satisfied. The High Priority Q-Slot is referred to as 1. The Q-Slot points to the IOPB of the transaction and provides a Q-Status byte which should, at this time, be either 02 (successfully completed) or 03 (error in transaction), and the IOPB carries even further information. The interrupt handler at this time may either clear the Q-Slot by writing a 00 to the Q-Slot status byte and incrementing the Unused Q-Count byte of the Q-Control section, or possibly refer that operation to the Operating Sytem or other software. In any case, the interrupt handler must set the INTERRUPT CODE to 00 to allow the next interrupt to be set by the LNC. Remember to Clear the Hardware Interrupt (so long as interrupts are being used) before setting the INTERRUPT CODE to 00.

see LNC-21 what is range of normal priority Q-slots?

RECEIVED MESSAGE INTERRUPT

After a completed message has been received and the resulting data is in memory ~~(and if the sending device said to generate a Receive Interrupt)~~, the LNC sets the RX MESSAGE STATUS byte to 02, sets the RX TRANSACTION TYPE to the appropriate value, puts the Message Tags sent with the Message into RX MESSAGE TAG 1, 2, 3, & 4, enters the size of the data field into RX BYTE COUNT, sets the RX BUFFER STARTING ADDRESS, and then sets the INTERRUPT CODE to 01, ~~and sets the hardware interrupt~~. If the sending device said to generate a receive interrupt, then the hardware interrupt will be set.

~~If the received message has the "REMOTE INTERRUPT ENABLE" turned Off none of the above happens but the data is put into memory. No response is required.~~

If a second receive message comes in before the previous interrupt is handled, ~~it will wait in internal buffers (so long as buffer space is available)~~ and it will be reported as

soon as the INTERRUPT CODE is cleared. Only one such incoming message can be "stacked up" in this fashion. Any further attempts to send messages to this station will ~~be responded to with a "RECEIVE BUFFER FULL" error.~~ result in a "No Response" error.

One further action, the setting of the Mailbox to FFH, also takes place so long as the receiving LNC is initialized to have a Mailbox array. This is true so long as a complete message has been successfully received whether or not REMOTE INTERRUPT ENABLE was specified by the sending device. ~~(If REMOTE INTERRUPT ENABLE was set to 00 by the sending device the only indication to the receiving device is seen in the Mailbox).~~

Note that all of the above refers only to SEND DATA Messages sent to the receiving device and does not apply to DATA REQUESTS (IMMEDIATE OR DELAYED) from another network device, or LNC control messages that may be sent as part of the transparent network protocol. In those cases the receiving device CPU is completely unaware of the transaction.

From the User's point of view the only thing that the Interrupt Handler must do with RECEIVE MESSAGE Interrupts (with RX MESSAGE STATUS = 02) is recognize and respond to the information in the RX MESSAGE CONTROL section of the CONTROL BLOCK, clear the hardware Interrupt (if used) and write a 00 to the INTERRUPT CODE to allow the LNC to proceed with the next interrupt.

MEMORY MANAGEMENT INTERRUPT

If the LNC has been initialized with Memory Management enabled (or "forced") and a message is received that specifies a request for Memory Management (or always in case of the "forced" option), the successful receipt of the first packet of possibly a Multipacket transaction

will cause a Received Message Interrupt (INTERRUPT CODE = 01) but with RX MESSAGE STATUS = 01 (Request Memory Management). In addition the RX TRANSACTION TYPE, RX MESSAGE TAGS, RX BYTE COUNT (for the entire transaction to be performed), and the RX BUFFER STARTING ADDRESS (set by the sending device) will be set up. Also the MEMORY MANAGEMENT STATUS byte will be set to 01 (Awaiting Memory Management).

At this point the interrupt handler, or the Operating System, should modify the RX BUFFER STARTING ADDRESS field as desired, write a 02 (Memory Management Complete) into the MEMORY MANAGEMENT STATUS byte, and write 00 (clear) to the INTERRUPT CODE. The LNC will proceed with the transaction and provide a normal Receive Message Interrupt upon completion.

The order of setting MEMORY MANAGEMENT STATUS = 02 and setting INTERRUPT CODE = 00 is irrelevant since the LNC will proceed as soon as it sees the 02 in MEMORY MANAGEMENT STATUS and will not set a new interrupt until the previous is cleared (INTERRUPT CODE = 00). This allows the interrupt handler to normally clear the hardware interrupt and set the INTERRUPT CODE = 00 and pass a call to the Operating System to perform the Memory Management function. The function should, however, be performed fairly expeditiously since the LNC may be receiving further packets of data and may need to empty its buffers~~, (if the transaction is larger than the buffer space) before the transaction can be~~ completed. If this happens the transactions can be successfully completed but the entire network will effectively be waiting for the transaction to be completed.

OK
No Change Needed

MESSAGE ABORTION INTERRUPT

A special case of the RECEIVE MESSAGE INTERRUPT is seen if a message transaction is aborted after Memory Management has

This page OK

been performed but before the transaction is complete. In this case INTERRUPT CODE = 01 but RX MESSAGE STATUS = 03. The reporting of this special condition allows the receiving station software to deallocate the assigned buffer space and cancel the transaction. The interrupt handler should simply clear the interrupt and set INTERRUPT CODE = 00.

Another special case of partial message completion is seen if the transaction is larger than the internal buffer space whether or not Memory Management is applied. (INTERRUPT CODE = 01, RX MESSAGE STATUS = 03, and MEMORY MANAGEMENT STATUS = 00 or 02). In this case the earliest packets have already been moved into Multibus memory starting at the RX BUFFER STARTING ADDRESS but for some reason the transaction aborted before the full message (Size = RX BYTE COUNT) was received. How to handle this anomalous event is strictly determined by the application and the nature of the data. ~~The situation is avoided, when possible, by the practice of receiving and verifying the entire message (possibly multiple packets), up to the available receive buffer size, before putting any of it into system memory.~~

POLLING ERROR INTERRUPTS

When the LNC 5180 is initialized to be a MASTER and Automatic Polling is turned ON there are two more potential causes of interrupts. If INTERRUPT CODE = 03 a polled device, specified in the INTERRUPT Q REFERENCE byte, has failed to respond to the last three attempts to poll him. The source of the problem should normally be investigated and the device should probably be deleted from the POLLING TABLE using the SET POLLING CONTROL TABLE command.

If INTERRUPT CODE = 04, a polled device specified in the INTERRUPT Q REFERENCE byte, has failed to relinquish control of the network within the specified time period. The problem could be either in the polled device, or in the device he is attempting to transact with. Since all

devices have logic to defeat such a condition when within their control it is probably that the polled device is the problem.

These types of interrupts commonly requires some type of user interaction. The interrupt handler should clear the hardware interrupts, set INTERRUPT CODE = 00 and refer the situation to a higher level of software.

Q U E U E H A N D L I N G

To cause a transaction to take place the User builds an IOPB and then "Queues it up". The CONTROL BLOCK (See Figure 3) has a section of it called the TRANSACTION QUEUE which consists of a 3 byte Q-Control subsection, a 4 byte HIGH PRIORITY Q, and up to (N-1) normal priority Q's when (N-1) is from 1-255 and is set at initialization time. Each Q slot, High or Normal priority, consists of a 3 byte pointer and a single Q-Status byte.

To "QUEUE UP" a transaction in any of the queues one must first verify that the Q-slot is available (Q-STATUS=00). If so, write the pointer to the IOPB into the first three bytes of the slot and a 01 into the Q-Status. If you have selected the High Priority Q-Slot or if that is the only Q-Slot (N=1) you have completed the "QUEUE UP" operation. If N ≠ 0 or 1 and you are using a normal priority Q-Slot you will normally also manipulate the Q-Control section (UNUSED Q-COUNT and NEXT Q POINTER) as suggested below to complete the "QUEUE UP" operation. The use of the Q-Control section is optional since the LNC actually uses none of the bytes directly, but the suggested conventions have been seen to work well in real systems. In addition, use of them as described will insure the ability to use higher level functionality that may be available on this product and other Interphase LNC products in the future, some of which may include self queueing of transactions by the LNC.

Before describing the queueing software conventions it is instructive to understand how the LNC uses the queue.

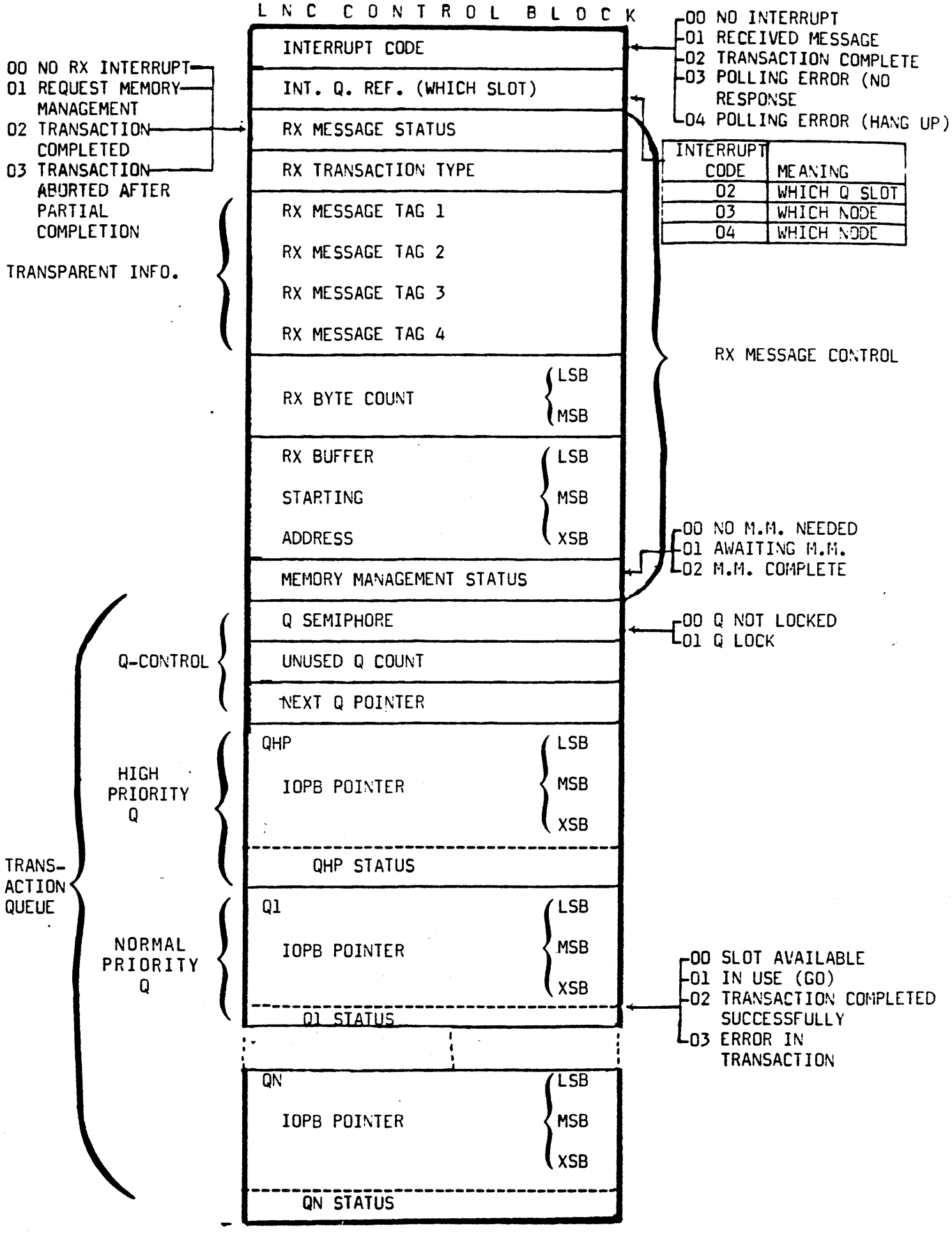

LNC CONTROL BLOCK
INTERRUPT CODE
INT. Q. REF. (WHICH SLOT)
RX MESSAGE STATUS
RX TRANSACTION TYPE
RX MESSAGE TAG 1
RX MESSAGE TAG 2
RX MESSAGE TAG 3
RX MESSAGE TAG 4
RX BYTE COUNT
LSB
MSB
RX BUFFER STARTING ADDRESS
LSB
MSB
XSB
MEMORY MANAGEMENT STATUS
Q SEMIPHORE
UNUSED Q COUNT
NEXT Q POINTER
QHP IOPB POINTER
QHP STATUS
Q1 IOPB POINTER
Q1 STATUS
QN IOPB POINTER
QN STATUS
00 NO INTERRUPT
01 RECEIVED MESSAGE
02 TRANSACTION COMPLETE
03 POLLING ERROR (NO RESPONSE
04 POLLING ERROR (HANG UP)
INTERRUPT CODE
MEANING
02 WHICH Q SLOT
03 WHICH NODE
04 WHICH NODE
00 NO RX INTERRUPT
01 REQUEST MEMORY MANAGEMENT
02 TRANSACTION COMPLETED
03 TRANSACTION ABORTED AFTER PARTIAL COMPLETION
TRANSPARENT INFO.
RX MESSAGE CONTROL
00 NO M.M. NEEDED
01 AWAITING M.M.
02 M.M. COMPLETE
00 Q NOT LOCKED
01 Q LOCK
Q-CONTROL
HIGH PRIORITY Q
NORMAL PRIORITY Q
TRANS-ACTION QUEUE
00 SLOT AVAILABLE
01 IN USE (GO)
02 TRANSACTION COMPLETED SUCCESSFULLY
03 ERROR IN TRANSACTION

FIGURE 3

LNC USEAGE OF THE TRANSACTION QUEUE

The LNC determines that it has something to do by looking at the Q-STATUS byte of the Q-SLOTS. After initialization it first looks at the High Priority Q-Slot (QHP STATUS) and, if a transaction is queued up (Q-STATUS=01), it starts the transaction. If the Q_{HP} Status is not 01, the LNC looks at the Q1 STATUS (if it exists) and starts it if Q1-STATUS=01. If not it goes back to QHP-STATUS. It will continue alternating between QHP-STATUS and Q1-STATUS until Q1-STATUS has work to do. After completing that work it will start alternating between QHP-STATUS and Q2-STATUS until Q2-STATUS has work to do. It will go down through the Q-SLOTS until QN-STATUS has work to do. After completing that work it will again alternate between QHP-STATUS and Q1-STATS, and the "round robin" starts over. The LNC will not look at Q2 STATUS until Q1 STATUS is satisfied. This insures that transactions are performed in the order in which they are queued up except for the high priority queue which is always looked at every other time.

Note that the Q-SLOT has new work to do only when Q-STATUS = 01. If QSTATUS=02 or 03 the old transaction has been completed but the completion has not been acknoweldged (or recognized) by the CPU software. If QSTATUS=00 the old transaction was cleared and a new one may be queued up at any instant. Based on this logic the LNC modifies how long it waits between looks (to that slot). If QSTATUS was 00 it waits only about 50 µsec. If QSTATUS was 02 or 03 it will wait about 250 µsec before it looks again in order to reduce the overall bus overhead.

Q-SEMIPHOR

The Q-SEMIPHOR (See Figure 3) is an optional feature that should be used when more than one CPU (or other BUS MASTER) can queue up transactions for a single LNC and the CPUs don't have the ability to "Lock the bus" while they manipulate

the QUEUE. Some applications where interruptable and re-entrant routines do the manipulation you may use the Q-SEMIPHOR even if you can "Lock the bus". When used at all, the Q-SEMIPHORE should be used both when using the High Priority Q-Slot and when using a normal priority Q-SLOT. THE Q-SEMIPHOR IS NOT ACUTALLY USED BY THE LNC BUT IS MERELY A SOFTWARE USEAGE CONVENTION.

Simply stated, to use the TRANSACTION QUEUE the User should see if the Q-SEMIPHORE = 0, and, if so, set it to 01, do the needed manipulation, then set it back to 00. If it is already set to 01, the User must wait until it has been cleared.

UNUSED Q COUNT

This byte should originally be initialzed by the CPU to be equal to (N-1). It refers only to the number of unused normal priority Q-Slots. When a User queues up a transaction in a normal priority Q-Slot he must first see that the UNUSED Q COUNT is NON ZERO, do the operation described above, decrement the UNUSED Q COUNT, and finally manipulate the NEXT Q POINTER as described below. THE UNUSED Q COUNT IS NOT ACTUALLY USED BY THE LNC BUT IS MERELY A SOFTWARE USEAGE CONVENTION.

Recall that the interrupt handler, or other software that handles acknowledgement of completion of a transaction will have the job of incrementing the UNUSED Q COUNT and clearing the Q-STATUS.

NEXT Q POINTER

This byte should always point to the next available Q-SLOT and is for the convenience of the queueing software. To finish the queue-up procedure for normal priority Q-Slots (NOT HIGH PRIORITY) increment the next Q POINTER by 1 unless it is already (N-1), at which time set it to 1. It

is still, of course, necessary for the queueing software to verify that the slot is truely available before reusing it. THIS BYTE IS NOT ACTUALLY USED BY THE LNC BUT IS MERELY A SOFTWARE USEAGE CONVENTION.

AUTOMATIC POLLING

In a MASTER/SLAVE system, one device, the MASTER (or PRIMARY Station) is responsible for servicing the needs of the Slaves (or SECONDARY Station) such that information is transferred between network devices. Some MASTER/SLAVE networks force all data to be passed through the MASTER. In those cases for information to pass between SLAVES the sending SLAVE must send to the MASTER and the MASTER sends to the receiving SLAVE

Although one can use the LNC in this fashion it is not necessary to do so. Any device can talk to any other device directly. The MASTER simply controls who may start a transaction by sending a "poll" to that device, who then can start transactions for up to a specified period of time.

The MASTER knows who to poll and what the duration of the time slot is by being sent a "POLLING CONTROL TABLE" using the SET POLLING CONTROL TABLE command. For that command the LOCAL BUFFER POINTER (IOPB) must point to the beginning of the Polling Control Table of the form shown in Figure 7. The TRANSACTION BYTE COUNT (IOPB) must be exactly the size of the entire table. This value is checked for validity against the "NUMBER OF SLOTS" specified in the table due to the special sensitivity to errors in this operation.

Automatic POLLING can be turned ON and OFF with normal LNC commands (03 and 04 respectively). On power up Polling is turned OFF. Polling is also turned OFF every time the POLLING CONTROL TABLE is set. Before turning polling ON the POLLING CONTROL TABLE must be set.

When a station is polled it can start transactions so long as the elapsed time from the beginning of the slot is less than the specified TIME SLOT DURATION. If the TIME SLOT DURATION = 0 the Slave can start only 1 transaction, or more accurately, only one Q-SLOT can be worked on. The transaction can, however, be a linked transaction. If a station has no work to do or finishes its work before the slot expires it will immediately send a control message to the MASTER, relinquishing the slot.

Network Node Addresses can be entered in any order and duplicated as many times as desired. This allows you to poll given network devices more often than others based on expected loading factors. The table can be changed dynamically to allow adding and deleting devices on the network.

If a polled device fails to answer the poll more than three times in a row an interrupt is generated as described in an earlier section. ~~If a polled device fails to relinquish the network, the MASTER generates an interrupt as described earlier and automatically attempts to unjam the network by sending repeated control messages. The LNC will not attempt further data type transactions until the network is unjammed, but it will respond to localized commands, such as REQUEST LOCAL STATUS, so long as they are queued in the HIGH PRIORITY Q-SLOT.~~

If automatic polling is not used, but Manual Polling is done via the SEND POLL command, the SET POLLING TABLE command must still be performed, ~~if a SLOT DURATION of greater than 0 is desired.~~

PHYSICAL LAYER INTERFACE

As described in early parts of this manual, the LNC 5180 encompasses ISO Model layers 2-5 and, by design, does not include the "PHYSICAL LAYER" device. The PHYSICAL LAYER device, or MEDIA ACCESS UNIT (MAU), is what physically attaches to the network media (broadband or baseband coax, microwave, optical fibers, etc.) and provides whatever modulation of signals, data and clock mixing, and other signal conditioning that there is to be done. Media Access Units can include baseband or broadband modulators/ demodulators, standard high speed modems operating on leased lines (commonly at 56 Kb/s), T1 carrier access units (commonly operating at 1.544 Mb/s), or simple signal line drivers and receivers. MAUs can be supplied by Interphase, purchased from other manufacturers, or manufactured by the OEM. The needed characteristics vary widely based on the length of the network, data rate (set by the MAU), number of devices, etc.

Interphase manufactures a Multibus card mounted FSK modulated MAU that directly cables (via a flat cable) to the LNC, an adapter to Bell DDS Service, an adapter to T1 carrier access units, and various other accessories described in other literature. For further advice in this area contact Interphase.

The standard Physical Layer interface satisfies the EIA Standard RS422, the Foreward of which can be found in Appendix B. Figure 9 shows the definition of the signals on the 40 pin connector on the LNC 5180.

The LNC 5180 implements 9 category I circuits as classified by the RS449 interface specification. RS449 specifies that for applications where the signaling rate on the data interchange circuits is above 20,000 bits per second, all category I circuits shall use the balanced electrical characteristics of RS422. The maximum signaling rate on these balanced lines is 2 megabits/sec.

The following Category I circuits are implemented by the LNC:

CIRCUIT SD (SEND DATA)
The data signals originated by the LNC to be transmitted to the remote LNC's is transferred on this circuit to the modem.

CIRCUIT RD (RECEIVE DATA)
The data signals generated by the modem, in response to data channel line signals received from a remote LNC, are transferred on this circuit to the LNC.

CIRCUIT ST (SEND TIMING)
Signals on this circuit provide the LNC with transmit signal element timing information.

CIRCUIT RT (RECEIVE TIMING)
Signals on this circuit provide the LNC with receive signal element timing.

CIRCUIT RS (REQUEST TO SEND)
Signals on this circuit control the data channel transmit function of the local modem.

CIRCUIT CS (CLEAR TO SEND)
Signals on this circuit indicate whether the local modem is conditioned to transmit data.

CIRCUIT RR (RECEIVER READY)
Indicates whether the receiver in the Modem is conditioned to receive data signals. (Equivalent to "Line Signal Detect" circuit of RS232C)

CIRCUIT TR (TERMINAL READY)
Signals on this circuit indicate whether the LNC is conditioned to communicate with the modem.

CIRCUIT DM (DATA MODE)
Signals on this line indicate to the LNC that the modem is powered on and ready to communicate. (Equivalent to "Data Set Ready" circuit of RS232C)

APPENDIX A: FIGURES

TABLE OF FIGURES

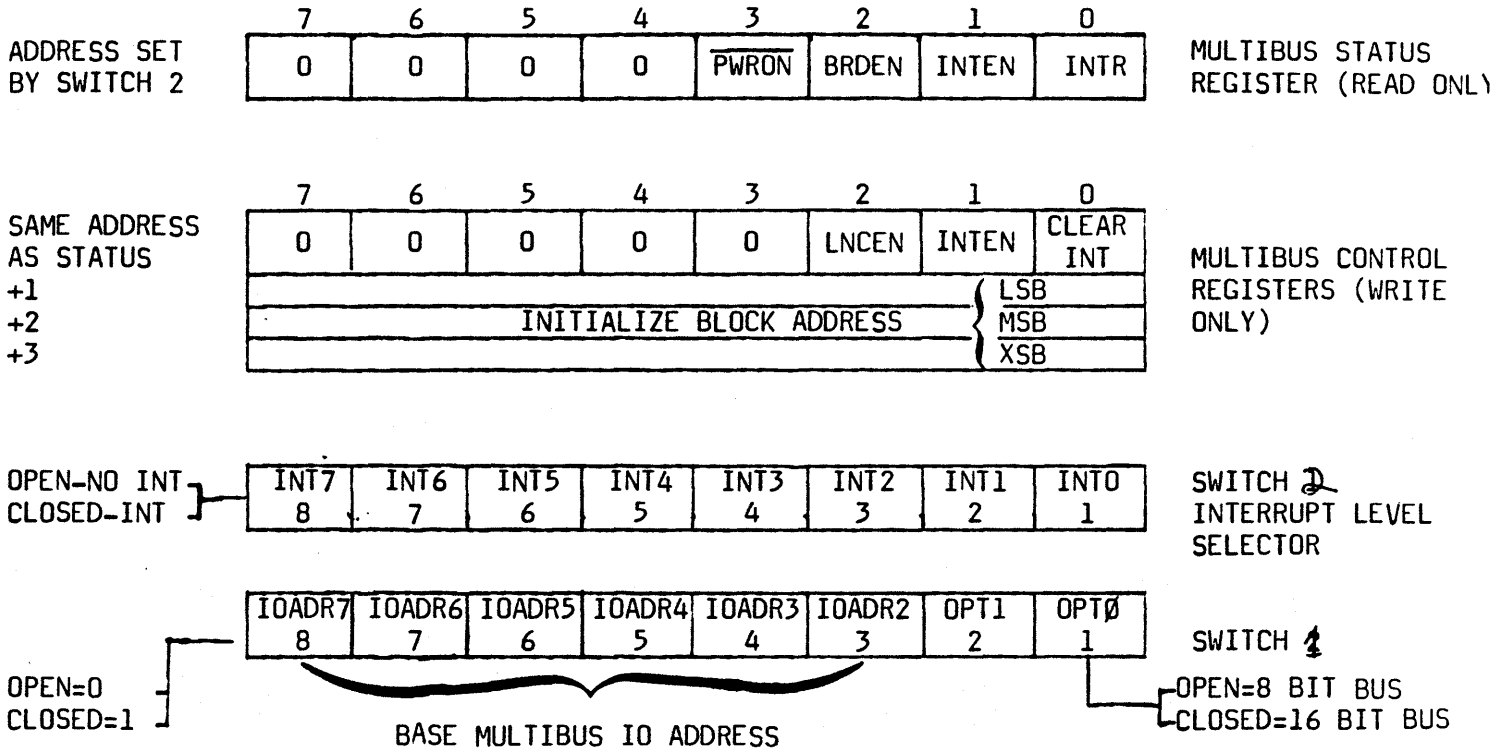
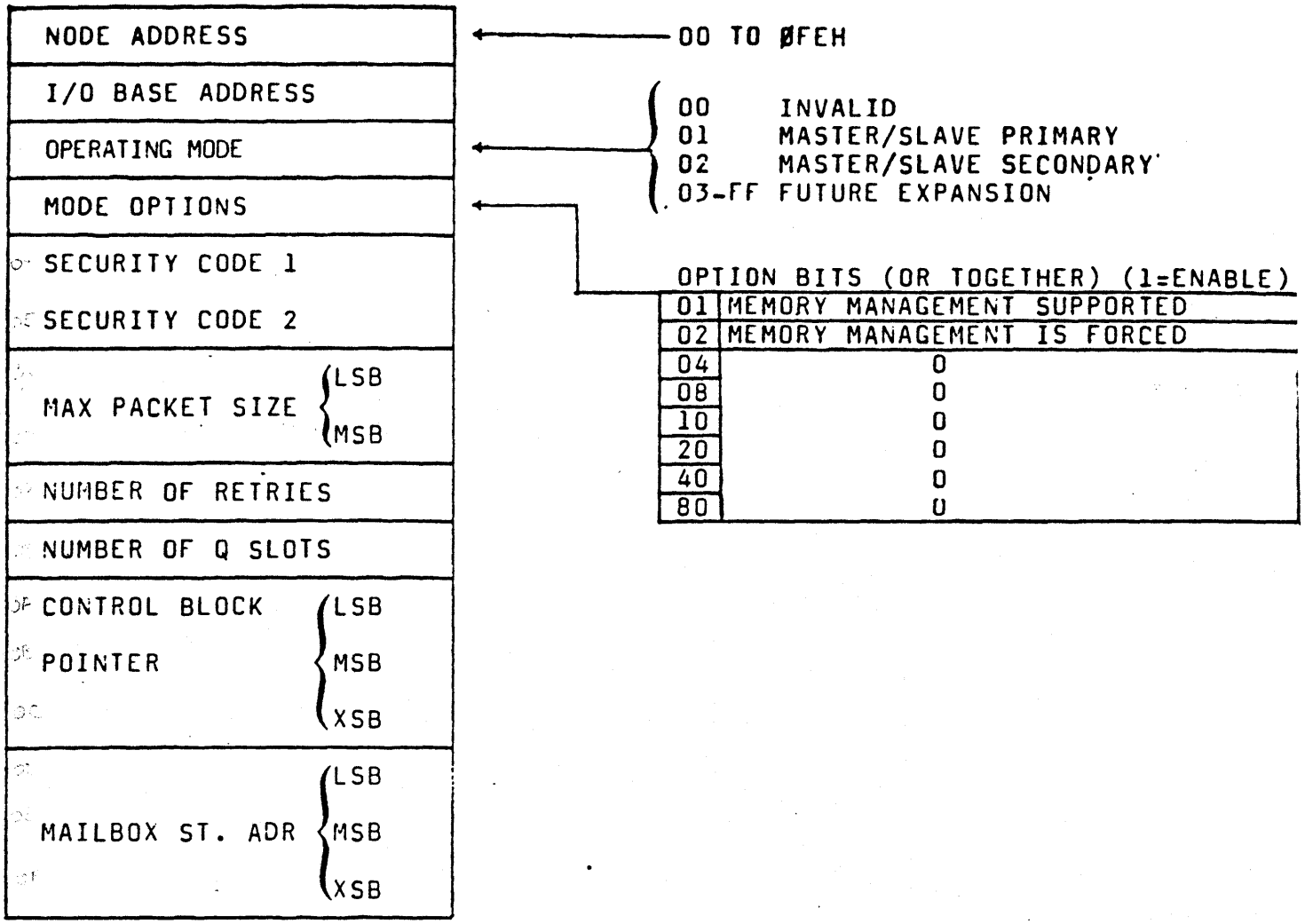

7 6 5 4 3 2 1 0
ADDRESS SET BY SWITCH 2
0 0 0 0 PWRON BRDEN INTEN INTR
MULTIBUS STATUS REGISTER (READ ONLY)
7 6 5 4 3 2 1 0
SAME ADDRESS AS STATUS
0 0 0 0 0 LNCEN INTEN CLEAR INT
+1
+2
+3
INITIALIZE BLOCK ADDRESS
LSB
MSB
XSB
MULTIBUS CONTROL REGISTERS (WRITE ONLY)
OPEN-NO INT
CLOSED-INT
INT7 8
INT6 7
INT5 6
INT4 5
INT3 4
INT2 3
INT1 2
INT0 1
SWITCH 2
INTERRUPT LEVEL SELECTOR
IOADR7 8
IOADR6 7
IOADR5 6
IOADR4 5
IOADR3 4
IOADR2 3
OPT1 2
OPTØ 1
SWITCH 1
OPEN=0
CLOSED=1
BASE MULTIBUS IO ADDRESS
OPEN=8 BIT BUS
CLOSED=16 BIT BUS

FIGURE 1

LNC INITIALIZE BLOCK



F I G U R E 2

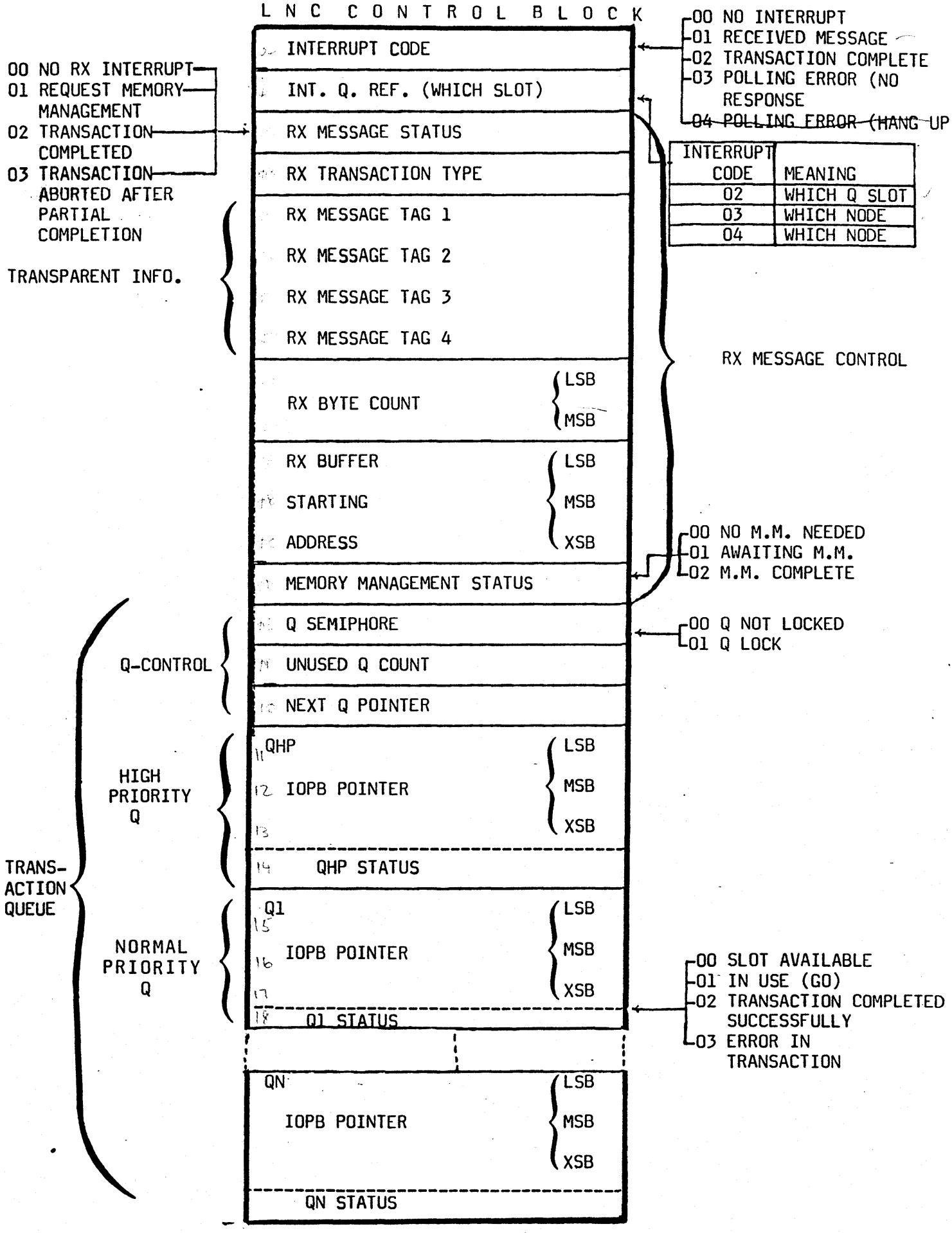

L N C C O N T R O L B L O C K
INTERRUPT CODE
INT. Q. REF. (WHICH SLOT)
RX MESSAGE STATUS
RX TRANSACTION TYPE
RX MESSAGE TAG 1
RX MESSAGE TAG 2
RX MESSAGE TAG 3
RX MESSAGE TAG 4
RX BYTE COUNT
LSB
MSB
RX BUFFER
STARTING
ADDRESS
LSB
MSB
XSB
MEMORY MANAGEMENT STATUS
Q SEMIPHORE
UNUSED Q COUNT
NEXT Q POINTER
QHP
IOPB POINTER
LSB
MSB
XSB
QHP STATUS
Q1
IOPB POINTER
LSB
MSB
XSB
Q1 STATUS
QN
IOPB POINTER
LSB
MSB
XSB
QN STATUS
00 NO INTERRUPT
01 RECEIVED MESSAGE
02 TRANSACTION COMPLETE
03 POLLING ERROR (NO RESPONSE
04 POLLING ERROR (HANG UP
INTERRUPT CODE
MEANING
02
WHICH Q SLOT
03
WHICH NODE
04
WHICH NODE
00 NO RX INTERRUPT
01 REQUEST MEMORY MANAGEMENT
02 TRANSACTION COMPLETED
03 TRANSACTION ABORTED AFTER PARTIAL COMPLETION
TRANSPARENT INFO.
RX MESSAGE CONTROL
00 NO M.M. NEEDED
01 AWAITING M.M.
02 M.M. COMPLETE
00 Q NOT LOCKED
01 Q LOCK
Q-CONTROL
HIGH PRIORITY Q
NORMAL PRIORITY Q
TRANS-ACTION QUEUE
00 SLOT AVAILABLE
01 IN USE (GO)
02 TRANSACTION COMPLETED SUCCESSFULLY
03 ERROR IN TRANSACTION

FIGURE 3

512K

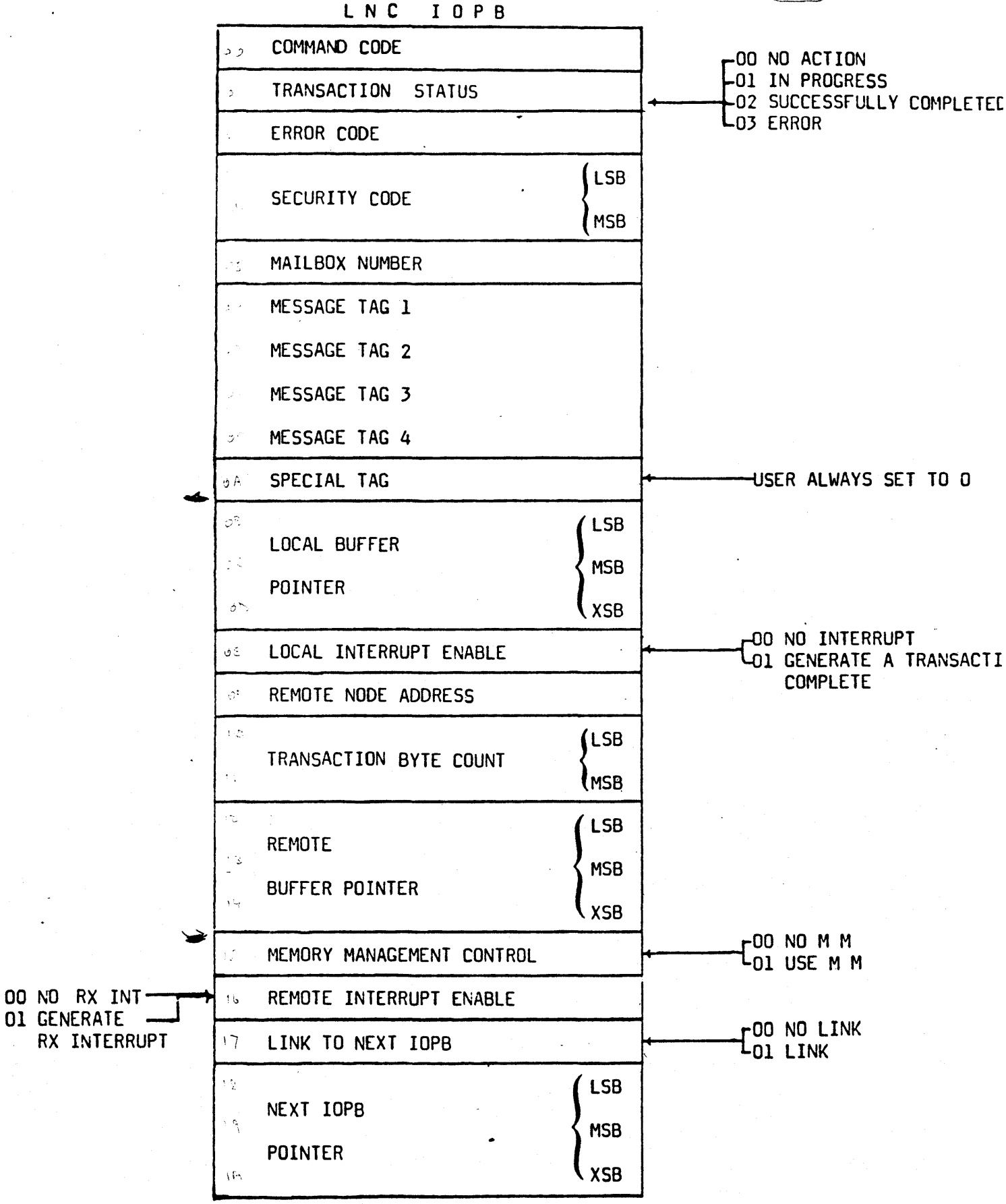


FIGURE 4

L N C C O M M A N D C O D E S

CODE	COMMAND
00	NO OP
01	REINITIALIZE
02	LOCAL SELF TEST
03	TURN ON POLLING
04	TURN OFF POLLING
05	SET POLLING CONTROL TABLE
06	SEND POLL (MANUAL POLLING)
	Delete
07	SEND DATA
08	SEND NO OP
09	REQUEST IMMEDIATE DATA
0A	REQUEST DELAYED DATA
0B	REQUEST LOCAL STATUS
0C	REQUEST REMOTE STATUS

F I G U R E 5

E R R O R C O D E S

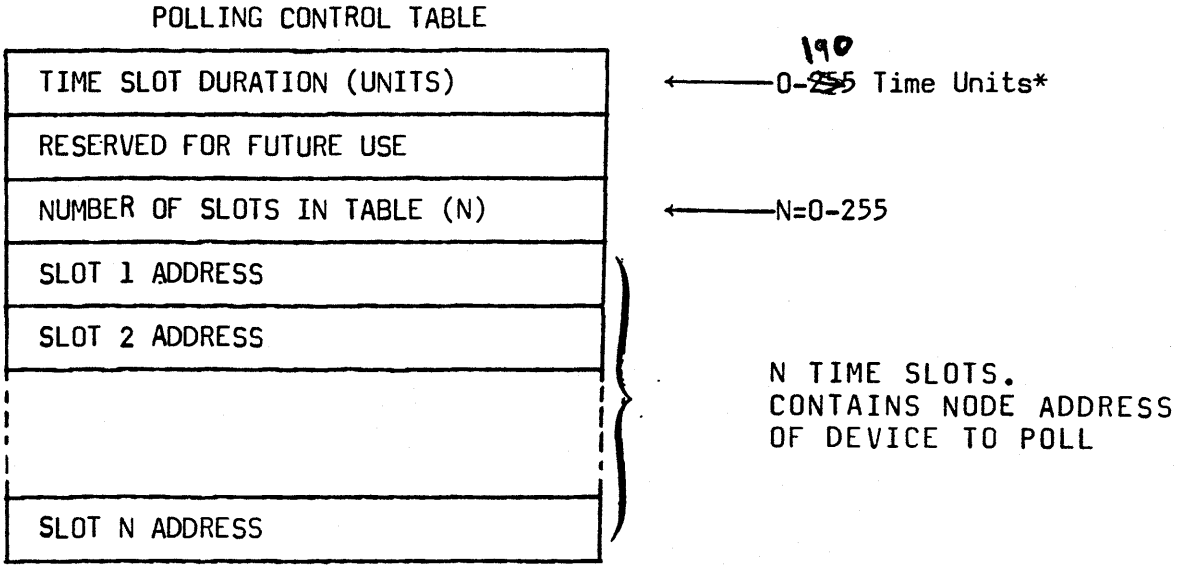
CODE	ERROR TYPE
01	INVALID PARAMETER IN IOPB
02	NO RESPONSE FROM DESTINATION (TIMEOUT)
03	TOO MANY DATA RETRIES
04	DID NOT RECEIVE REQUESTED IMMEDIATE DATA/STATUS
10	LOCAL BUS TIMEOUT
~~11~~	CANNOT READ BUS DATA
~~12~~	CANNOT WRITE BUS DATA
20	REMOTE Q WAS FULL ON REQUEST DELAYED DATA
21	TRANSACTION REJECTED BY DESTINATION
	FUTURE ERROR CODES MAY BE ADDED

ϕB

ϕC

F I G U R E 6

17 Security Error Detected on RECEIVED PACKET



4000

*A time unit is defined to be ~~1000~~ bit times at whatever data rate is used. If 0 units is selected the Slave can start only 1 transaction. At a data rate of 1 Mb/s the unit time is 4 msec.

FIGURE 7

LNC 5180 Statistics

0	Number of attempts to receive frames	LSB	0
1		MSB	1
2		XSB	2
3	Number of "No Response" errors	LSB	3
4		MSB	4
5		XSB	5
6	Number of aborted receive frames	LSB	6
7		MSB	7
8		XSB	8
9	Number of receiver overruns	LSB	9
10		MSB	A
11		XSB	B
12	Number of frames received with	LSB	C
13	CRC Errors	MSB	D
14		XSB	E
15	Number of undifferentiated	LSB	F
16	receive errors	MSB	10
17		XSB	11
18	Number of "No Data Mode" errors	LSB	12
19		MSB	13
20		XSB	14
21	Number of times polled	LSB	15
22		MSB	16
23		XSB	17

A "Data Mode" error generally indicates a Modem problem

When statistics are requested the LNC5180 will build the statistics in user memory at the address specified by the Local Buffer Pointer in the IOPB.

Figure 8

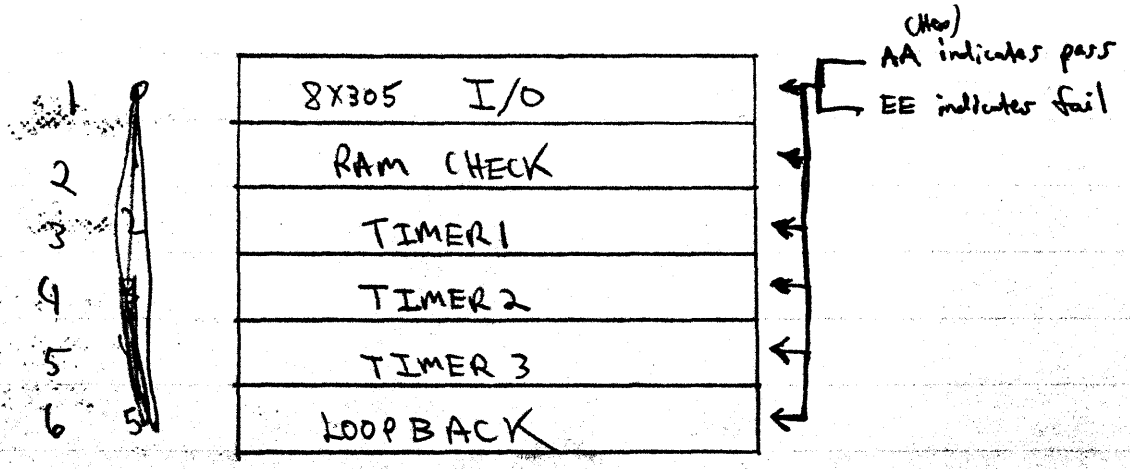
FIGURE 9

PIN	SIGNAL	PIN	SIGNAL
1	GROUND	21	RT B A (RXCLK+)
2	GROUND	22	RT A B (RXCLK-)
3	SD AB (TXDAT+)	23	GROUND
4	SD BA (TXDAT-)	24	GROUND
5	GROUND	25	RD AB (RXDAT+)
6	GROUND	26	RD BA (RXDAT-)
7	RS AB (RTS/+)	27	GROUND
8	RS BA (RTS/-)	28	GROUND
9	GROUND	29	CS AB (CTS/+)
10	GROUND	30	CS BA (CTS/-)
11	RESERVED	31	~~RR~~ AB (DSR/+)
12	RESERVED	32	~~RR~~ BA (DSR/-)
13	TR AB (DTR/+)	33	~~DM~~ AB (LSD/+)
14	TR BA (DTR/-)	34	~~DM~~ BA (LSD/-)
15	GROUND	35	RESERVED
16	GROUND	36	RESERVED
17	ST AB (TXCLK+)	37	RESERVED
18	ST BA (TXCLK-)	38	RESERVED
19	GROUND	39	NOT USED
20	GROUND	40	NOT USED

NOTES: The above signal designations are the official RS 422 signal names with more descriptive initials in parenthesis. All signal lines are balanced.

The mating connector type is ANSLEY 609-4030 (a female 40 contact) connector intended for flat cable mass termination. The flat cable length to the MAU should be limited to 100 ft. or less.

LNC SELF TEST REPORT



Upon completion of the on-board diagnostics the LNC5180 will build the report in user memory at the address specified by the Local Buffer Pointer in the IOPB. When an LNC5180 is operating properly the Self Test Report will contain 6 bytes, of AA Hexadecimal. A byte of EE indicates that the corresponding test has failed. For the timer and loopback tests the LNC5180 must be connected to its corresponding modem or equivalent DCE interface. This is necessary so as to supply TxCLK and RxCLK to the LNC5180. This does not imply that external loopback is executed. All diagnostics are local to the LNC5180.

Figure 10.

GOES IN UNDER "Request Local Status" pg. 27 and 28

All statistics are maintained in 3 byte fields and are described as follows:

ATTEMPTS TO RECEIVE FRAMES - The LNC updates this count each time it recognizes its network address. If any type of error occurs the count will still be incremented.

NO RESPONSE ERRORS - If the LNC is expecting a packet, but does not receive one, this count is incremented.

ABORTED FRAMES - If while in the process of receiving a packet an "abort" character (7 consecutive ones) is detected, the LNC will suspend its attempt to receive and increment this count.

RECEIVER OVERRUNS - If the LNC can not keep up with data coming in off of the network the overrun count is incremented. This condition will only occur in the case of some type of hardware failure.

CRC ERRORS - Bit errors within a packet are automatically detected by the LNC hardware. If a packet is received with error(s) the data is ignored and a positive acknowledgement is not returned to the source node.

UNDIFFERENTIATED RECEIVE ERRORS - This condition occurs when the hardware indicates multiple receive error conditions. An error of this type usually indicates a problem with the USART.

NO DATA MODE - Each time the LWC wants to transmit a packet it checks the "DATA MODE" signal from the associated modem. If the modem is not in "Data Mode" the LWC will not attempt to transmit.

TIMES POLLED - Each time an LWC receives a "POLL" this parameter is incremented. This provides the system level software a mechanism for periodically checking to see if the network is still functioning.

Does Domain software use this to check for deadlock?

Insert Under LOCAL SELF TEST pg. 28

The SELF TEST feature exercises six critical areas of the LNC5180 (fig. 10). It is a hierarchical test sequence with the pinnacle being a full digital loopback check. The six areas are exercised in the following order:

1. 8X305 I/O - The 8X305 is the on-board microcontroller which executes all the protocol firmware and handles I/O on the LNC's internal bus. Various I/O loopback tests are executed to ensure integrity at this most basic level of the hardware architecture.

2. RAM CHECK - The LNC has a 16bit X2048 Random Access Memory array. An extensive statistical analysis is performed on the entire array. The RAM is used for packet buffers and local scratchpad therefore it is imperative that the array is fully intact.

3. TIMER 1 - This timer is used for keeping the "time slot" associated with polling. When the primary station polls a secondary this timer is set as the response window. If the secondary does not reply before a timeout occurs a poll error is posted (Assuming all retries have been exhausted). The LNC must be connected to a fully operational modem running between 1 and 2 MHz (The TXCLK is supplied by the modem in a synchronous architecture) for this test to be completed successfully.

④ TIMER2 - This timer is used for maintaining the "data link" layer of the LNC firmware. If packets or acknowledgements are lost, the timer is used to initiate error recovery procedures. As with TIMER1, the LNC must be connected to a modem supplying a TXCLK running at a rate between 1 and 2 MHZ.

⑤ TIMER3 - This timer is used as a "watchdog" on all data transfer across the multibus. If a read or write cycle does not get "acknowledged" within 1.5ms, the TIMER3 expires and a bus transfer error is posted.

⑥ LOOPBACK - A digital loopback test is executed which checks the communication interface up to, but not including, the line drivers and receivers. Data is fed back through the USART and integrity is checked.

GENERAL PURPOSE 37-POSITION AND 9-POSITION INTERFACE FOR DATA TERMINAL EQUIPMENT AND DATA CIRCUIT-TERMINATING EQUIPMENT EMPLOYING SERIAL BINARY DATA INTERCHANGE

FOREWORD

(This Foreword provides additional information and does not form an integral part of the EIA Standard specifying the General Purpose 37-Position and 9-Position Interface for Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange.)

This Standard, together with EIA Standards RS-422 and RS-423, is intended to gradually replace EIA Standard RS-232-C as the specification for the interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) employing serial binary data interchange. With a few additional provisions for interoperability, equipment conforming to this standard can interoperate with equipment designed to RS-232-C. This standard is intended primarily for data applications using analog telecommunications networks.

EIA Standard RS-232-C is in need of replacement in order to specify new electrical characteristics and to define several new interchange circuits. New electrical characteristics are needed to accommodate advances in integrated circuit design, to reduce crosstalk between interchange circuits, to permit greater distances between equipments, and to permit higher data signaling rates. With the expected increase in use of standard electrical interface characteristics between many different kinds of equipment, it is now appropriate to publish the electrical interface characteristics in separate standards. Two electrical interface standards have been published for voltage digital interface circuits:

858

EIA Standard RS-422, Electrical Characteristics of Balanced Voltage Digital Interface Circuits

EIA Standard RS-423, Electrical Characteristics of Unbalanced Voltage Digital Interface Circuits

With the adoption of EIA Standards RS-422 and RS-423, it became necessary to create a new standard which specifies the remaining characteristics (i.e., the functional and mechanical characteristics) of the interface between data terminal equipment and data circuit-terminating equipment. That is the purpose of this standard.

The basic interchange circuit functional definitions of EIA Standard RS-232-C have been retained in this standard. However, there are a number of significant differences:

a. Application of this standard has been expanded to include signaling rates up to 2,000,000 bits per second.

b. Ten circuit functions have been defined in this standard which were not part of RS-232-C. These include three circuits for control and status of testing functions in the DCE (Circuit LL, Local Loopback; Circuit RL, Remote Loopback; and Circuit TM, Test Mode), two circuits for control and status of the transfer of the DCE to a standby channel (Circuit SS, Select Standby; and Circuit SB, Standby Indicator), a circuit to provide an out-of-service function under control of the DTE (Circuit IS, Terminal In Service), a circuit to provide a new signal function (Circuit NS, New Signal) and a circuit for DCE frequency selection (Circuit SF, Select Frequency). In addition, two circuits have been defined to provide a common reference for each direction of transmission across the interface (Circuit SC, Send Common; and Circuit RC, Receive Common).

c. Three interchange circuits defined in RS-232-C have not been included in this standard. Protective ground (RS-232-C Circuit AA) is not included as part of the interface to permit bonding of equipment frames, when necessary, to be done in a manner which is in compliance with national and local electrical codes. However, a contact on the interface connector is assigned to facilitate the use of shielded interconnecting cable. The two circuits reserved for data set testing (RS-232-C contacts 9 and 10) have not been included in order to minimize the size of the interface connector.

d. Some changes have been made to the circuit function definitions. For example, operation of the Data Set Ready circuit has been changed and a new name, Data Mode, has been established due to the inclusion of a separate interchange circuit (Test Mode) to indicate a DCE test condition.

e. A new set of standard interfaces for selected communication system configurations has been established. In order to achieve a greater degree of standardization, the option in RS-232-C which permitted the omission of the Request to Send interchange circuit for certain transmit only or duplex primary channel applications has been eliminated.

f. A new set of circuit names and mnemonics has been established. To avoid confusion with RS-232-C, all mnemonics in this standard are different from those used in RS-232-C. The new mnemonics were chosen to be easily related to circuit functions and circuit names.

g. A different interface connector size and interface connector latching arrangement has been specified. A larger size connector (37-position) is specified to accommodate the additional interface leads required for the ten newly defined circuit functions and to accommodate balanced operation for ten interchange circuits. In addition, a separate 9-position connector is specified to accommodate the secondary channel interchange

circuits. The 37-position and 9-position connectors are from the same connector family as the 25-position connector in general use by equipment conforming to EIA Standard RS-232-C. A connector latching block is specified to permit latching and unlatching of the connectors without the use of a tool. This latching block will also permit the use of screws to fasten together the connectors. The different connectors will also serve as an indication that certain precautions with regard to interface voltage levels, signal risetimes, fail safe circuitry, grounding, etc. must be taken into account before equipment conforming to RS-232-C can be connected to equipment conforming to the new electrical characteristic standards. The connector contact assignments have been chosen to facilitate connection of equipment conforming to this standard to equipment conforming to RS-232-C.

Close attention was given during the development of RS-449 and RS-423 to facilitate an orderly transition from the existing RS-232-C equipment to the next generation without forcing obsolescence or costly retrofits. It will therefore be possible to connect new equipment designed to RS-449 on one side of a interface to equipment designed to RS-232-C on the other side of the interface. Such interconnections can be accomplished with a few additional provisions associated only with the new RS-449 equipment. These provisions are discussed in an EIA Industrial Electronics Bulletin (IE Bulletin No. 12). Application Notes on Interconnection Between Interface Circuits Using RS-449 and RS-232-C.

This standard is designed to be compatible with the specifications of the International Telegraph and Telephone Consultative Committee (C.C.I.T.T.) and the International Organization for Standardization (ISO). However, it should be noted that this standard contains a few specifications which are subjects of further study in C.C.I.T.T. and ISO. These are:

860

1) Use of interchange circuits Terminal In Service and New Signal.

2) Status of interchange circuits during an equalizer retraining period.

The U.S.A. is actively participating in C.C.I.T.T. and ISO to gain international agreement on these items.

Work is presently underway, in cooperation with C.C.I.T.T. and ISO, to expand the Remote Loopback test function to include testing on multipoint networks. This augmentation will not affect the point-to-point testing capability specified in this document. Work is also underway to augment this standard to cover direct DTE to DTE applications. This augmentation will not affect, in any way, the DTE to DCE operation specified in this document. In addition, work will proceed in cooperation with C.C.I.T.T. toward the development of a more efficient all-balanced interface which minimizes the number of interchange circuits. It is expected that RS-449 will provide the basis for this new work.