DP83956EB-AT LERIC™ (LitE Repeater Interface Controller) PC-AT® Adapter

National Semiconductor
Application Note 854
Marc Clevenger
Imad Ayoub
C.S. Balasvbramanian
November 1992

1.0 INTRODUCTION

This LERIC-NIC Evaluation Board provides IBM® PC-AT and AT compatible computers with Twisted Pair conductivity. The board uses the DP8390 (NIC) to perform the Ethernet® protocol operations and the DMA operations. The dual DMA (local and remote) capabilities of the NIC, along with 16 kBytes of buffer RAM, allow the entire Network Interface Adapter to appear as a standard I/O Port to the system. The NIC module's local DMA channel buffers packets between the local memory (16 kBytes of buffer RAM) and the network, while the NIC module's remote DMA channel passes data between the local memory and the system memory by way of an I/O Port. This I/O Port architecture, which isolates the CPU from the network traffic, proves to be the simplest method to interface the DP8390 to the system. The DP83956 (LERIC) is used to interface to twisted pair Ethernet and provides IEEE 802.3 (Chapter 9) compliant repeater functions to six twisted pair ports. The LERIC has an on-chip PLL for Manchester data decoding, a Manchester encoder and an Elasticity buffer for preamble regeneration. It also has 6 integrated 10BASE-T transceivers. The LERIC's internal registers can be accessed using the same I/O port architecture as the NIC. This board provides the required attributes for compliance with Novell's® Hub Management Interface (HMI) basic control capability.

2.0 BOARD OVERVIEW

The LERIC-NIC board allows direct connection to the network using the RJ-45 phone jacks. There are 6 ports on a card. In addition, up to 4 boards can be cascaded together in a PC-AT, thus providing 24 Twisted Pair ports.

The block diagram shown in *Figure 1* illustrates the architecture of the LERIC-NIC Evaluation Board. The LERIC-NIC Board as seen by the PC-AT system appears only to be an I/O port. With this architecture the LERIC-NIC board has its own local bus to access the board memory. The system never has to intrude further than the I/O ports for any packet data operation.

2.1 Hardware Features

- Utilizes DP83956 LitE Repeater Interface Controller (LERIC)
- Six 10BASE-T connections per card and one node connection utilizing the NIC
- Cascadability of up to 4 boards
- 16 kByte on-board Packet Buffer
- Simple I/O Port Interface to IBM PC-AT
- Interfaces to Twisted Pair (10BASE-T)
- Boot EPROM Socket

The detailed schematics for this design are shown at the end of this document.

3.0 BOARD ARCHITECTURE

3.1 Board I/O Map

The LERIC-NIC Board requires a 32-byte I/O space to allow for decoding the data buffers, the reset port, and the NIC and LERIC registers. The first 16 bytes (300h–30Fh) are used to address the LERIC (4 bits wide) and NIC registers (8 bits wide) and the next 8 bytes (310h–317h) are used to address the data buffers which are 16 bits wide. Finally, the reset port (also software selectable) may be addressed by 318h–31Fh.

TABLE I. I/O Map in PC-AT

Address	Part Addressed
300h–30f	NIC/LERIC Select
310h–317	Data Buffers
318h–31f	Reset

Although in the description above the I/O map is positioned at the addresses 300–31F, it may also be placed in the following address spaces: 320–33F, 340–35F, 360–37F. These alternate address spaces may be selected by the two jumpers (JP1 and JP0) as shown in Table II.

TABLE II. Optional Address Spaces

JP1	JP0	I/O Address Space
ON	ON	300h–31Fh
ON	OFF	320h–33Fh
OFF	ON	340h–35Fh
OFF	OFF	360h–37Fh

3.2 Data and Address Paths

The following paragraph may be better understood by looking at the block diagram shown in *Figure 1*. Twenty address lines from the PC® go onto the LERIC-NIC Board, but only four of them actually go to the LERIC and the NIC. These four addresses along with the $\overline{IOR}$ (low-asserted I/O read) or $\overline{IOW}$ (low-asserted I/O write) and the $\overline{CS}$ (NIC chip select signal) allow the PC to read or write to the LERIC and NIC's registers. If the system wants to read from or write to the LERIC or NIC registers, the data (8 bits for the NIC and 4 bits for the LERIC) must pass through the appropriate 245 buffer.

All of the packet data will pass through the I/O ports (the 374's). Each 374 is unidirectional and can only drive 8 bits, therefore it is necessary to have four 374's. Two of which drive data from the ports to the board memory and two of which drive the data from the ports to the AT bus. Even the PROM, which can only be addressed by the NIC, sends its 8 bits of data out through the 374's. When the PROM does this, two of the 374's will be enabled but only the lower 8 bits will have valid data. The RAM is also accessed by the NIC. However, it is addressed by 14 bits and drives out 16 bits of data.

RRD-B30M115/Printed in U. S. A.

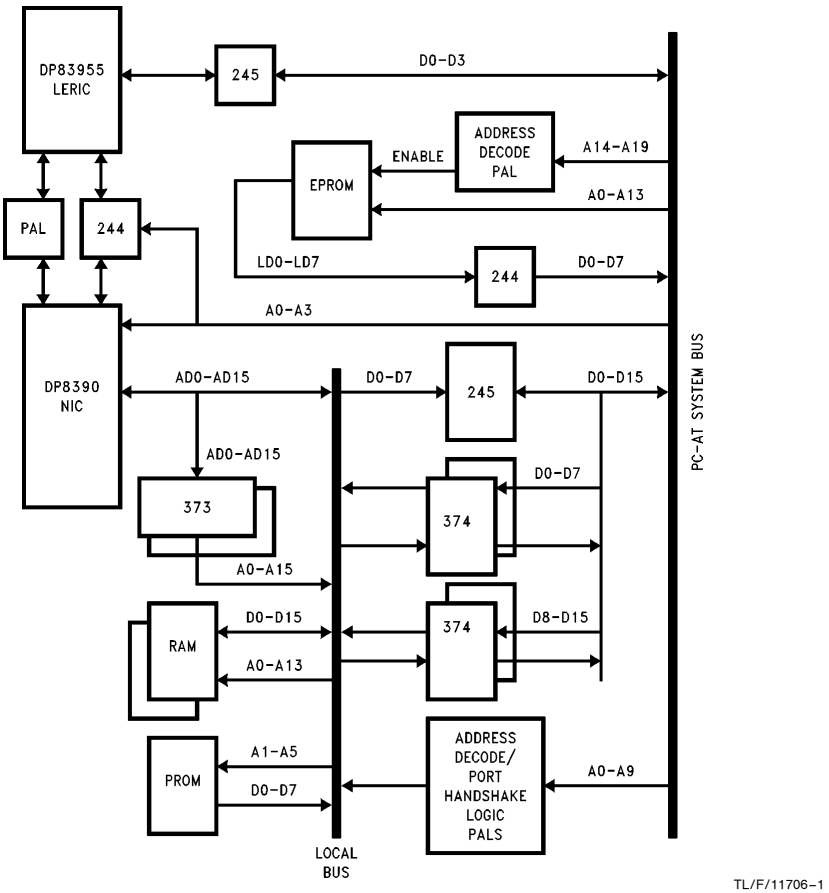


FIGURE 1. LERIC-NIC PC-AT System Interface

The PALs® receive 7 address lines among many other signals such as $\overline{IOR}$, $\overline{IOW}$, $\overline{ACK}$, MRD, etc. With these signals the PALs do all of the decodes, such as selecting the LERIC-NIC Board, the LERIC chip, the NIC chip, the RAM, and the PROM.

The EPROM socket is provided so that the user may add an EPROM to the system. This EPROM would normally contain a program and a driver to enable the PC-AT to be booted up through the network. The chips necessary to interface the EPROM to the system are the 27128 (EPROM), a 16L8 (PAL), and a 74ALS244 (buffer). Also, JP7 must be placed in the proper selection as described in the jumper section. The PAL decodes SA14–SA19, along with SMRDC (system memory read), in order to generate the EPROMEN signal. This signal, issued when the PC wants to execute the EPROM and the 244 buffer.

3.3 Global Register Description

An additional 3-bit write only register is provided on the board to allow for accesses to the LERIC. This register is also accessed using the I/O port architecture, and one address location in the I/O space has been allocated for it. This register can be mapped to one of four address location. This is done by using jumpers JP2 and JP3 as shown in Table III. The CPU can only write to this register.

TABLE III. Optional Address Spaces for Global Register

JP2	JP3	I/O Address Space
ON	ON	200h
OFF	ON	220h
ON	OFF	240h
OFF	OFF	260h

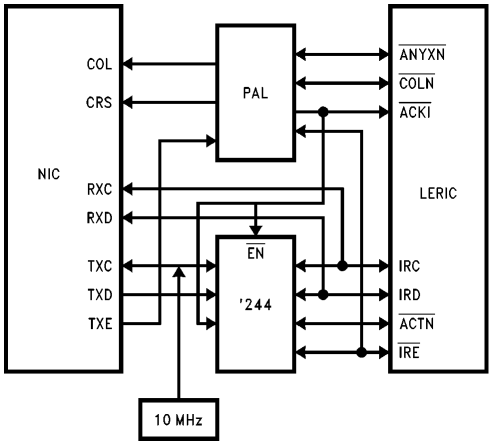
The three bits in this register are shown in Table IV. Bits 0 and 1 are used to specify the board number. Since there can be up to four boards cascaded together, a unique board address is necessary to distinguish between them. Bit 2 is used to select the NIC or the LERIC when CPU accesses are being made. On power-up this bit defaults to one and selects the NIC. In order to access the LERIC, this bit has to be set to zero.

TABLE IV. Global Register Bits

Bit 2	Bit 1	Bit 0
NIC/LERIC	Board #	Board #

4.0 LERIC-NIC INTERFACE

The LERIC-NIC interface makes use of the Inter-LERIC™ Bus which consists of the following signals: $\overline{ACKO}$, $\overline{ACKI}$, IRC, IRD, $\overline{IRE}$, $\overline{ACTN}$, $\overline{ANYXN}$ and $\overline{COLN}$. Besides NRZ data (IRD) and clock (IRC) this bus provides other signals necessary for cascading one LERIC to another. The NIC is treated as another LERIC when it is connected to the Inter-LERIC bus. The Inter-LERIC Bus also eliminates the need for an encoder/decoder chip to which the NIC is usually connected. Since the Inter-LERIC Bus is bidirectional and some logical operations are necessary to convert the signals to be compatible with the NIC, a PAL and some TRI-STATE® buffers are used to implement this function. *Figure 2* shows the interface between the LERIC and NIC. In this implementation the NIC is placed on the top of the arbitration chain. The NIC input pins RXC and RXD are connected directly to the LERIC pins IRC and IRD, respectively. The COL input of the NIC is derived by combining the $\overline{COLN}$ and $\overline{ANYXN}$ signals from the LERIC. The CRS input on the NIC comes from the inverted $\overline{IRE}$ signal of the LERIC. When the NIC wants to transmit, it drives the $\overline{ACKI}$ input of the LERIC with the inverted TXE signal. The inverted TXE signal is also used to enable the 244 TRI-STATE buffer which connects the NIC output signals TXD, TXC to the IRD and IRC signals on the Inter-LERIC Bus. $\overline{TXE}$ is used to drive the $\overline{ACTN}$ and $\overline{IRE}$ signals during transmission.



TL/F/11706–2

CRS = $\overline{IRE}$
COL = $\overline{COLN}$ + $\overline{ANYXN}$
$\overline{EN}$ = $\overline{TXE}$

FIGURE 2. LERIC-NIC Interface

4.1 Inter-LERIC Bus Interface

The LERIC-NIC board, or master board, may also be used without the NIC. A board with the LERIC only (no NIC), or slave board, is desirable when more boards are to be used in the same PC. One master and up to three slave boards can be cascaded using the Inter-LERIC Bus interface to form a larger logical repeater, (i.e., one that meets IEEE's specification for a single repeater). There are two 14-pin headers on the board for cascading the Inter-LERIC bus signals. Both headers contain the signals IRC, IRD, $\overline{IRE}$, $\overline{ACTN}$, $\overline{ANYXN}$, and $\overline{COLN}$. The input header, J7, also contains the $\overline{ACKI}$ signal, and the output header, J9, contains $\overline{ACKO}$. These signals enable multiple LERICs to be cascaded together. The $\overline{ACKI}$ and $\overline{ACKO}$ signals are daisy chained between the boards. The $\overline{ACKO}$ signal will drive the $\overline{ACKI}$ input of the board which is next on the arbitration chain. Jumper JP8 (see schematic) is used to tie $\overline{ACKI}$ to the inverted TXE signal from the NIC, which puts the NIC at the top of the arbitration chain. If all boards are used in slave mode, JP8 can also tie $\overline{ACKI}$ high, putting that board on top of the chain. Otherwise, with JP8 removed, the $\overline{ACKI}$ signal will be driven by the $\overline{ACKO}$ output of the board higher up on the arbitration chain. Since these signals are held TRI-STATE or open collector they are pulled up by resistors. The resistor value of 8.2 kΩ is selected for these pull-ups. When all four boards are cascaded, the smallest pull-up value on any Inter-LERIC signal will be approximately 2 kΩ. This elevates the need to remove some of these pull-up resistors when additional boards are cascaded.

5.0 BOARD OPERATION

The following pages describe the slave accesses to the LERIC-NIC and the local DMA and remote DMA operation.

5.1 Global Register Operations

Accesses to the board are register operations to the NIC or the LERIC, which are done to set up the NIC to control the operation of the NIC's DMA channels, and read and write to the LERIC registers. Since the NIC and LERIC share the same I/O space for the registers (300h–30Fh), an additional CPU operation is required. Before any register read or write, the CPU performs a write to the global register bit 2 in order to select the LERIC or NIC and to bits 0 and 1 selecting one of the four possible boards. The usage of this bit depends on the software used. If a normal network (no hub access) driver is used the card looks like a pure adapter to the software. The board normally would be set with the NIC selected. When the driver needs to access the LERIC it would first write to the Global Register, do the LERIC operations, then set it back to enable the NIC access. This minimizes the changes to the NIC portion of the driver.

To begin the global register write (see Section 3.0 for the Global Register description), the CPU drives the SA0–SA3 address lines to the LERIC-NIC board and the SA4–SA9 address lines to the PAL. With these address lines, the PAL decodes to 200, 220, 240 or 260 depending on the settings of jumpers JP2 and JP3. The CPU then drives the $\overline{IOW}$ strobe which is used to latch in the data on the AT bus into the Global Register on the rising edge of $\overline{IOW}$. This ends the cycle of the global register write.

5.2 LERIC Register Accesses

Before any LERIC register access, the CPU must write to the global register in order to select the LERIC and the appropriate LERIC-NIC board. After the register access, the CPU must perform another write to the global register to select the NIC.

5.2.1 LERIC Register Read

To begin the register read, the CPU drives the four address lines (SA0–SA3) to the LERIC and the SA4–SA9 address lines to the PAL. With these address lines and the $\overline{IOR}$ line, the PAL decodes to 300–30F (the LERIC registers) and the $\overline{LRD}$ signal is enabled. Once the LERIC receives this $\overline{LRD}$, it then sends out a low assertion on $\overline{BUFEN}$. The $\overline{BUFEN}$ signal is used by the PAL to assert the IOCHRDY signal false. The LERIC then drives out the data from its internal registers to the 245 buffer. The 245 buffer is then enabled by the $\overline{LERICEN}$ signal and the data is driven onto the AT BUS. A 3-bit counter is used to indicate when the LERIC has

driven the data out. This gives the LERIC enough time to output the data. This is required since the LERIC does not have any other signal which indicates that the data is available. This is indicated by the signal COUNT__4 going high (after about 400 ns) which causes the PAL to assert IOCHRDY true. As a result, $\overline{IOR}$ is driven high by the CPU, thereby de-asserting the $\overline{LRD}$. On the rising edge of the $\overline{IOR}$, the data which is on the AT BUS is latched into the system. The addresses are removed at the same time, causing the LERIC chip select to become de-asserted, ending the register read cycle.

5.2.2 LERIC Register Write

To begin the register write, the CPU drives the four address lines (SA0–SA3) to the LERIC and the SA4–SA9 address lines to the PAL. With these address lines and the $\overline{IOW}$ line, the PAL decodes to 300–30F (the LERIC registers) and the $\overline{LWR}$ signal is enabled. Once the LERIC receives this $\overline{LWR}$, it then sends out a low assertion on $\overline{BUFEN}$. The $\overline{BUFEN}$ signal is used by the PAL to assert the IOCHRDY (used to insert wait states) signal false. The CPU then drives out the data onto the AT BUS where it goes into the 245 buffer. The 245 buffer is then enabled by the $\overline{LERICEN}$ signal and the data is driven to the LERIC. A 3-bit counter is used to indicate when the LERIC has latched the data in. This is indicated by the signal COUNT__4 going high which causes the PAL to assert IOCHRDY true. As a result, $\overline{IOW}$ is driven high by the CPU, thereby de-asserting the $\overline{LWR}$. The addresses are removed at the same time, causing the LERIC chip select to become de-asserted, ending the register write cycle.

5.3 NIC Register Accesses

The following discussion assumes a jumper setting for I/O address space of 300h–31Fh.

5.3.1 NIC Register Read

To begin the register read, the CPU drives the four address lines (SA0–SA3) to the NIC and the SA4–SA9 address lines to the PAL. With these address lines, PAL #2 decodes to 300–30F (the NIC registers) thereby enabling the chip select for the NIC. The CPU also drives the $\overline{IOR}$ line which the NIC sees as the $\overline{SRD}$ (slave read). Since the NIC may be a local bus master when the CPU attempts to read or write the controller, an $\overline{ACK}$ line is used by the PAL to assert the IOCHRDY signal false and wait state the CPU. The NIC drives out the data from its internal registers to the 245 buffer. When the NIC is ready to be in slave move and complete the read cycle, it asserts $\overline{ACK}$ true which enables the 245 buffer and the data is driven onto the AT BUS. Driving $\overline{ACK}$ true also causes the PAL to assert IOCHRDY true. As a result, $\overline{IOR}$ is driven high by the CPU, thereby de-asserting the $\overline{SRD}$. On the rising edge of the $\overline{IOR}$, the data which is on the AT BUS is latched into the system. The addresses are removed at the same time, causing the NIC chip select to become de-asserted, ending the register read cycle.

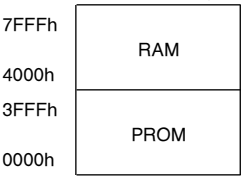
5.3.2 NIC Register Write

To begin the register write, the CPU drives the SA0–SA3 address lines to the NIC and the SA4–SA9 address lines to the PAL. With these address lines, the PAL decodes to 300–30F (the NIC registers) thereby enabling the chip select for the NIC. The CPU then drives the $\overline{IOW}$ strobe which the NIC sees as $\overline{SWR}$ (slave write). Once the NIC receives this $\overline{SWR}$ it sends back a low assertion on $\overline{ACK}$ to acknowledge that it is in slave mode and ready to perform the write. A low assertion on $\overline{ACK}$ will generate IOCHRDY true and enable the 245 buffer. The 245 buffer then drives the data from the AT BUS to the NIC. The system drives $\overline{IOW}$ high, thereby de-asserting the $\overline{SWR}$ and latching the data. The addresses also are taken away and the chip select then goes high (de-asserted). This ends the cycle of the register write.

5.4 NIC Local Memory Map

There are only two items mapped into the local memory space. These two items, shown in Table V, are the 8k x 16k buffer RAM and the ID address PROM. The buffer RAM is used for temporary storage of transmit and receive packets. For transmit packets, the remote DMA puts data from the I/O ports into the RAM and the local DMA moves the data from the RAM to the NIC. For the receive packets, the local DMA carries the data from the NIC to the RAM and the remote DMA moves the data from the RAM to the I/O ports.

TABLE V. NIC Local Memory Map



The ID address PROM (74S288 32 x 8) contains the physical address of the evaluation board. Each PROM holds it own unique physical address which is installed during its manufacture. Besides this address, the PROM also contains a checksum. This checksum, calculated by exclusive ORing the six address bytes with each other, is provided in order to check the addresses. At the initialization of the evaluation board the software commands the NIC to transfer the PROM data to the I/O Port where it is read by the CPU. The CPU then verifies the checksum and loads the NIC's physical address registers. Table VI shows the contents of the PROM.

TABLE VI. PROM Contents

PROM Location	Location Contents
00h	Ethernet Address 0 (most significant byte)
01h	Ethernet Address 1
02h	Ethernet Address 2
03h	Ethernet Address 3
04h	Ethernet Address 4
05h	Ethernet Address 5
06h–0Dh	00h
0Eh, 0Fh	57h
10h–15h	Ethernet Address 0 through 5
16h–1Dh	Reserved
1Eh, 1Fh	42h

5.5 NIC Remote DMA Packet Data Transfers

Remote DMA transfers are operations performed by the NIC on the board. These operations occur when the NIC is programmed to transfer packet data between the PC-AT and the card's on-board RAM. These transfers take place through the I/O Port interfacing.

5.5.1 Remote Read

To program the NIC for a remote read, the CPU must take five slave accesses to the NIC. The CPU must write the Remote Start Address (2 bytes) and the Remote Byte Count (2 bytes). Then the CPU issues the Remote DMA Read command.

Once the NIC has received all of the above data, it drives out BREQ and waits for BACK. The NIC immediately receives BACK because it is tied to the BREQ line. BREQ can be tied to BACK because there are no other devices contending for the local bus. After receiving the BACK, the NIC drives out the address from which the data is required to be read. This address flows into the 373's and is latched by ADS0. From here, the address flows to the RAM. The RAM waits until it receives $\overline{\text{MRD}}$ from the NIC and then it drives the data into the 374 ports. The 374 ports then latch the data on the rising edge of the $\overline{\text{PWR}}$ strobe from the NIC. PRQ is then sent out by the NIC to let the system know that there is data waiting in the ports.

If the AT reads the I/O ports before the NIC has loaded the 374's, then the port request (PRQ) from the NIC will not yet be driven. This unasserted PRQ signal causes the AT's ready line to be set low, indicating that the NIC has yet to load the data. After the data is in the ports, the system must then read the 374 data ports. This begins with the AT driving out an address which is decoded (inside PAL #1) to the data I/O Ports (310–317). PAL #2 then drives $\overline{\text{RACK}}$ to the NIC, indicating that the CPU is ready to accept data. This $\overline{\text{RACK}}$ signal then reads the data from the 374 ports onto the AT BUS. The system deasserts $\overline{\text{IOR}}$ which finishes the cycle.

5.5.2 Remote Write

Like the remote read, the remote write cycle also begins with five slave accesses to the internal registers. The CPU must write the Remote Start Address (2 bytes), the Remote Bytes Count (2 bytes), and issue the Remote DMA write command. The NIC then issues a PRQ. The CPU responds by sending an $\overline{\text{IOW}}$, indicating that it is ready to write to the ports. The CPU also drives out the address which corresponds to the I/O Ports. PAL #2 generates $\overline{\text{WACK}}$ on an address decode to the data buffers along with PRQ and $\overline{\text{IOW}}$. This $\overline{\text{WACK}}$ signal latches the data into the 374 ports. The NIC issues a BREQ and immediately receives BACK since the two lines are tied together. The NIC, upon receiving BACK, drives out address lines to the 373's. These address lines are latched by ADS0 and then are driven to the RAM. The NIC then sends out a $\overline{\text{PRD}}$ and a $\overline{\text{MWR}}$ which drives the data from the 374 ports into the already specified address of the onboard RAM. $\overline{\text{PRD}}$ and $\overline{\text{MWR}}$ are then deasserted and the cycle ends.

5.6 Network Transfers from NIC to Buffer RAM

Transfers to and from the network are controlled by the NIC's local DMA channel which transfers packet data to/from the NIC's internal FIFO from/to the card's buffer RAM.

5.6.1 Data Reception

The data received from the network, is deserialized and is loaded into the FIFO inside of the NIC. The NIC then issues a BREQ and immediately receives BACK since the lines are tied together. After receiving BACK, the NIC drives the address lines to the 373's. The 373's are latched by ADS0 and the address is allowed to flow to the RAM. Then the NIC drives out $\overline{\text{MWR}}$ along with the data from the FIFO. The data flows into the RAM at the address given earlier. The $\overline{\text{MWR}}$ strobe is then de-asserted, ending the cycle.

5.6.2 Data Transmision

To begin the transmit cycle, the NIC issues a BREQ and waits for BACK. Since BREQ and BACK lines are tied together, BACK is received immediately. Upon reception of this signal, the NIC drives out the address to the 373's which latch the address with the ADS0 strobe. The address then flows to the on-board memory. MRD, driven by the NIC, causes the RAM to drive the data out of the given address and into the NIC. The NIC then latches the data into the FIFO on the rising edge of $\overline{\text{MRD}}$. The high assertion of $\overline{\text{MRD}}$ signifies the ending of this cycle. From the FIFO, the data is serialized and transmitted on the network.

6.0 TWISTED PAIR INTERFACE

The interface to the network through the LERIC twisted pair ports is shown in *Figure 3*. To drive the transmitted signal through 100 meters of Unshielded Twisted Pair (UTP) cable, the LERIC requires external drivers. The optimized resistor network shown provides the proper pre-emphasis on the transmit signals and the 100Ω termination on the receive pair. Standard Filter Transformer Choke modules (such as Valor FL1012) are used to provide the required filtering and isolation.

7.0 BOARD CONFIGURATION

The LERIC is initialized during power-on reset. On the rising edge of the reset signal from the PC-AT, the data on the pins D0–D7 are loaded into the configuration registers. This reset is tied directly to the $\overline{\text{MLOAD}}$ pin of the LERIC. (Refer to the LERIC datasheet for the description of $\overline{\text{MLOAD}}$.) On the LERIC-NIC board there are pull-up resistors on pins D0–D7, which will load 1's into the configuration registers during power up. The all 1's pattern configures the LERIC in the six twisted pair ports and one full AUI mode, enables the polarity switching on the twisted pair ports, selects the 31 consecutive collision counts on a port before partition and sets the LED update operation in the maximum mode.

The LERIC possesses control logic and interface pins which may be used to provide status information concerning activity on the attached network segments and the current status of repeater functions. On the LERIC-NIC board, 7 LED's are provided (one for each port and one for updating ''any'' port status). A '259 addressable latch is used to latch the data and address information contained in the D(7:0) pins of the LERIC (refer to the LERIC datasheet for the description of each pin). A toggle switch, SW1, is also provided to allow the display of either the status of the link integrity or reception activity on a per port basis and the status of collisions or jabber on the ''any'' port LED. This switch selects which LERIC data pin to use as data input to the '259 latch.

An on-board crystal oscillator provides the clock inputs for the NIC and LERIC. The oscillator's output is 20 MHz, which is fed directly to the LERIC and the NIC. Since the NIC also requires a 10 MHz clock, a flip-flop (74ALS74) is used to divide the 20 MHz down to 10 MHz. This is fed to the TXC input of the NIC.

The 10 MHz clock is also used as an input to the 74LS93 counter. Since there is no ''Ready'' signal from the LERIC indicating the completion of the data latching on a register read or write operation, this counter is used to generate a wait state before the I/OCHRDY signal is asserted back to the CPU.

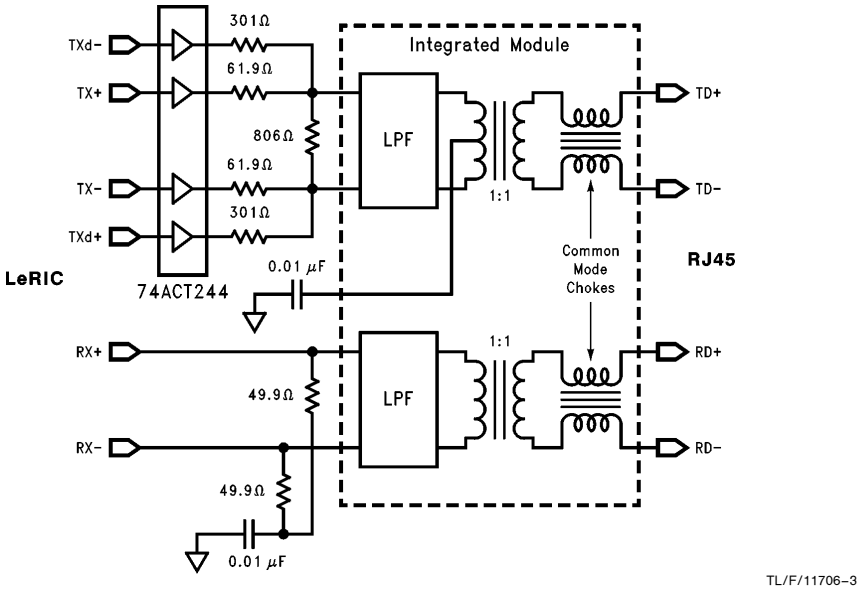


FIGURE 3. Twisted Pair Interface

7.1 Jumper Options

The default jumper block configurations are shown in *Figure 4*. On JB4, there are six possible connections. Four of these are to select an interrupt line. The available interrupt lines include INT3, INT4, INT5, and INT9. The last two possible connections, JP1 and JP0, are used to select the base address for the board. However, if JP7 is connected to V_{CC}, then these last two connections also select the address of the EPROM.

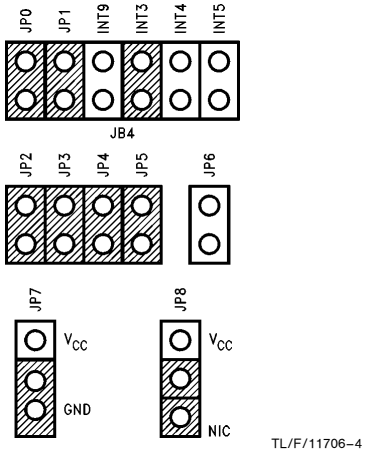


FIGURE 4. Jumper Blocks

The possible selections and the jumpers which should be ON (closed) are shown in Table VII. The factory configuration uses the INT3 line for interrupts and has JP1 and JP0 in the ON position. JP2 and JP3 are used to set the optional address space for the global register. The default position is ON for these jumpers (see Table III).

TABLE VII. Base and EPROM Addresses

JP1	JP0	Base Add	EPROM Add
ON	ON	300h–31Fh	C800h
ON	OFF	320h–33Fh	CC00h
OFF	ON	340h–35Fh	D000h
OFF	OFF	360h–37Fh	D400h

JP4 and JP5 are used to set the board address when multiple boards are cascaded together. These jumpers are normally ON in the default position. JP6 is used to take care of the IOCHRDY timing issues in some clone ATs. The default for this jumper is OFF (refer to PAL #2 description for more details on the IOCHRDY timing inconsistency). JP8 is used for arbitration when cascading several boards. The default position assumes that the NIC is on top of the arbitration chain, and ACKI is generated from PAL #4. When cascading several boards, the boards lower on the chain would have JP8 removed.

8.0 PAL EQUATIONS

PAL # (U1)

In this PAL, the output signals are NIO16, NIOEN, NNICB and SELREG.

NIO16 is only asserted if the CPU is trying to access the local RAM buffers and the board is in a 16-bit slot. NIO16 is used by the CPU to determine if it should perform 8- or 16-bit operations. If NIO16 is false, then the CPU will only perform 8-bit operations. Since it is necessary to assert NIO16 as soon as possible, this PAL has been selected to be a 10 ns "D" PAL. The NIO16 signal must be TRI-STATE when it is not asserted. Therefore, we use an enable signal (NIOEN) which is equal to the decode for the I/O Ports (310–31F) and NAEN high (NAEN high signifies that the system DMA does not have control of the bus). The enable signal (NIOEN) loops back into the PAL to bring NIO16 out of TRI-STATE. The NIO16 signal is set to zero so that whenever it is enabled it will be asserted.

The NNICB signal consists of simple address decodes along with NAEN, and will be asserted when the CPU wants to access the LERIC-NIC board. The addresses decode to one of four address slots which were earlier mentioned in the board configuration section.

The SELREG signal is similar to NNICB and decodes to one of four addresses which were mentioned in the board architecture section. SELREG is asserted when the CPU wants to access the Global Register.

PAL 1

```
begin header
date: 4/2/91
functions:
NIC/LERIC BOARD DECODE, IO16 DECODE, AND GLOBAL REGISTER DECODE';
end header

begin definition
  device gal16v8;
  inputs
    NEN16=1, NAEN=2, SA9=3,
    SA8=4, SA7=5, SA6=6,
    SA5=7, SA4=8, SA3=9,
    JP0=13, JP1=14, JP2=15, JP3=16;
  outputs (com)
    /NNICB=12, /NIO16=18, SELREG=19;
  feedbacks (com)
    NIOEN=17;
end definition

begin equations

NNICB =         (!NAEN & SA9 & SA8 & !SA7 & !SA6 & !SA5 & !JP1 & !JP0
                # !NAEN & SA9 & SA8 & !SA7 & !SA6 & SA5 & !JP1 & JP0
                # !NAEN & SA9 & SA8 & !SA7 & SA6 & !SA5 & JP1 & !JP0
                # !NAEN & SA9 & SA8 & !SA7 & SA6 & SA5 & JP1 & JP0);

SELREG =        (!NAEN & SA9 & !SA8 & !SA7 & !SA6 & !SA5 & !JP3 & !JP2
                # !NAEN & SA9 & !SA8 & !SA7 & !SA6 & SA5 & !JP3 & JP2
                # !NAEN & SA9 & !SA8 & !SA7 & SA6 & !SA5 & JP3 & !JP2
                # !NAEN & SA9 & !SA8 & !SA7 & SA6 & SA5 & JP3 & JP2);

NIOEN =         (!NAEN & SA9 & SA8 & !SA7 & !SA6 & !SA5 & !JP1 & !JP0
                        & !NEN16 & SA4 & !SA3
                # !NAEN & SA9 & SA8 & !SA7 & !SA6 & SA5 & !JP1 & JP0
                        & !NEN16 & SA4 & !SA3
                # !NAEN & SA9 & SA8 & !SA7 & SA6 & !SA5 & JP1 & !JP0
                        & !NEN16 & SA4 & !SA3
                # !NAEN & SA9 & SA8 & !SA7 & SA6 & SA5 & JP1 & JP0
                        & !NEN16 & SA4 & !SA3);

NIO16.oe = NIOEN;

NIO16 = 1;

end equations
```

TL/F/11706–5

PAL # (U2)

In this PAL, there are seven outputs which include NRESET, NIOCHR, NIOCHW, NIOCHC, NCS, NRACK, and NWACK.

The IOCHRDY signal is used to wait state the CPU. Normally an I/O card drives IOCHRDY low (not ready) only *after* the address and I/O read or write signals have been asserted. On some PC-AT compatible PCs (those using Chips and Technologies or VLSI Inc. chipsets), during a 16-bit I/O operation, the bus controller actually samples the IOCHRDY signal *before* the I/O read or write signal is asserted.

NIOCHC is used to drive IOCHRDY low (not ready) based only on an address decode, thus allowing IOCHRDY to be asserted earlier and in the proper state when it is sampled by the bus controller.

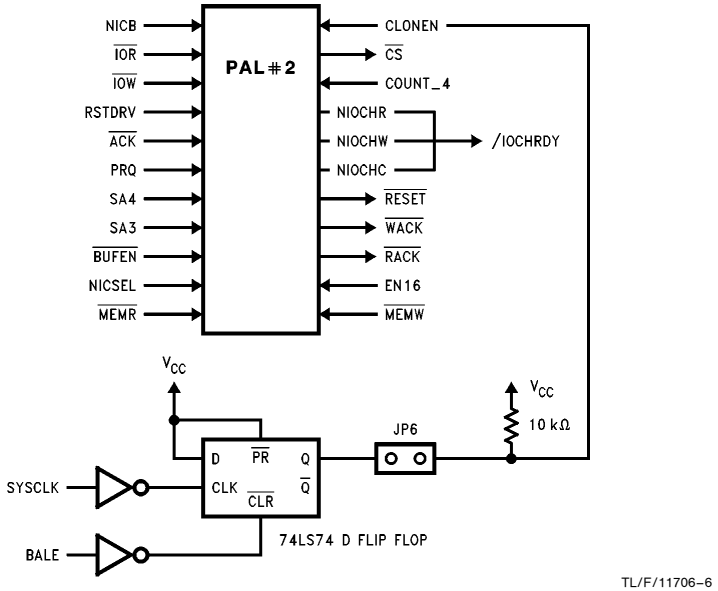
As shown in *Figure 5*, NIOCHR, NIOCHW, and NIOCHC are all externally wire-ORed together to generate the IOCHRDY signal. NIOCHR along with NIOCHR.OE (enable NIOCHR) are for slave read and remote read cycles. NIOCHW and NIOCHW.OE (enable NIOCHW) are for slave write and remote write cycles. These signals together generate the ''normal'' IOCHRDY signal.

For more details on the IOCHRDY fix, refer to the application note *PC-AT Interface Design Considerations for the DP83902EB-AT*. NIOCHC may cause problems on PC's that do not use Chips and Technologies or VLSI Inc. chipsets. A D-flip flop is used to generate CLONEN to implement the timing modification. The variable CLONEN is tied to a jumper that is used to switch the IOCHRDY signal characteristics to ''normal'' or modified timing. If the system operates under ''normal'' timing characteristics, JP6 should be removed.

NCS is asserted by the CPU when it needs to access the NIC registers. NCS is decoded in the address range of 300–30F along with NICSEL from PAL #5. NICSEL is set high to access NIC registers and low to access LERIC registers.

The next two signals, NRACK and NWACK, are used to acknowledge successful transfers between the CPU and the NIC data buffers. The NRACK occurs with an address decode to 310–317, an NIOR, and a PRQ. NWACK occurs with an address decode to 310–317, an NIOW, and a PRQ.

The last signal is NRESET, which is used to reset both the NIC and LERIC configuration registers. NRESET can be asserted by the system with RSTDRV or through software after the system has booted up. Once NRESET has been asserted, it is held low until IOW is received from the CPU. This will guarantee that NRESET has the pulse width required by the NIC.



TL/F/11706–6

FIGURE 5

PAL 2

```
begin header
date: 4/2/92
functions:
RESET LATCH, NIC SELECT, IOCHRDY, RACK, WACK ';
end header

begin definition

device PAL20L8;

inputs
NNICB=1, NIOW=2, NIOR=3,
RSTDRV=4, NACK=5, PRQ=6,
SA4=7, SA3=8, NBUFEN=9, NICSEL=10,
NMEMR=11, NMEMW=13, EN16=14, COUNT_4=21, CLONEN=23;

outputs(com)
!NRACK=15, !NWACK=16,
!NIOCHR=18, !NIOCHW=19,
!NIOCHC=20, !NCS=22;

feedbacks(com)
!NRESET=17;

end definition

begin equations

NCS =   (!NNICB & !NIOR & !SA4 & NICSEL
                # !NNICB & !NIOW & !SA4 & NICSEL);
NRACK =         (!NNICB & PRQ & !NIOR & SA4 & !SA3 & NICSEL);
NWACK =         (!NNICB & PRQ & !NIOW & SA4 & !SA3 & NICSEL);
NIOCHR= (!PRQ & SA4 & !SA3 & NICSEL
                # NACK & !SA4 & NICSEL
                # !NICSEL & !NBUFEN & !COUNT_4);
NIOCHR.oe=  (!NNICB & !NIOR);
NIOCHW= (!PRQ & SA4 & !SA3 & NICSEL
                # NACK & !SA4 & NICSEL
                # !NICSEL & !NBUFEN & !COUNT_4);
NIOCHW.oe=  (!NNICB & !NIOW);
NIOCHC=1;
NIOCHC.oe=  (!NNICB & NMEMR & NMEMW & !PRQ & !EN16 &
                        !CLONEN & SA4 & !SA3);
NRESET =       (!NNICB & !NIOR & SA4 & SA3 # NIOW & NRESET # RSTDRV);

end equations
```

TL/F/11706–7

PAL #3 (U16)

The third PAL does a decode to enable the optional EPROM. This decode consists of an address decode to C8000h, CC000h, D0000h, or D4000h depending on JP1 and JP0 as shown in the board configuration section. JP7 must also be jumpered for selection of the EPROM. NAEN, a low asserted signal should be low to indicate that the DMA does not have control of the bus and the NSMRDC signal should be asserted low since the CPU is doing a system memory read.

The A013 signal is used for 8-bit mode operation. When the board is placed into an 8-bit slot the EN16 signal is used to detect the existence of the second PC-AT bus connector. If EN16 is low then the board is in a 16-bit slot. In this condition, A13 (from the NIC) is enabled to A013 which goes to the LSB RAM. If the board is placed into an 8-bit slot then EN16 is pulled high by a resistor, and this causes the LERIC-NIC's A0 signal to be enabled. This allows the LSB RAM to be used as an 8-bit RAM.

PAL 3

```
begin header
date: 4/2/92
function:
EPROM DECODE';
end header

begin definition

device pal16l8;

inputs
A0=1, A13=2, EN16=3,
NSMRDC=4, SA19=5, SA18=6,
SA17=7, SA16=8, SA15=9,
SA14=11, NAEN=13, JP2=14,
JP0=15, JP1=16;

outputs(com)

!NEPROMEN=19, !A013=12;

{Note: I/O pins 17 and 18 are not being used.}

end definition

begin equations

NEPROMEN =      (SA19 & SA18 & !SA17 & !SA16 & SA15 & !SA14 & !NAEN
                        & JP2 & !JP1 & !JP0 & !NSMRDC
                # SA19 & SA18 & !SA17 & !SA16 & SA15 & SA14 & !NAEN
                        & JP2 & !JP1 & JP0 & !NSMRDC
                # SA19 & SA18 & !SA17 & SA16 & !SA15 & !SA14 & !NAEN
                        & JP2 & JP1 & !JP0 & !NSMRDC
                # SA19 & SA18 & !SA17 & SA16 & !SA15 & SA14 & !NAEN
                        & JP2 & JP1 & JP0 & !NSMRDC);

A013 =  (!A0 & EN16 # !A13 & !EN16);

end equations
```

TL/F/11706–8

PAL #4 (U30)

In this PAL there are seven outputs which include CRS, COL, TXENABLE, NLRD, NLWR, NLERICEN, and GRDATA.

The first three outputs generate the interface signals between the LERIC and NIC. CRS is just an inverted NIRE signal which notifies the NIC that there is activity on the network, and COL is asserted if there is a receive or transmit collision coming from the LERIC. TXENABLE is tied to $\overline{\text{ACKI}}$, putting the NIC at the top of the LERIC's arbitration chain. TXENABLE is also used to enable the 244 buffer that controls the flow of $\overline{\text{TXE}}$, TXC, and TXD coming from the NIC.

NLRD, and NLWR generate the read and write strobes to the LERIC based on an address decode, $\overline{\text{IOR}}$ or $\overline{\text{IOW}}$, and LERICHIT. LERICHIT is the Global Register decode from PAL #5.

The NLERICEN signal is used to enable the TRI-STATE buffers and receivers that access the LERIC registers. It will only be asserted after NLRD or NLWR are true and the LERIC has driven $\overline{\text{BUFEN}}$ low, signifying that the inter-LERIC BUS is available to do a register read or write.

The last output, GRDATA, is used to latch the Global Register bits into PAL #5. The data bits from the AT-BUS will be valid after $\overline{\text{IOW}}$ is asserted low along with SELREG, which is an address decode for the Global Register coming from PAL #1.

PAL 4

```
begin header
date: 4/2/92
function:
LERIC READ, LERIC WRITE, LERIC MLOAD, CRS, Global Register Data;
end header

begin definition
device pal16l8;

inputs

NIOR=1, NIOW=2, SELREG=3, LERICHIT=4,
NBUFEN=5, NANYXN=6, NCOLN=7, NIRE=8,
SA4=9, NICB=11, TXE=16;

outputs(com)
!NLWR=12, !NLRD=13,!TXENABLE=14, !NLERICEN=15,
!CRS=17, !COL=18,!GRDATA=19;

end definition

begin equations

CRS =          NIRE ;
!COL =          !NCOLN # !NANYXN;

TXENABLE =     TXE;

NLRD =  (!NICB & !NIOR & !SA4 & LERICHIT ) ;
NLWR =  (!NICB & !NIOW & !SA4 & LERICHIT ) ;

NLERICEN =    (!NICB & !NIOR & !SA4 & LERICHIT & !NBUFEN
                # !NICB & !NIOW & !SA4 & LERICHIT & !NBUFEN);

!GRDATA =       SELREG & !NIOW;

end equations
```

TL/F/11706–9

PAL #5 (U29)

This GAL includes the outputs NCSROM, INTO, LERICHIT and NICSEL. The two registered outputs ID0 and ID1 are only used internally by the GAL.

The LERICHIT signal comes from the decode of the Global Register, and indicates that the CPU needs to access the LERIC.

GRDATA is used to clock in the registered outputs consisting of ID0, ID1, and NICSEL. These three bits store the contents of the Global Register.

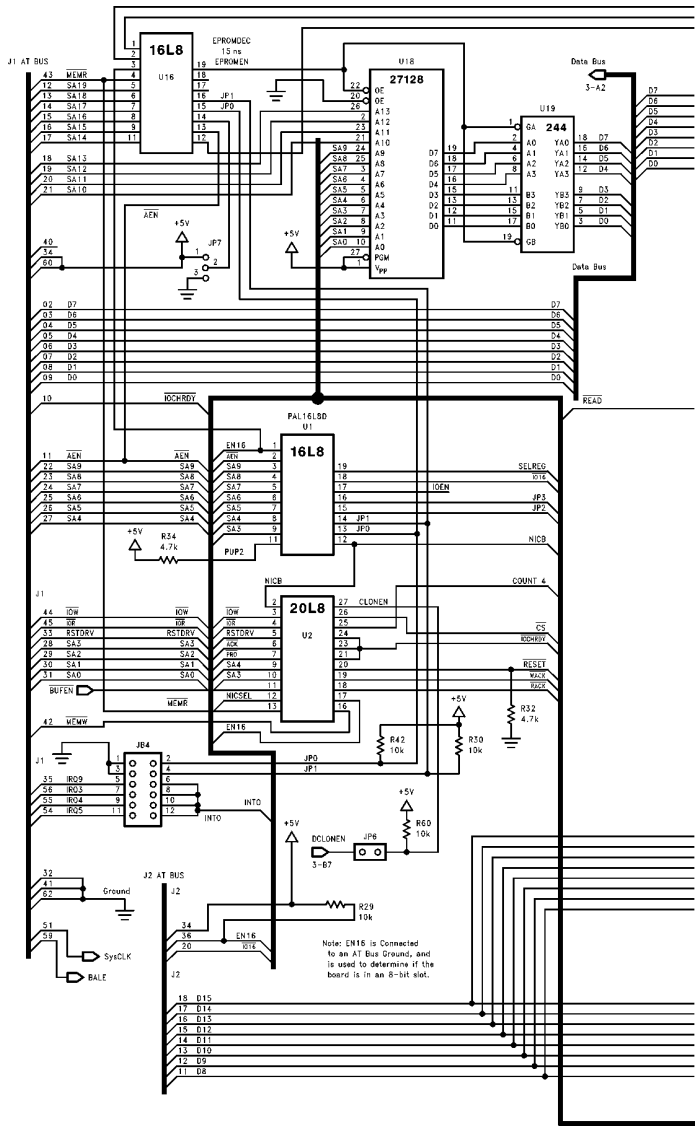
INT is just sent through the GAL to be buffered. The buffered signal which comes out of the GAL is INTO. The NCSROM is a very simple signal as it consists only of AD14 and NMRD. AD14 comes from the NIC and selects either the PROM (when low) or the on-board RAM (when high).

PAL 5

```
begin header
date: 8/7/92
function:
LERICHIT, NICSEL, INTO, NCSROM';
end header

begin definition

device gal16v8;

inputs

GRDATA=1, JP4=2, JP5=3,
D2=4, D1=5, D0=6, NIOR=7,
A14=8, NMRD=9, INT=18;

outputs(com)
!NCSROM=12, INTO=13,
LERICHIT=19, NREAD=14;

feedbacks(reg)
ID0=15, ID1=16, NICSEL=17;

end definition

begin equations

INTO = INT;
NREAD= NIOR;

ID0 := D0;
ID1 := D1;

NICSEL := D2;

NCSROM =  ( !A14 & !NMRD);

LERICHIT =     (ID0 & ID1 & JP4 & JP5 & !NICSEL
                # !ID0 & !JP4 & ID1 & JP5 & !NICSEL
                # ID0 & JP4 & !ID1 & !JP5 & !NICSEL
                # !ID0 & !JP4 & !ID1 & !JP5 & !NICSEL);

end equations
```

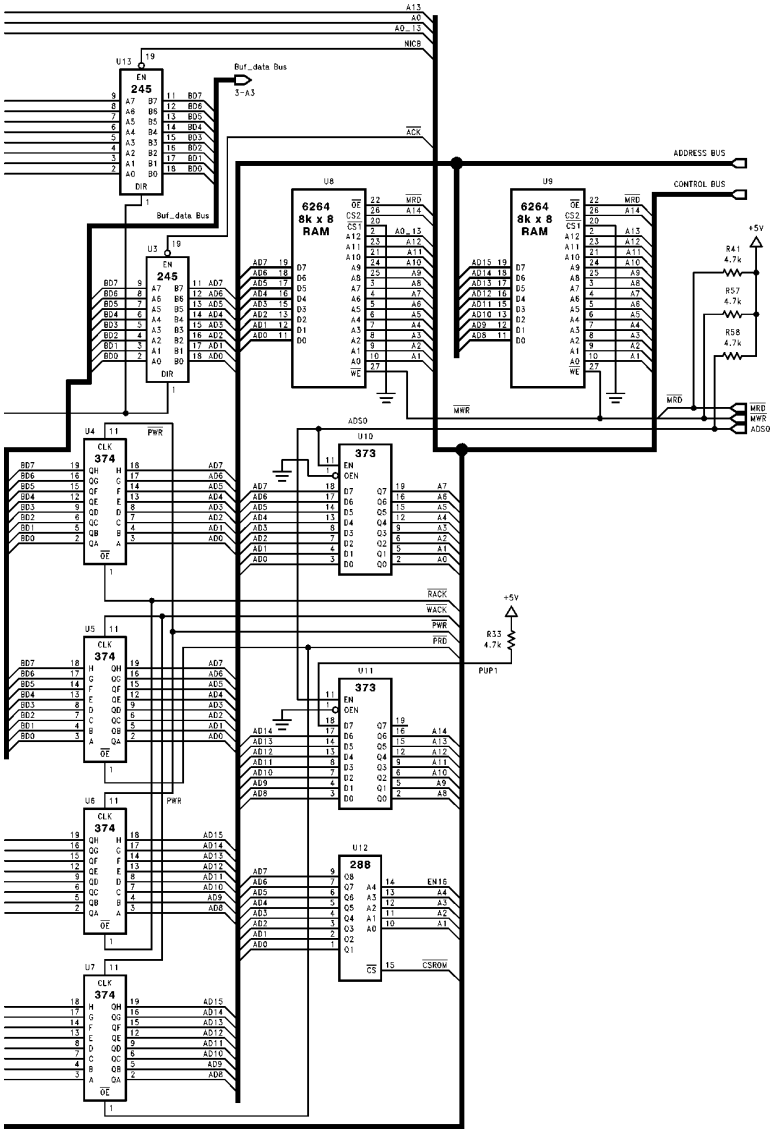
TL/F/11706–10

LERIC/NIC PC-AT Evaluation Board



TL/F/11706–11

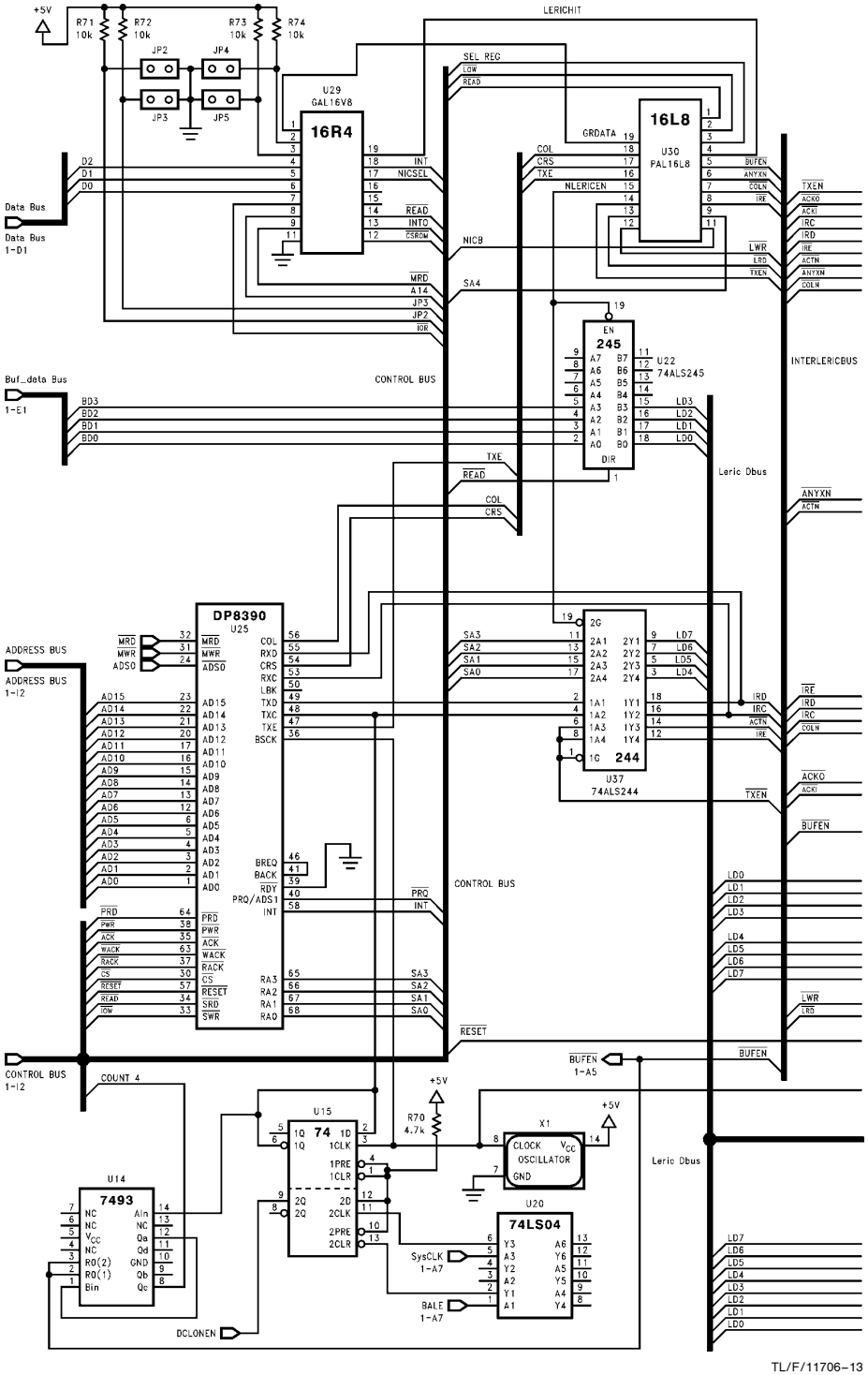
LERIC/NIC PC-AT Evaluation Board (Continued)



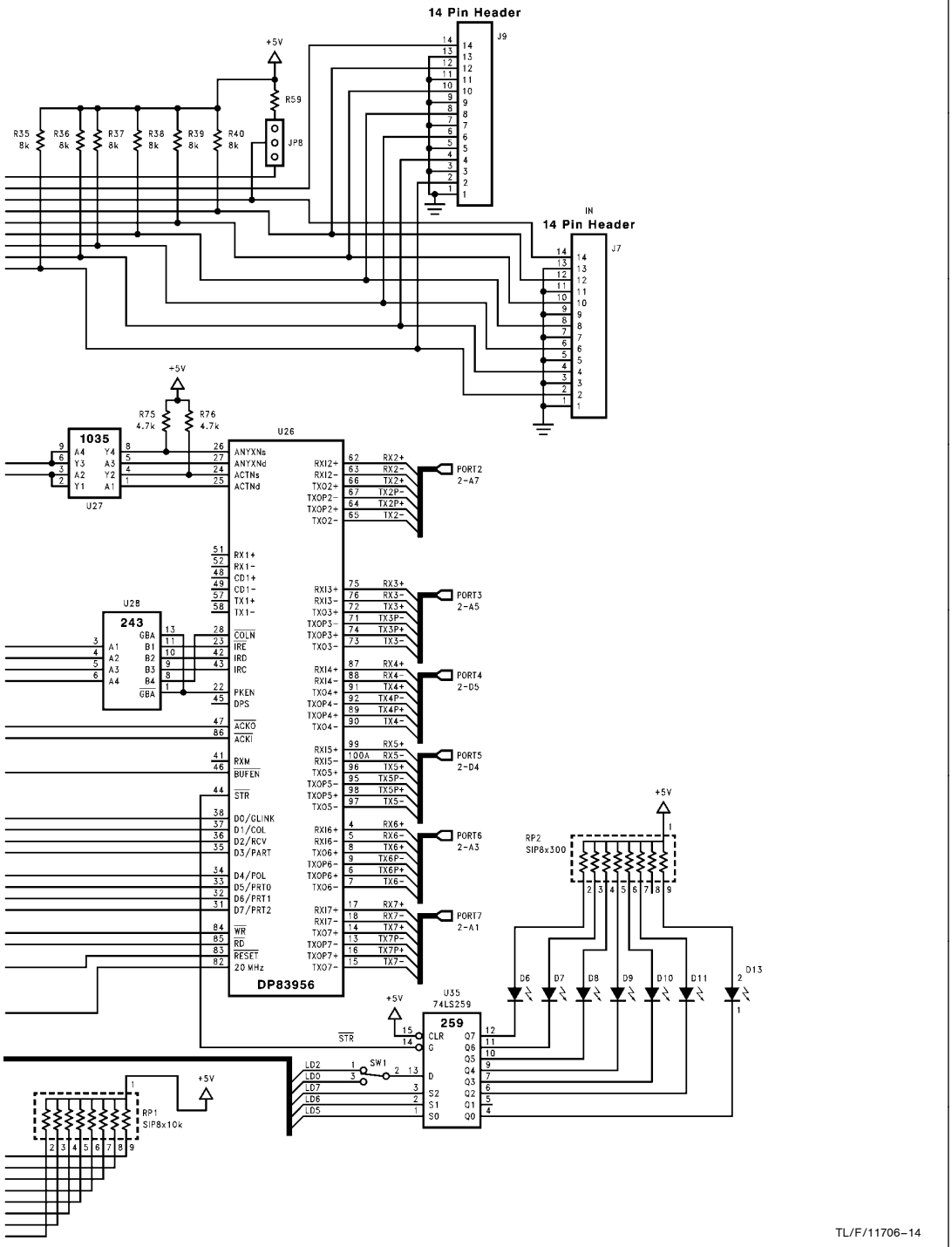
TL/F/11706–12

AT Pins	Assigned Pins
A2	02
A3	03
A4	04
A5	05
A6	06
A7	07
A8	08
A9	09
A10	10
A11	11
A12	12
A13	13
A14	14
A15	15
A16	16
A17	17
A18	18
A19	19
A20	20
A21	21
A22	22
A23	23
A24	24
A25	25
A26	26
A28	28
A27	27
A29	29
A30	30
A31	31
B1	32
B3	34
B4	35
B9	40
B10	41
B13	44
B14	45
B23	54
B24	55
B25	56
B29	60
B31	62
C11	11
C12	12
C13	13
C14	14
C15	15
C16	16
C17	17
C18	18
D2	20
D16	34
D18	36

LERIC/NIC PC-AT Evaluation Board (Continued)



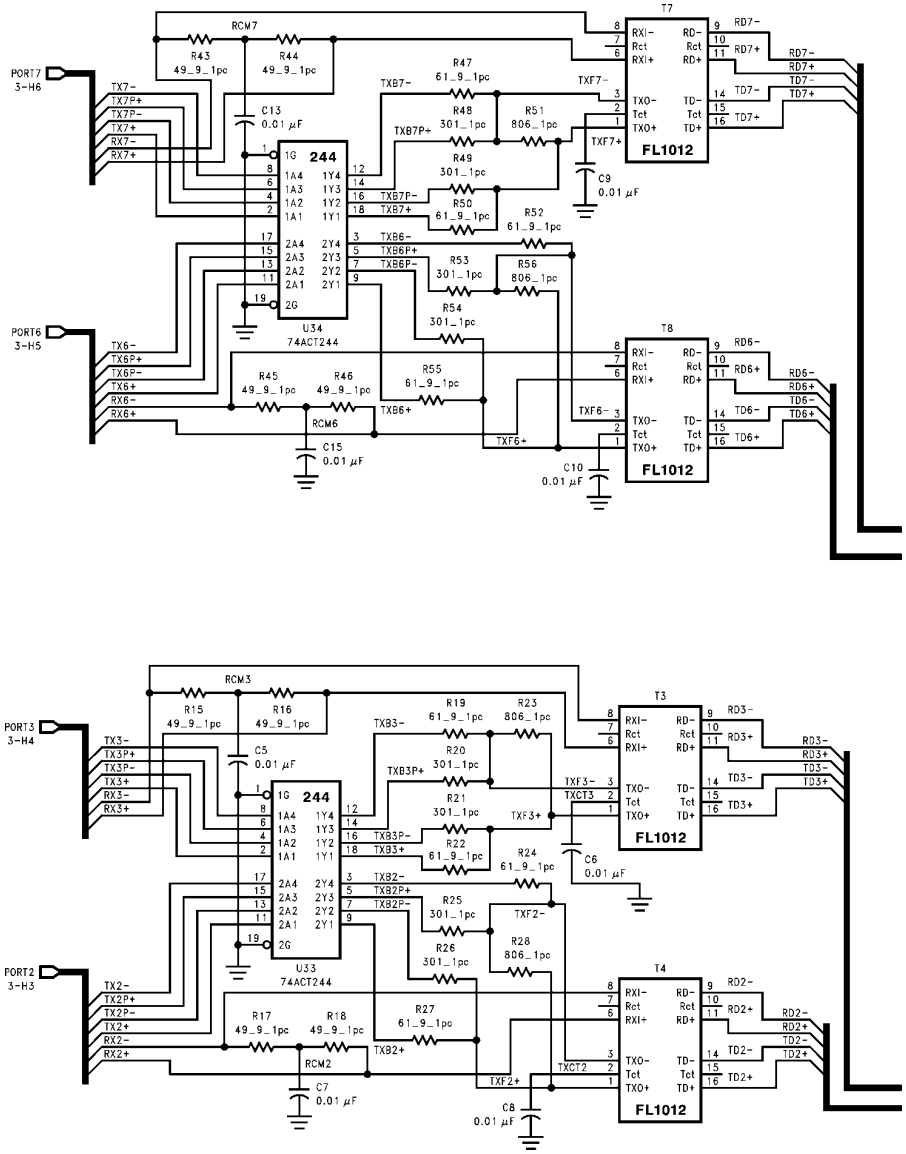
LERIC/NIC PC-AT Evaluation Board (Continued)



TL/F/11706–14

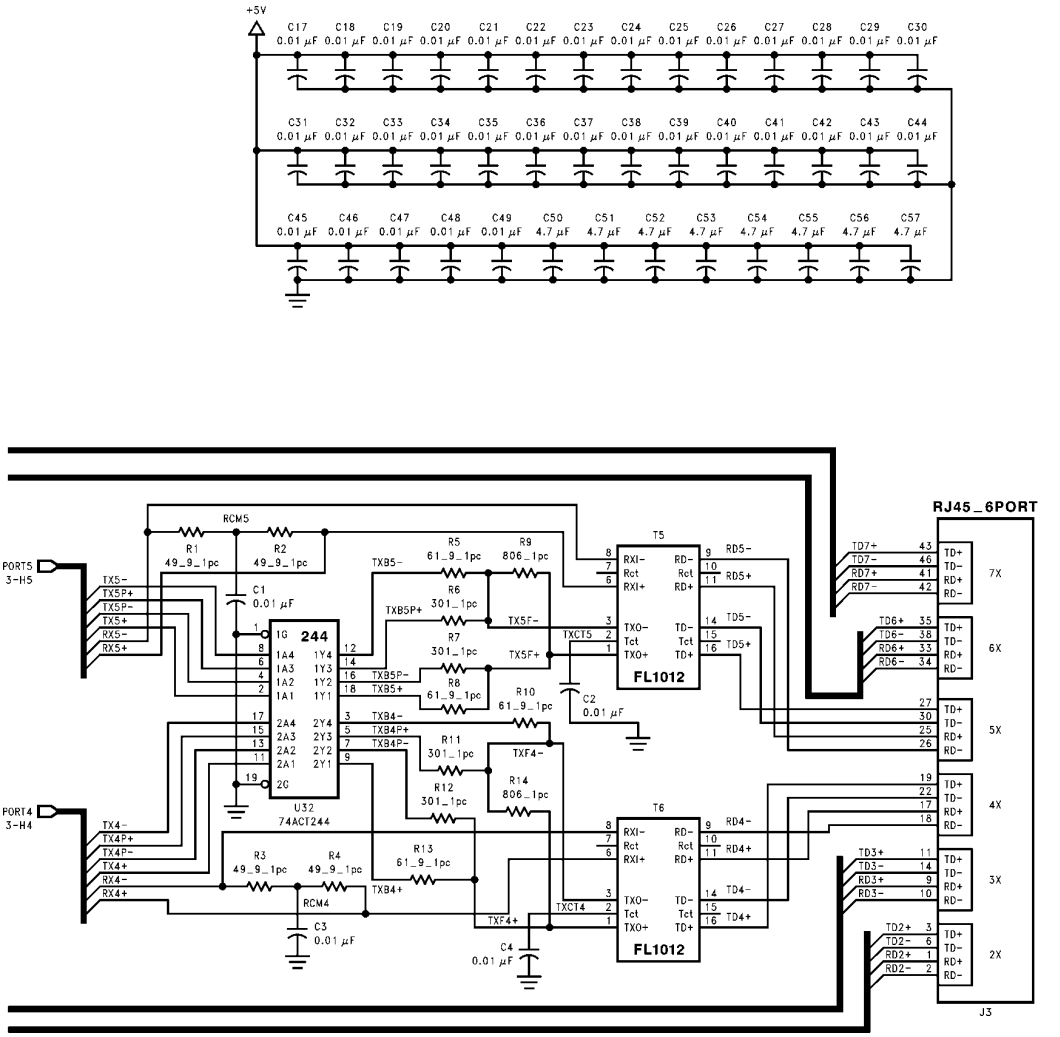
Note: All resistors to be 5%, 1/4W unless otherwise indicated.

LERIC/NIC PC-AT Evaluation Board (Continued)



TL/F/11706–15

LERIC/NIC PC-AT Evaluation Board (Continued)



TL/F/11706–16

Note: Resistor Nomenclature is as follows:

49__9__1pc = 49.9Ω, 1%

61__9__1pc = 61.9Ω, 1%

301__1pc = 301Ω, 1%

806__1pc = 806Ω, 1%

9.0 BILL OF MATERIALS

Capacitors

0.01 μF			C1..C10, C13, C15, C17, C18....C49	
4.7 μF			C50..C57	

Miscellaneous

*0.5mmGreenLED			D6..D11, D13	
*SS-668806-NF			J7	(6PORT, 8 POSITION, RJ45 FROM STEWART)
*3__pin__jumper			JP7, JP8	
*2__pin__jumper			JP2..JP6	
*jumpers6x2			JP4	(jumper block)
*TOGGLESWITCH__DPST			SW1	
*FL1012			T3..T8	(VALOR)
*102160-2			J7, J9	AMP PIN HEADER
746194-2			J6, J8	AMP RECEPTACLES

Resistors

8k	5%	¼W	R35..R40
10k	5%	¼W	R29, R30, R42, R60, R71..R74
49.9	1%	¼W	R1..R4, R15..R18, R43..R46
4.7k	5%	¼W	R32, R33, R34, R41, R57, R58, R59, R70, R75, R76
61.9	1%	¼W	R5, R8, R10, R13, R19, R22, R24, R27, R47, R50, R52, R55
301	1%	¼W	R6, R7, R11, R12, R20, R21, R25, R26, R48, R49, R53, R54
806	1%	¼W	R9, R14, R23, R28, R51, R56
*SIP8x300	5%	¼W	RP2
*SIP8x10k	5%	¼W	RP1

IC's

PAL16L8 (15 ns)		U16, U30	PLCC Type (SOCKETED)
GAL16V8 (10 ns)		U1	PLCC Type (SOCKETED)
GAL16V8 (15 ns)		U29	PLCC Type (SOCKETED)
PAL20L8D (10 ns)		U2	PLCC Type (SOCKETED)
74ALS245		U3, U22, U13	
74ALS374		U4, U5, U6, U7	
6264RAM		U8, U9	8k x 8 STATIC RAM 100 ns
74ALS373		U10, U11	
*74S288		U12	SOCKETED
74LS93		U14	
*27128		U18	EPROM (SOCKET ONLY)
74ALS244		U19, U37	
*20 MHz		X1 (0.01%)	Crystal Oscillator
74LS04		U20	
74ALS243		U28	
74ALS1035		U27	
DP8390		U25	**NIC** (SOCKETED PLCC)
74ALS74		U26	
74ACT244		U32, U33, U34	
74LS259		U35	
DP83956		U36	**LERIC**

__Note:__ Everything is surface mount except the devices that were marked with an ''''. devices with an ''*'' are through hole.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408