Am79C960 PCnet-ISA™

Technical Manual

May 1992

A D V A N C E D   M I C R O   D E V I C E S

Table of Contents

Introduction

This section provided an overview of IEEE 802.3/Ethernet. It is not intended to be an exhaustive study of the technology, but rather a primer for anyone not familiar with the concepts of an Ethernet or IEEE 802.3-compatible network. Those already familiar with 802.3/Ethernet will probably want to skip this section.

This tutorial information is included within the technical manual for the PCnet-ISA™ chip because of the nature of the device itself. The PCnet-ISA chip is effectively an integrated Ethernet adapter card intended primarily for PC/AT-compatible Industry Standard Architecture (ISA) bus systems. Because it integrates all of the fixed logic for an entire adapter, the PCnet-ISA chip offers an extremely simple solution for Ethernet connectivity on a personal computer motherboard, as well as a very-low-cost adapter. As few PCs are currently shipped with Ethernet installed on the motherboard, this section will assist designers in gaining an understanding of this network technology.

The section also provides a brief historical perspective of Ethernet, clarifies the terminology used, and identifies the key advantages of 10BASE-T.

10BASE-T TECHNOLOGY BENEFITS

Topology

Traditional Ethernet (802.3, 10BASE5)[1] and Cheapernet (802.3, 10BASE2)[2] are coaxial-wired systems. Coaxial cable provides the linear bus to which all nodes are connected. Signaling uses a current sink technique, with the center conductor used for signal and the shield as ground reference. See Figures 1 and 2.
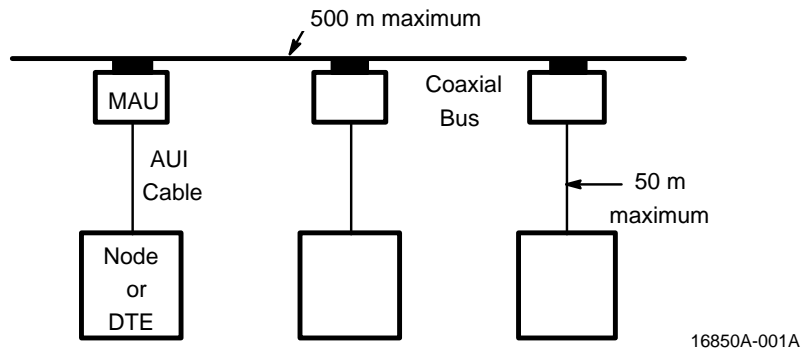
Figure 1. 10BASE5 Coaxial Bus Topology
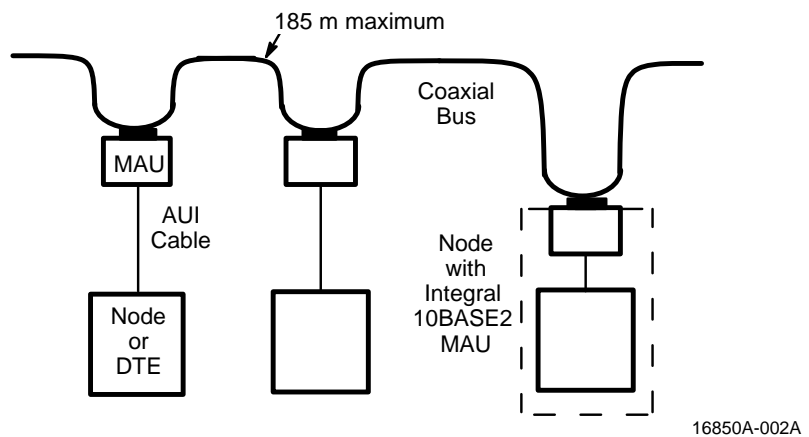
Figure 2. 10BASE2 Coaxial Bus Topology

Twisted-Pair Ethernet (802.3 10BASE-T)[3] uses standard voice-grade telephone cable (22–26 gauge) that employs separate transmit and receive pairs (4 wires). The system uses a star topology. At the center of the star (Figure 3) is a repeater. The repeater (or hub) performs signal amplitude and timing restoration. It takes the incoming bit stream and repeats it to all other ports connected to it (but not back to the originating port). In this sense, the repeater acts as "logical coax," so that any node connected to the network will see another's transmission. Differential signaling is employed, with one pair acting as the transmit path and the other as receive.

---

[1] ISO/IEC 8802-3 :1990 (E); ANSI/IEEE Std 802.3-1990 Edition, Section 8.

[2] ISO/IEC 8802-3 :1990 (E); ANSI/IEEE Std 802.3-1990 Edition, Section 10.

[3] ISO/IEC 8802-3 :1990 (E); ANSI/IEEE Std 802.3-1990 Edition, Section 13/14.

16850A-003A
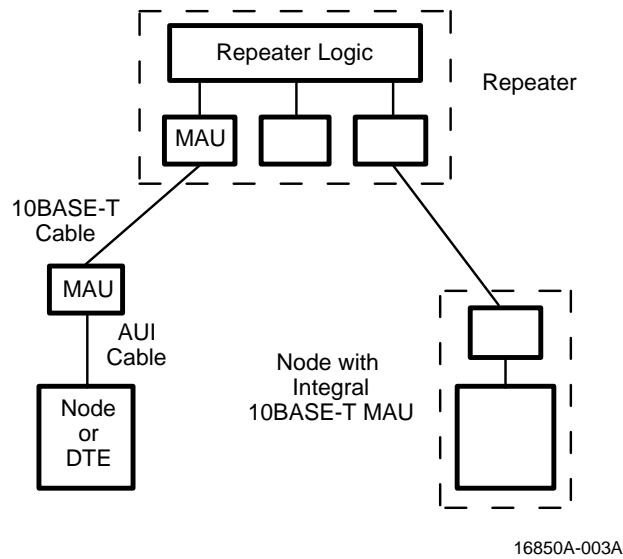
Figure 3. 10BASE-T Twisted-Pair Star Topology

Collision Detection

In 802.3-based networks, all devices are permitted to listen to the channel, but only one may transmit at any given time. If two or more devices transmit simultaneously, a "collision" is sensed, and the nodes involved are forced to reschedule their transmissions after a random interval.

The Medium Attachment Unit (MAU) is responsible for the detection of collisions. Current MAU implementations consist largely of a transceiver, with additional passive components, power supply, and connectors (see Figures 4 and 5). Serial data originating from the LAN controller in the Data Terminal Equipment (DTE) are passed to the MAU using the Attachment Unit Interface (AUI). If the MAU detects a collision, it reports back to the controller using the AUI.

With a coaxial topology, all nodes are connected to the center cable conductor. The transceiver can detect two or more devices transmitting on the network because the voltage on the center conductor will exceed a collision threshold (–1.6 V nominally). To drive the coax cable, the transceiver requires a relatively high negative supply voltage (typically –9 V), making it an unsuitable candidate for most mainstream CMOS silicon processes currently available.
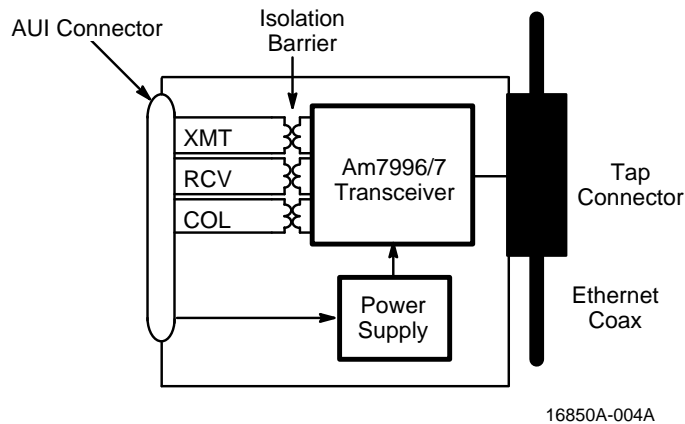
16850A-004A

Figure 4. 10BASE5 MAU
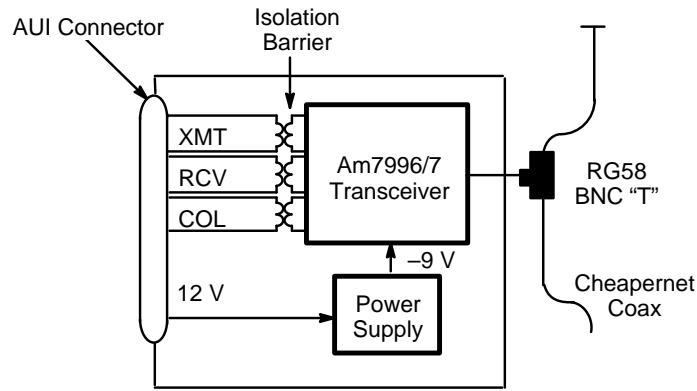
16850A-005A

Figure 5. 10BASE2 MAU

Because 10BASE-T uses separate transmit and receive signal paths, "logical" collision detection is implemented. While data are transmitted from the node to the repeater over the transmit wire pair, the receive pair should remain idle. If both the transmit and receive pairs become active simultaneously, the 10BASE-T transceiver detects a collision. The voltage levels required by 10BASE-T (5.0 V ±0.6 V peak-to-peak) can be met using standard 5 V CMOS logic levels.
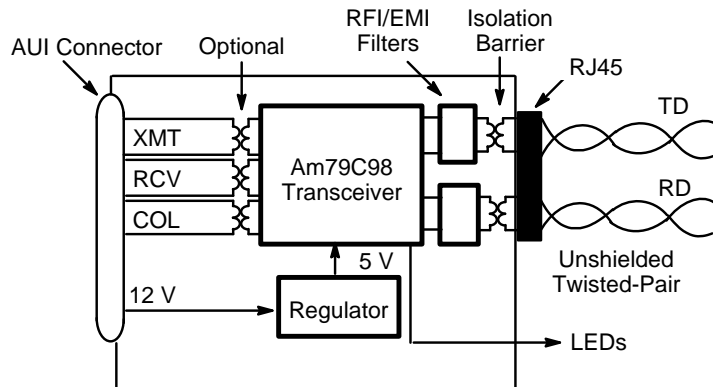
16850A-006A

Figure 6. 10BASE-T MAU

Isolation

An isolation barrier is required in all long-distance network topologies to protect the DTE from potentially hazardous voltages that may be present during fault conditions on the medium.

In a coax-based network, this direct current (DC) isolation is located in the AUI path because the coax transceiver must be DC-coupled to the center conductor to permit collision detection.

With 10BASE-T the logical collision detection scheme eliminates the need for a DC path to the medium (see Figures 5 and 6). The isolation barrier can be relocated to the medium side of the 10BASE-T transceiver. Removing the requirement for isolation in the AUI path allows the 10BASE-T transceiver to be integrated with the remainder of the 802.3 node components, i.e., Medium Access Controller (MAC) and Physical Layer Signaling (PLS), as a single piece of silicon (see Figure 7).
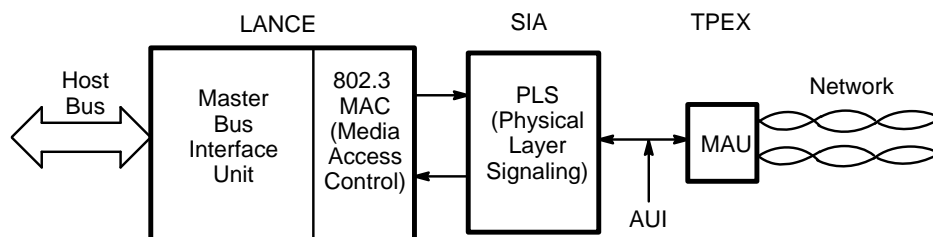
16850A-007A

Figure 7. Ethernet Node Silicon

Link Integrity

Separation of the transmit and receive paths in 10BASE-T has two potential drawbacks.

When the node transmits in a coax system, the MAU simultaneously receives the transmission (because both the transmitter and receiver are connected to the coax center tap) and returns it to the controller as an indication of transmit-to-receive path integrity.

In a 10BASE-T system, separation of the transmit and receive cable pairs combined with logical collision detection means that when driving the transmit twisted-pair, the transceiver does not see activity on the receive pair unless a collision occurs. Unlike the coax system, the transceiver can no longer observe its own signal transmission, so the loopback path to the controller is implemented internally to the MAU. The controller is made to believe that the transmit-to-receive integrity is present, and no difference is detected between a coax or a twisted-pair medium. However, a mechanism is necessary to ensure that a failure in the transmit or receive path is detected.

In the case of a broken transmit path, the node is unable to send data over the network. However, a broken receiver (or receive cable) has far more serious implications. The node loses its ability to monitor the network for activity or collisions. A node with data to transmit would do so regardless of current network activity, possibly causing a collision with an existing message.

Because recovery from a collision is a fundamental of the 802.3 Medium Access Control function, it appears that this is not a serious problem. However, in a correctly configured network, collisions are guaranteed to occur within a defined window (known as "slot time") after transmission commences. A transmission that experiences a collision within the slot time is automatically retried by the sending node. Up to 15 re-tries (16 total attempts) are permitted before the node aborts the transmission. On the other hand collision that occurs after the slot time (512 bits or 51.2 μs) results in a late collision, and the node abandons the transmission attempt immediately. Most 802.3 LAN controllers incorporate a late collision indicator to advise the host processor of this condition. Upper-layer software then has the responsibility to recognize a late collision indicator and reschedule the transmission.

A feature referred to as "link test" ensures network integrity. In the absence of network traffic, a simple heartbeat pulse is sent by the transmitter of the 10BASE-T MAU (at the repeater or DTE). If the MAU receiver does not see either packet data or a link test pulse within a defined time window (16 ms ±8 ms), it enters a "link fail" condition, thus disabling the data transmit, data receive, and loopback functions (see Figure 8). Disabling the transmit function prevents the disturbance of existing network traffic. Disabling the loopback path warns the node that there is a failure (the transmit-to-receive path is interrupted). If the receive pair is disconnected, the MAU enters the link fail state, and further transmission is disabled. During link fail, the transmission and reception of link test pulses continues. To reestablish the link, at least two consecutive link test pulses or a single receive packet must be received.
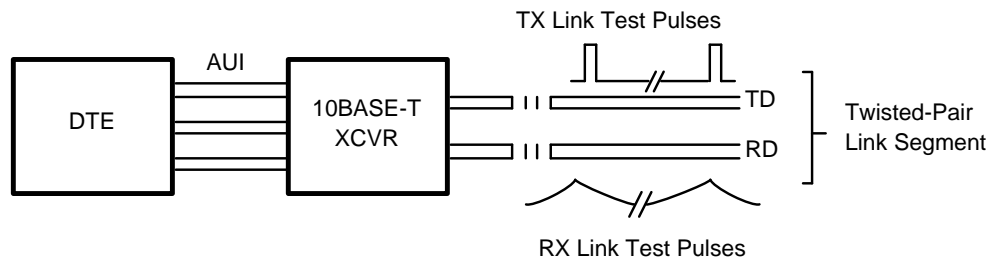
16850A-008A

Figure 8. Link Test Transmission/Reception

Status Indicators

Most 10BASE-T transceivers incorporate additional status features to allow simple diagnoses of the network or node state.

The external link status indicator is mandated by the 10BASE-T standard. Transmit, receive, and collision activity can be separately indicated (in the coax MAU, receive is always active during transmit) to provide a simple network activity display. In addition, because the link test pulse (unlike normal differential data transmissions) is unipolar, it is possible to detect the polarity of the receive signal path. This is a useful feature that can be used to automatically detect and correct simple wiring installation errors.

Key Advantages

For the user community, 10BASE-T brings several key advantages.

Reduced Installation Cost

LAN cable installation costs far more than the cable itself. If existing telephone cabling is used, savings are significant. Even if new cabling must be installed, costs are generally lower because identical connection technology is widely used in telecom applications. Therefore, it is inexpensive, and installation is easily procured.

Long-term Cost of Ownership (COO) Benefits

During the lifetime of a LAN, the long-term COO benefits may far outweigh the original equipment cost. Addition or movement of users is more difficult in coax-based Ethernet. Connection to the coax cable requires special tools, and the coax bus may not be physically accessible. A 10BASE-T system is literally "plug and play." Reconfiguration is as simple as adding a new wall plug or plugging into a pre-wired connection from the repeater. When the user moves (switches off the system), link test pulses stop being transmitted, and the repeater port effectively shuts down.

Ease of Fault Isolation, Management, and Security

In large corporate networks, the ability to manage and maintain the network is vital. Many businesses depend on their communications facilities. The LAN is a utility much like the telephone system. When the telephone is picked up, a dial tone is expected. In the same way, when networked service is accessed, it is expected to be available.

On a coax bus, faults are difficult to analyze because all nodes are connected to the cable at all times. With 10BASE-T, only one DTE (or repeater) is connected to any port on a repeater, so the behavior of any connection can be individually monitored. A failure can be isolated quickly, and the remainder of the network operates unimpaired while the problem is fixed.

Access control and network security is readily administered through the repeater. For instance, the manager can instruct the hub to shut down a particular link at 5 p.m. on Friday and enable the link at 8 a.m. on Monday. Other features such as configuration mapping can also be monitored at the hub because each 10BASE-T link is generally

connected to a single node. Connections to repeaters and coax segments do not obey this rule, but this behavior also allows these links to be identified.

Even in small or low-cost installations in which a network manager is not warranted, the single point of concentration, point-to-point connectivity, and rudimentary status indicators make diagnosing a problem on a 10BASE-T network a less-daunting task than checking the entire coax segment for a potential connection problem.

Volume Manufacturing

The 10BASE-T topology enables further silicon integration, which will ultimately drive down the system cost in two ways. First, the high demand for integrated circuits (ICs) and board level products will generate fierce competition for market share from both silicon and systems companies. Second, the sheer size of the market opportunity will lead to high volume manufacturing.

Interoperability and Standardization

The overwhelming demand from the user community is for interoperability based on open standards.

A standard, in itself 10BASE-T, is driving the creation and adoption of additional standardization. The lack of a centrally located point for monitoring and network management has long been a criticism leveled at coax-based 802.3/Ethernet. The 10BASE-T hub-based star architecture removes this obstacle.

Due to the widespread adoption of hub-based topologies that support Ethernet, Token Ring, and FDDI (Fiber Distributed Data Interface), the standardization of network management information and its exchange is also underway. Currently, Simple Network Management Protocol (SNMP) is the *defacto* standard employed in Ethernet networks for the management of infrastructure components, i.e., bridges and routers. The Institute of Electrical and Electronics Engineers is working to create a Layer Management for Repeater Devices standard (frequently referred to as the Hub or Repeater Management Draft[4]). This standard will specify the manageable attributes of a repeater. This draft has been adopted and is being reviewed simultaneously by the Internet Engineering Task Force to speed publication of the specification within the Internet community. This will ensure that migration to the hub-based topology will be rapid and widespread.

INTEGRATION AT THE NODE

Silicon vendors began the integration process at the transceiver. The huge installed base of AUI-compatible machines made silicon integration the natural choice, and silicon allowed the technology to be proven in a small device. The AMD Am79C98 TPEX™ (Twisted-Pair Ethernet Transceiver) is a fully compliant AUI-based 10BASE-T transceiver. The device incorporates 10BASE-T drivers and receivers. It also provides for status LEDs, as well as automatic detection and correction of the receive signal polarity. The device was jointly developed by AMD and SynOptics Communications Inc., an acknowledged leader in the 10BASE-T marketplace.

Clearly, volume consumption of 10BASE-T node connections will be in the desktop computer market, including IBM™ and Macintosh™ personnal computers, engineering workstations, X-Windows™ terminals, and similar desktop systems. In addition, significant opportunity exists for network growth in the mobile computer (laptops and palmtops) arena.

Personal Computers

In the PC arena, LAN connectivity is almost entirely serviced by the add-in card market. This is by far the largest market segment for Ethernet chipsets. Typically, the card

---

[4] Draft Supplement to ANSI/IEEE Std 802.3 -1990 Edition, Layer Management for Repeater Devices, Section 19.

consists of LAN components (supporting thick, thin, or twisted-pair Ethernet) and "glue" logic that interface to the appropriate bus, of which the PC/AT™, EISA, MCA™ and various Macintosh buses are the most prevalent.

Two distinct add-in card architectures are present, the bus master and the bus slave. Both are generally non-intelligent. Popular when bus and processor speeds were lower, the use of node processors on the card is now rare in Ethernet applications except in specialized areas that require some form of protocol processing, e.g., bridges and routers.

Bus mastership allows direct memory access (DMA) directly between the LAN controller and system memory, so no on-board shared memory is required. A good example of a bus master design is the Am2110 PC/AT card. The card is software-compatible with Novell NE2100 or EXOS™ 105 products (see Figure 9). The board is ideally suited for client applications because of its high performance and low cost (no on-board memory).

Numerous papers have been written that compare these two competing approaches, and enumerable tests have attempted to prove that one approach is universally better than the other. Basically, there is no clear winner for all applications.

Bus mastership also allows DMA directly into the system memory, so once again no on-board shared memory is required. However, if the Network Operating System (NOS) cannot react fast enough to a receive message request to allocate a buffer for the receive frame, then the receive packet must be placed in an intermediate memory area. It is then copied to the final location once the NOS provides the address. This duplicated copying effort wastes time.

A shared-memory approach allows additional latency for the NOS to prepare a receive packet location, but this approach requires an on-board memory plus logic to arbitrate access to the memory between the LAN controller and the host.

In a client-server architecture, the receipt of client information (packet data) is governed by the server's response speed. Because the server receives requests for data from all clients, there is minimal client-to-client traffic, so the server requires fast disk and LAN subsystems.

16850A-009A
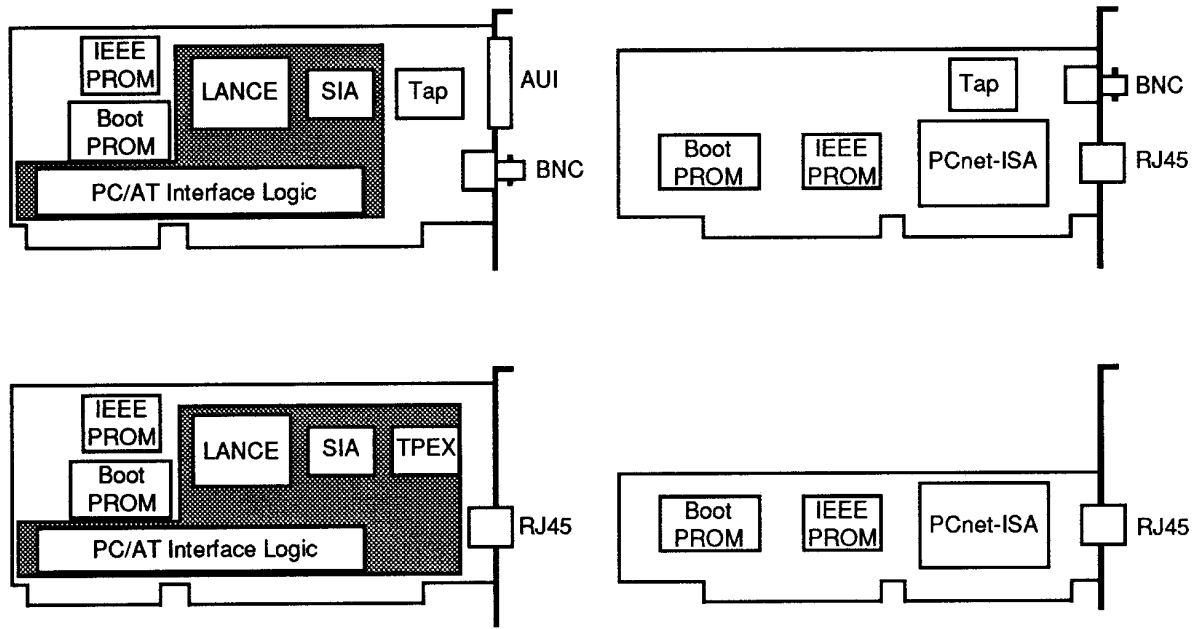
Figure 9. NE2100/NE1500 Equivalent Architectures

Introduction

With the submicron-level geometries available to mainstream silicon vendors and the 10BASE-T transceiver's effectiveness when implemented in this technology, it is now possible to integrate virtually the entire add-in card onto a single chip. Only those components that require custom programming (i.e., remote boot PROM and IEEE address PROM) need remain external to the integrated Ethernet node silicon. Note that it is already possible to integrate the MAC, PLS[5], and 10BASE-T transceiver functions. However, this still does not provide the most cost-effective system solution. The incorporation of the bus interface logic to provide a single-chip, bus-specific LAN controller offers the optimal level of integration.

This approach clearly reduces the component count, complexity, and form factor for the Ethernet add-in card market. In addition, the RJ45 phone jack connector used for 10BASE-T incurs minimun board-area and computer backpanel space. The increased integration, reduced component cost, and reduction in form factor that 10BASE-T brings are the attributes that will allow Ethernet to become a standard feature on the PC motherboard.

Currently, laptop and palmtop PCs are most frequently networked using an external adaptor that provides Ethernet connection through the computer's parallel printer port. Battery life is the important factor in the mobile computer environment. To address this issue, AMD Ethernet node chips will incorporate sleep capabilities for low-power applications, thus allowing 10BASE-T to be incorporated onto the mobile computer motherboard.

Non-PC-Based Applications

Several significant market segments outside the PC area require networking as a standard or optional feature. These include applications such as the engineering workstation (EWS), X-Windows terminal, and laser beam printer (LBP).

Virtually all EWS systems are shipped with Ethernet on the motherboard, although the AUI is usually the network connector of choice, requiring an external transceiver. The Am7990 LANCE™ (Local Area Network Controller for Ethernet), a first generation Ethernet controller, is widely used with engineering workstations.

Most medium-to-high performance LBPs offer networking as an optional feature. Some incorporate the entire print server hardware and software capabilities. In almost all cases, the LAN connection is offered as an add-in card because it is impossible to predict the end user's preferred LAN technology. However, the prevalence of 10BASE-T will cause some vendors to integrate the chipset onto the LBP motherboard as a standard feature. Thus, 10BASE-T will become the cheaper and preferred technology, with other LAN options offered as retrofit.

AMD's Am79C900 ILACC™ (Integrated Local Area Communication Controller) has already taken the first step in the integration process by incorporating the controller (DMA controller and MAC function) and the Manchester encoder/decoder (the PLS function of the Serial Interface Adaptor) in a single CMOS device (see Figure 5). The device, a 32-bit version of the popular LANCE, offers software compatibility.

Both the LANCE and ILACC are bus master devices that are suitable in many systems. However, in some applications a slave-based Ethernet controller is preferred because of the synergy with other slave peripherals in the system.

---

[5] ISO/IEC 8802-3 :1990 (E); ANSI/IEEE Std 802.3-1990 Edition, Section 7.
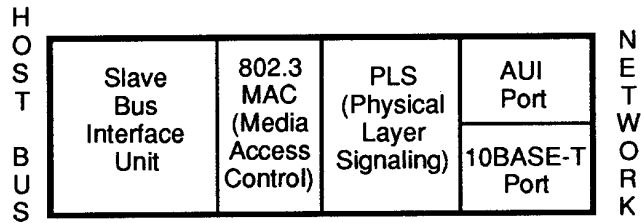
16850A-010A

Figure 10. MACE™ Block Diagram

The Am79C940 MACE (Media Access Controller for Ethernet) provides a simple slave interface and a high level of integration. The device integrates the MAC and PLS functions, providing an AUI port and 10BASE-T transceiver (Figure 10). There are pins for the 10BASE-T driver/receiver, status LEDs, and power supply functions.

INTEGRATION AT THE REPEATER

Several key influences are at work in the evolution of silicon for the repeater market.

Market Demand

Prior to the development of 10BASE-T, repeaters were low-volume devices. Semiconductor suppliers attempted to "siliconize" the repeater function (although several network system vendors produced implementations using gate arrays or similar designs). In the 10BASE-T star topology, the repeater is the network. Coupled with market demand, this reliance on silicon has encouraged semiconductor vendor participation. The 10BASE-T market will immediately benefit from lower costs. Longer-term benefits will be realized as silicon vendors amortize the 10BASE-T development effort and apply the repeater technology to coaxial and fiber media. This will result in an overall lowering of repeater technology cost, an increase in offerings, and a more stable base line of standard features. These factors in turn will increase user acceptance and broaden demand.

Hub Architecture

Manageability of the network has become a key issue. The star architecture preferred in most office environments for cabling and management fits perfectly with 10BASE-T. Nodes can be selectively enabled or disabled, or the link status can be examined. The network traffic on any port can be monitored to generate statistics on network activity or collision frequency for any specific time period.

Standardization

The 10BASE-T standard enforces a mandatory feature set on repeaters to ensure interoperability and minimize network disruption. The IEEE Repeater Standard[6] allows the "partitioning" algorithm to be optionally implemented. In 10BASE-T the partitioning algorithm must be present on all repeater ports with a 10BASE-T MAU. The purpose of partitioning is to monitor the port for either an excessive number of collisions or collisions with an excessive duration. In either occurrence the repeater port is partitioned so that its transmit function is disabled, although the receive function remains active to permit automatic healing once the fault condition is relieved.

A simple wiring error or physical short circuit that directly connects the transmit and receive circuits together is one example in which partitioning protects network integrity. Each time the repeater receives data on a correctly configured port, it repeats the data to all other ports. The MAU with transmit and receive shorted together always generates a

---

[6] ISO/IEC 8802-3 :1990 (E); ANSI/IEEE Std 802.3-1990 Edition, Section 9.

collision indication because both transmit and receive functions become active simultaneously. Hence, a collision is generated for every packet. In this case, the partitioning state machine for the port isolates the transmit function of the miss-wired port after a predefined maximum number of consecutive collisions are detected (>30).

The need for multi-vendor interoperability is generating a great deal of activity in the standardization of management information and its transaction, as seen by the activities of the IEEE Hub Management Task Force. Once the Hub Management Draft becomes stable, silicon vendors will resume the integration process to allow full support of the IEEE-defined Management Information Base.

High Reliability

The star architecture has one distinct drawback in that the repeater is itself a potential single-point failure. Because the repeater is responsible for the regeneration of all network traffic, failure could isolate all attached nodes. Silicon integration of the repeater brings with it a reduction in the number of inter-chip connections, which is a major contributor to mechanically related failures. From the system perspective, the integration can and should allow for swappable modularity and redundancy at the medium level if required.

Silicon Availability

AMD's Am79C980 Integrated Multiport Repeater™ (IMR) integrates all functions of an 802.3-compliant repeater on a single chip, including the Manchester encoder/decoder, first in-first out (FIFO) buffers, repeater state machines, and eight complete 10BASE-T MAUs. In addition, the device supports a single AUI port (see Figure 11), a management port, and an expansion port. AMD worked jointly with Hewlett-Packard to implement the functionality of HP's original EtherTwist™ hub product. HP's network systems expertise and AMD's mixed analog/digital CMOS IC strength have produced a fully compliant and highly integrated product.
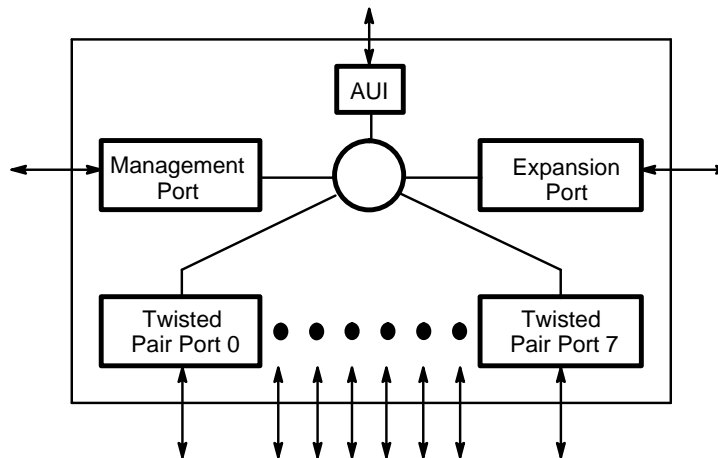
16850A-011A

Figure 11. IMR Block Diagram

The IMR reduces the cost and complexity of repeater technology, enhances reliability, and opens the market to a wide variety of repeater solutions. Because virtually all the expertise to perform the repeater function is contained in the silicon itself, the technology entry barrier has been removed.

Velcro Hubs

The IMR provides a completely scalable solution to the repeater market. For low-end systems, the IMR can be combined with a power supply, crystal, and electromagnetic interference/radio frequency interference (EMI/RFI) filter and transformer modules to effectively produce a fully operational 10BASE-T repeater with an AUI port that allows connection to an existing 10BASE2/5 coax backbone.

16850A-012A
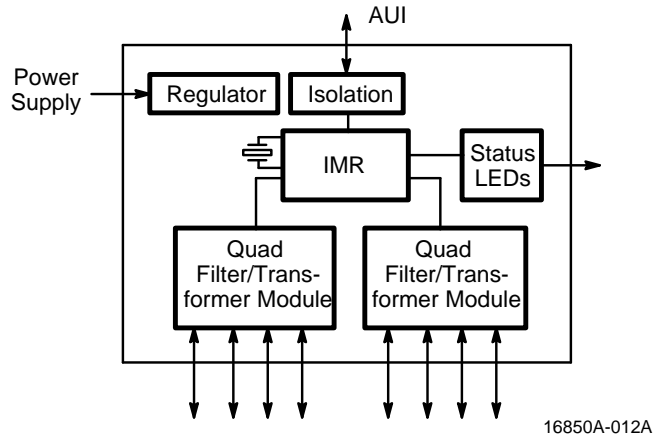
Figure 12. Simple Velcro Hub Example

ISA-HUB™

An excellent example of a low-cost managed repeater is the ISA-HUB concept. In a client-server environment, the file-server already acts as a hub for shared resources. By implementing a repeater in a PC/AT-compatible format, the power supply and case for the hub are provided by the server itself. In addition, the server has additional compute power available to perform hub management. The port count can be increased by cascading multiple IMRs together using the built in expansion port.

Fully Managed Hubs

For high-end applications requiring facilities such as modularity, fault-tolerance manage-ment to the Hub Management Draft, and SNMP agent capabilities, external logic and intelligence can be added to provide these features in a cost-effective manner. In this way, low-end repeaters can be produced at optimum system cost, while the cost per port will increase accordingly as sophisticated management features are added.

SUMMARY

Simpler configuration and reconfiguration capabilities combined with network manage-ment capabilities mean that a 10BASE-T network is far more able to respond to person-nel and desktop computer movement than a coax-based Ethernet.

The cost to implement 10BASE-T Ethernet will be driven down to the point that it will become a standard feature in many desktop computer applications, including laser printers, PCs, and X-Windows terminals.

The scalability and low entry cost of 10BASE-T will ensure its success. 10BASE-T will have a major impact in the networked office, from the small user group requiring simple connectivity through remote offices connected by dial-up lines to interconnection with existing Ethernet LANs in large-scale corporate installations.

Current generation silicon is already fostering price reductions in 10BASE-T MAUs, add-in cards, and repeaters. Future integration of the 10BASE-T silicon is assured as the Hub Management Standard solidifies and security and fault tolerance features are incorporated.

CHAPTER 1
System Considerations

The PCnet-ISA is designed to interface seamlessly to the PC/AT expansion bus, known formally as the Industry Standard Architecture (ISA) bus. In bus master mode, it is possible to connect the ISA bus pins of the PCnet-ISA chip directly to the edge connector pads on an adapter card without using external buffers, drivers, or decoders. All system bus buffering and address decoding can be done inside the PCnet-ISA. Shared-memory mode requires some external hardware, but it offers additional architectural flexibility.

1.1 Bus Master Mode

When configured in bus master mode, the PCnet-ISA allows the user to design a high-performance Ethernet system with minimal cost and parts count. The PCnet-ISA is also compatible with the Novell NE2100 and NE1500T Ethernet adapters, which are based upon the Am7990 LANCE (Local Area Network Controller for Ethernet) chip.

The basic operation of the PCnet-ISA in bus master mode consists of input/output slave cycles for control and status, and bus master DMA (direct memory access) operations (16-bit) for the movement of transmit and receive data. These DMA operations take place over the ISA bus; therefore, the host system must support bus mastering. During the PCnet-ISA bus mastering periods, the host CPU is inactive.

1.1.1 Hardware Considerations

When in bus master mode, the PCnet-ISA requires a 16-bit host platform such as the PC/AT. The PCnet-ISA will not function as a bus master in an 8-bit environment. Although the PCnet-ISA supports only 8-bit PROM accesses and can support 8-bit input/output slave cycles, the PCnet-ISA assumes that all bus master memory cycles are 16-bit; there is no support in the chip to support 8-bit DMA cycles.

From the host system software point of view, the PCnet-ISA-based adapter represents two ISA bus software resources, namely I/O read and write space and memory read-only space. The I/O space consists of internal PCnet-ISA registers and an optional IEEE Ethernet address PROM. The read-only memory space refers to an optional external boot PROM. While both resources can be accessed as 16-bit software entities, the internal PCnet-ISA registers must be coded as 16-bit word locations.

Connection-wise, the PCnet-ISA ISA bus pins should be connected directly to the ISA bus. External PROM devices share only the ISA bus system address lines with the PCnet-ISA, but all other PROM connections (data and output enable) are controlled by dedicated pins on the PCnet-ISA.

1.1.2 Register Addressing

The PCnet-ISA has internal user control and status registers (CSRs) that facilitate configuration and status transfers between the PCnet-ISA and the host system. Because there are too many CSRs for individual I/O addresses, an indexed addressing scheme is used to access each individual CSR. Specifically, the PCnet-ISA has a single register address port (RAP) into which the CSR number is written. Once the register address port contains the desired CSR number, the CSR data are read from the register data port (RDP). Sequential accesses to the same CSR do not require new writes to the register address port.

A second set of internal registers, the ISA bus configuration registers (ISACSRs), is accessed in the same way as the CSRs, except that the data are read not from the RDP, but from the ISA bus data port (IDP).

All I/O accesses to the IEEE address PROM locations are individually addressed as standard I/O locations.

For the 8-bit I/O accesses, the PCnet-ISA utilizes the system address lines (SA0-19), the system data lines (SD0-7), I/O Read ($\overline{IOR}$), I/O Write ($\overline{IOW}$), and address enable (AEN); the PCnet-ISA returns I/O channel ready (IOCHRDY). If the PCnet-ISA is configured to support 16-bit I/O bus cycles, the system data (SD8-15) lines and system byte hgh enable ($\overline{SBHE}$) are also used. In this configuration I/O chip select 16 ($\overline{IOCS16}$) is returned active only if the internal PCnet-ISA registers are accessed; accesses to the IEEE PROM will be 8-bit I/O bus cycles only.

The PCnet-ISA can be configured to perform only 8-bit I/O bus cycles, even for internal registers. This is accomplished by leaving the $\overline{IOCS16}$ pin disconnected from the ISA bus, and instead shorted to ground.

The following table shows the I/O address space supported by the PCnet-ISA.

Offset	# of Bytes	Register
0h	16	Address PROM
10h	2	RDP
12h	2	RAP (shared by RDP and IDP)
14h	2	Reset
16h	2	IDP
18h	2	Vendor-Specific Word

Vendor-specific word is not internally supported by the PCnet-ISA. It is shown here as an externally decoded address space that is available for use by the system designer; it will not be used by the PCnet-ISA or any future AMD Ethernet controller.

The offsets listed are referenced to the base I/O address of the PCnet-ISA. This base address is defined by the current state of the I/O address map (IOAM) pins. The IOAM pins also define the boot PROM memory address space to which the internal address decoder will respond. The IOAM pins are not sampled but simply inputs to the internal address decoder.

IOAM1-0	I/O Base	Boot PROM Base
0 0	300 h	C8000 h
0 1	320 h	CC000 h
1 0	340 h	D0000 h
1 1	360 h	D4000 h

1.1.3 Boot PROM
The PCnet-ISA performs the boot PROM address decoding and ISA bus buffering internally. The address decoder is defined by the state of the IOAM pins. For the 8-bit memory bus cycles, the PCnet-ISA utilizes the SA0-19 and unlatched address lines (LA17-23) address lines, the SD0-7 data lines, system memory read select ($\overline{SMEMR}$), memory refresh active ($\overline{REF}$) and AEN; the PCnet-ISA returns only IOCHRDY. The PCnet-ISA supports 8-bit slave memory bus cycles only; the PCnet-ISA does not utilize $\overline{SBHE}$ for slave memory cycles.

1.1.4 Hardware Compatibility Issues

As stated before, the PCnet-ISA in bus master mode must be installed in a host system that supports bus mastering. The PC/XT™ and some PC/AT compatibles do not support bus mastering and therefore will not function as PCnet-ISA (bus master mode) hosts. In addition, the PCnet-ISA in bus master mode requires 16-bit memory for DMA transfers. This precludes the use of the PC/XT and the 8-bit slots in PC/AT clones.

Another possible compatibility issue involves the $\overline{IOCS16}$ signal. The IEEE P996 ISA bus standard recommends that the $\overline{IOCS16}$ signal be generated as a decode of the SA address lines and AEN only, but some systems (i.e. motherboard chipsets) are not compatible with this timing. Attempts at universal compatibility have been implemented, most of them involving delaying the $\overline{IOCS16}$ signal until $\overline{IOR}$ or $\overline{IOW}$ become active. This scheme helps in some cases, but it causes new incompatibilities elsewhere. The PCnet-ISA follows the IEEE P996 recommendation for the generation of $\overline{IOCS16}$.

The 16-bit I/O bus cycle compatibility question can be eliminated simply by configuring the PCnet-ISA to support 8-bit I/O bus cycles only. This is the recommended configuration. Because the vast majority of PCnet-ISA operations are as bus master direct memory accesses, limiting the PCnet-ISA I/O bus cycles to 8-bit only will have virtually no effect on system throughput and performance. This configuration is accomplished by disconnecting the PCnet-ISA $\overline{IOCS16}$ pin from the ISA bus and tying the pin to ground. All I/O bus cycles (internal PCnet-ISA registers and IEEE PROM) as well as boot PROM memory bus cycles, will be 8-bit only. The PCnet-ISA always performs its bus master DMAs as 16-bit transfers.

Some PC/AT-compatibles have been known to spuriously assert DMA acknowledge (DACKx) signals (other than the refresh controller $\overline{DACK}$) during memory refresh cycles. If a bus master device such as the PCnet-ISA requests the bus (by asserting DMA request (DRQx)), it might interpret the spurious $\overline{DACKx}$ assertion as permission to assume bus mastership. For this reason, the $\overline{DACK}$ signal is ignored by the PCnet-ISA when the $\overline{REF}$ pin is asserted. The $\overline{SMEMR}$ signal is also ignored by the PCnet-ISA while $\overline{REF}$ is asserted.

1.1.5 Bus Bandwidth and Latency Requirements

The PCnet-ISA performs the vast majority of its tasks as bus master DMA transfers. These include initialization, ring polling, descriptor reads and writes, and frame data (buffer) transfers. Because the Ethernet interface operates at a set rate, the PCnet-ISA must ensure that sufficient data are available to transmit or that space is available to store the received data. This scenario requires some definitions.

Bus master - Any device that currently controls or owns the ISA bus, i.e., drives the address lines or the command lines (e.g., $\overline{IOR}$, $\overline{IOW}$, memory read select (MEMR), memory write select (MEMW)) and directs the flow of data over the ISA bus data lines. In an ISA environment, the CPU is the default bus master. Any non-CPU device may obtain bus master status by arbitrating for ISA bus ownership. This arbitration process is a request-acknowledge handshake protocol.

Bus bandwidth - The percentage of time that any device (bus master) controls the ISA bus. This percentage changes depending upon the current task being performed by the bus master. The PCnet-ISA uses the most bandwidth during frame transmission and reception; it uses the least bus bandwidth when the PCnet-ISA is simply polling the descriptor rings. Conversely, the amount of available bus bandwidth changes depending upon the tasks that the other bus masters are currently performing.

Bus latency - The time beginning when a device (bus master) requests ownership of the ISA bus until the device actually takes ownership of the bus. This time changes depending upon the available bandwidth of the ISA bus.

The critical issue here is bus latency. A PCnet-ISA that is receiving a frame must have access to the ISA bus so that it can transfer the frame to memory; making the PCnet-ISA wait too long for bus mastership results in a missed packet due to first in-first out (FIFO)

buffer overflow, error, or both. A transmitting PCnet-ISA must have system memory data continuously supplied to the FIFO to maintain the 10 Mbps Ethernet rate; late ownership of the ISA bus leaves the PCnet-ISA in the middle of a frame transmission with no data to finish transmission. It is, therefore, critical that enough bus bandwidth be available to ensure minimal bus latency. Because each bus master (CPU, refresh controller, DMA controller, PCnet-ISA) in an ISA environment uses a finite amount of bus bandwidth, it is important that the ISA bus is not overtaxed by any one bus master.

Under optimal conditions, bus bandwidth utilization by the PCnet-ISA can be approximated by the ratio of the Ethernet data rate to the ISA bus DMA data rate. The maximum data rate is based upon the bus master cycle time of the PCnet-ISA. The default value is 350 ns; this can be tuned for faster or slower operation by reprogramming the ISA configuration register. In addition, maximum performance is achieved when buffers are not chained. Chaining occurs when the frame data cannot fit into a single buffer. Assuming default bus timing and no buffer chaining for the PCnet-ISA, this maximum data rate is as follows:

One DMA bus cycle per 350 ns X 16 bits per DMA bus cycle = 45.7 Mbps

This calculation does not take into account descriptor accesses; it does give the PCnet-ISA a minimum bus bandwidth usage of approximately 22%. This number will degrade if the buffers per frame ratio increases.

The PCnet-ISA is most sensitive to bus latency when large Ethernet frames are involved and small data buffers are chained together. This is due to the PCnet-ISA having to do descriptor look-ahead operations for each chained buffer, thus using bus bandwidth without transferring frame data. Assuming 32-byte chained buffers, the calculations for each buffer are provided below.

32 bytes transferred to or from FIFO over the ISA bus = 16 bus cycles X 350 ns per cycle = 5.6 μs

(This requires one bus arbitration as the PCnet-ISA does 16 word memory cycles per arbitration.)

Descriptor look-ahead operation = 3 bus cycles X 350 ns per cycle = 1.1 μs

(This requires one bus arbitration as the PCnet-ISA does all descriptor reads in one arbitration.)

Descriptor update operation = 2 bus cycles (max) X 350 ns per cycle = 700 ns

(This requires one bus arbitration as the PCnet-ISA does all descriptor writes in one arbitration.)

Microcode travel time (internal PCnet-ISA microcode decision delay) ≈ 1 μs

These operations transfer 32 bytes, take a total of 8.4 μs (not including bus latency), and require a total of three bus arbitrations. The network takes 32 bytes X 8 bits per byte X 100 ns per bit = 25.6 μs to transmit 32 bytes of frame data; the PCnet-ISA must complete the above bus arbitrations and transfer operations at least once every 25.6 μs in order to keep up with the network. This leaves

25.6 μs total time - 8.4 μs operations time = 17.2 μs

of total allowable latency for three bus arbitrations, or 5.7 μs average bus latency tolerance. The bus bandwidth usage is 7.4 μs / 25.6 μs = 29%. The latency tolerance improves to 12.7 μs and the bus bandwidth improves to 24% when using 128-byte chained buffers.

1.2 Shared-Memory Mode

When configured in shared-memory mode, the PCnet-ISA allows the user to design a high-performance Ethernet system for environments that do not support 16-bit memory cycles or in which bus mastering is either not supported or not desired.

The basic operation of the PCnet-ISA in shared-memory mode consists of I/O slave cycles for control and status, and shared memory operations for the movement of transmit and receive data. These shared-memory operations take place over the PCnet-ISA private data bus (PRDB) and private address bus (PRAB) and are invisible to the host system. Therefore, during the PCnet-ISA PRDB/PRAB accesses, the host CPU is not interrupted or preempted. This aspect allows operation in host systems that either do not support bus mastering or cannot afford CPU preemption. In shared-memory mode, no PCnet-ISA DMA operations take place over the ISA bus.

1.2.1 Hardware Considerations

When in shared-memory mode, the PCnet-ISA does not require a 16-bit platform. If the host system is 8-bit only (such as the PC/XT), all bus cycles will obviously be 8-bit. However, if the platform is 16-bit, the PCnet-ISA can support 16-bit bus cycles for all resources except for the IEEE address PROM, which supports 8-bit I/O cycles only.

From the host system software point of view, the PCnet-ISA-based adapter represents three ISA bus software resources, namely I/O read and write space, memory read and write space, and memory read-only space. The I/O space consists of internal PCnet-ISA registers and an optional IEEE Ethernet address PROM. The read-only memory space refers to an optional external boot PROM. While all resources can be accessed as 16-bit software entities, internal PCnet-ISA registers must be coded as 16-bit word locations.

Connection-wise, the PCnet-ISA ISA bus pins should be connected directly to the ISA bus. The external PROM devices and SRAM devices are connected directly to the PCnet-ISA private address bus and private data bus, and all other PROM and SRAM connections (PROM output enable (OEs); SRAM $\overline{OE}$ and write enable (WE)) are controlled by dedicated pins on the PCnet-ISA. An external 6-bit address buffer is required, as are external address comparators for the boot PROM and SRAM.

1.2.2 Register Addressing

As in the bus master mode, the PCnet-ISA has internal user control and status registers in the shared-memory mode that facilitate configuration and status transfers between the PCnet-ISA and the host system. Because there are too many CSRs for individual I/O addresses, an indexed addressing scheme is used to access each individual CSR. Namely, the PCnet-ISA has a single register address port (RAP) into which the CSR number is written. Once the register address port contains the desired CSR number, the CSR data are read from the register data port. Sequential accesses to the same CSR do not require new writes to the register address port.

A second set of internal registers, the ISA bus configuration registers (ISACSRs) is accessed in the same way as the CSRs, except that the data are read not from the register data port, but from the ISA bus data port (IDP).

All I/O accesses to the IEEE address PROM locations are individually addressed as standard input/output locations.

For 8-bit I/O accesses, the PCnet-ISA uses the SA0-19 address lines, the SD0-7 data lines, $\overline{IOR}$, $\overline{IOW}$, and AEN; the PCnet-ISA returns IOCHRDY. If the PCnet-ISA is configured to support 16-bit I/O bus cycles, the SD8-15 lines and $\overline{SBHE}$ are also used. In this configuration $\overline{IOCS16}$ is returned active only if the internal PCnet-ISA registers are accessed; accesses to the IEEE PROM will be 8-bit I/O bus cycles only.

The PCnet-ISA can be configured to perform 8-bit I/O bus cycles only, even for internal registers. This is accomplished by leaving the $\overline{IOCS16}$ pin disconnected from the ISA bus and instead shorting it to ground.

The PCnet-ISA will be forced into 8-bit bus cycle mode for all accesses (registers, PROMs, and SRAM) if the $\overline{SBHE}$ pin is left unconnected. This situation would arise in a PC/XT or an 8-bit slot of a PC/AT.

The table below shows the I/O address space supported by the PCnet-ISA.

Offset	# of Bytes	Register
0h	16	Address PROM
10h	2	RDP
12h	2	RAP (shared by RDP and IDP)
14h	2	Reset
16h	2	IDP
18h	2	Vendor-Specific Word

Vendor specific word is not internally supported by the PCnet-ISA. It is shown here as an externally decoded address space that is available for use by the system designer; it will not be used by the PCnet-ISA or any future AMD Ethernet controller.

The offsets listed are referenced to the base I/O address of the PCnet-ISA. This base address is defined by the current state of the I/O address map (IOAM) pins. The IOAM pins are not sampled but simply inputs to the internal address decoder.

IOAM1-0	I/O Base
0 0	300 h
0 1	320 h
1 0	340 h
1 1	360 h

1.2.3 Boot PROM

The PCnet-ISA performs the boot PROM ISA bus buffering internally, but the address decoder function must be done externally. This external decoder defines the address space of the boot PROM and must generate the boot PROM address match (BPAM) signal to the PCnet-ISA; the $\overline{BPAM}$ signal is a unique address decode for the boot PROM address space. The PCnet-ISA assumes that boot PROM accesses are 16-bit ISA bus cycles. For these 16-bit memory bus cycles, the external address decoder should use the LA17-23 address lines and the appropriate SA address lines needed to generate a unique $\overline{BPAM}$ for the boot PROM address space. The external address decoder must also return $\overline{MEMCS16}$ to the ISA bus. The PCnet-ISA uses the SD0-15 data lines, $\overline{MEMR}$, $\overline{REF}$, $\overline{SBHE}$, and AEN; the PCnet-ISA returns only IOCHRDY.

The PCnet-ISA can be made to support 8-bit only boot PROM bus cycles while leaving the shared-memory SRAM supporting 8- and 16-bit bus cycles. This is accomplished through trickery with the $\overline{BPAM}$ and $\overline{SMAM}$ inputs. While the external address comparator would assert only $\overline{BPAM}$ for 16-bit boot PROM accesses, the decoder must assert $\overline{BPAM}$ and $\overline{SMAM}$ simultaneously for an 8-bit boot PROM bus cycle. While this is a logically illegal state, it will force the PCnet-ISA into an 8-bit boot PROM access. Asserting $\overline{BPAM}$ alone will cause the PCnet-ISA to execute a default 16-bit boot PROM bus cycle.

The PCnet-ISA will be forced into 8-bit bus cycle mode for all accesses (registers, PROMs, and SRAM) if the $\overline{SBHE}$ pin is left unconnected. This situation would arise in a PC/XT or an 8-bit slot of a PC/AT.

1.2.4 Shared Memory SRAM

The PCnet-ISA performs the shared-memory SRAM ISA bus buffering internally, but the address decoder function must be done externally. This external decoder defines the address space of the shared-memory SRAM and must generate the $\overline{\text{SMAM}}$ signal to the PCnet-ISA; the $\overline{\text{SMAM}}$ signal is a unique address decode for the SRAM address space. The PCnet-ISA assumes that SRAM accesses are 16-bit ISA bus cycles. For these 16-bit memory bus cycles, the external address decoder should use the LA17-23 address lines and the appropriate SA address lines needed to generate a unique $\overline{\text{SMAM}}$ for the SRAM address space. The external address decoder must also return $\overline{\text{MEMCS16}}$ to the ISA bus. The PCnet-ISA uses the SD0-15 data lines, $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{REF}}$, $\overline{\text{SBHE}}$, and AEN; the PCnet-ISA returns only IOCHRDY.

The PCnet-ISA can be made to support 8-bit only shared-memory SRAM bus cycles, but this forces the PCnet-ISA to support only 8-bit bus cycles for all resources (registers, PROMs, and SRAM). This is accomplished by leaving the $\overline{\text{SBHE}}$ pin unconnected. This situation would arise in a PC/XT or an 8-bit slot of a PC/AT. In this case the external address decoder for the SRAM should not return $\overline{\text{MEMCS16}}$ to the host system. Because the $\overline{\text{MEMCS16}}$ signal does not exist in the PC/XT environment, it is possible to design a single Ethernet adapter board that is compatible with both 8-bit and 16-bit environments but still takes full advantage of either system's resources.

1.2.5 Hardware Compatibility Issues

A possible PCnet-ISA and host system compatibility issue involves the $\overline{\text{IOCS16}}$ signal. The IEEE P996 ISA bus standard recommends that the $\overline{\text{IOCS16}}$ signal be generated as a decode of the SA address lines and AEN only, but some systems (e.g., motherboard chipsets) are not compatible with this timing. Attempts at universal compatibility have been implemented, most of them involving delay of the $\overline{\text{IOCS16}}$ signal until $\overline{\text{IOR}}$ or $\overline{\text{IOW}}$ become active. This scheme helps in some cases, but it causes new incompatibilities elsewhere. The PCnet-ISA follows the IEEE P996 recommendation for the generation of $\overline{\text{IOCS16}}$.

The 16-bit I/O bus cycle compatibility question can be eliminated simply by configuring the PCnet-ISA to support 8-bit I/O bus cycles only. This is the recommended configuration. Because the vast majority of PCnet-ISA operations are private address or data bus memory accesses and ISA bus memory cycles, limiting the PCnet-ISA I/O bus cycles to 8 bits only will have virtually no effect on system throughput and performance. This configuration is accomplished by disconnecting the PCnet-ISA $\overline{\text{IOCS16}}$ pin from the ISA bus, and tying the pin to the ground. All I/O bus cycles (internal PCnet-ISA registers, IEEE PROM) will then be 8 bits only. Leaving the $\overline{\text{SBHE}}$ pin disconnected will also force the PCnet-ISA to support only 8-bit I/O bus cycles, but it will also force all PCnet-ISA accesses (registers, PROMs, and SRAM) into 8-bit only bus cycle mode. This situation would arise in a PC/XT or an 8-bit slot of a PC/AT.

CHAPTER 2
Network Characteristics

2.1 Network Operation

As outlined in the Chapter 1, the PCnet-ISA acts as a bus mastering device when transferring packet data to or from host memory. In addition to bus mastership, additional features are built into the PCnet-ISA that increase its overall efficiency over other types of LAN controllers.

The primary objective of most of these features is to minimize bus occupancy that the PCnet-ISA requires for a given network throughput. These features will be discussed in terms of their purpose and their overall benefits. For a detailed description of the operation of each feature, consult the PCnet-ISA data sheet.[7] For additional detail on how various features can be accessed or programmed, consult Chapter 5 of this manual.

As a starting point, some definitions of the 802.3 and Ethernet frame formats are included here. These definitions will be the basis for understanding the network performance of the PCnet-ISA and its relative effect on the Industry Standard Architecture (ISA) bus. For additional details, consult the 802.3[8] and Ethernet[9] specifications.

Preamble 1010....1010	SFD 10101011	Dest Addr	Srce Addr	Length	LLC Data	Pad	FCS
56 Bits	8 Bits	6 Bytes	6 Bytes	2 Bytes	46—1500 Bytes		4 Bytes

16850A-013A
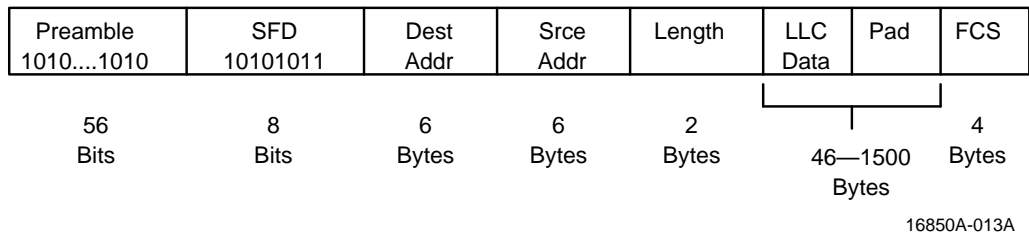
Figure 13. 802.3 Frame Organization

Preamble 1010....1010	SYNCH 11	Dest Addr	Srce Addr	Type	Data	FCS
62 Bits	2 Bits	6 Bytes	6 Bytes	2 Bytes	46—1500 Bytes	4 Bytes

16850A-014A
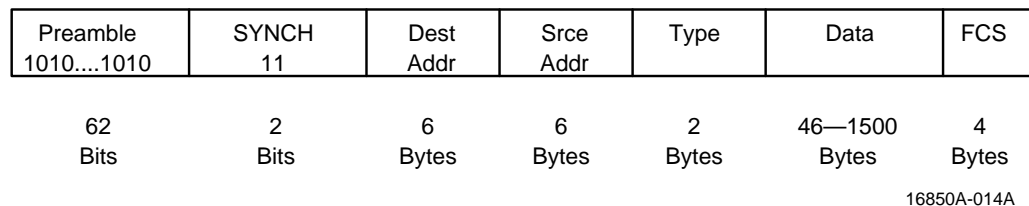
Figure 14. Ethernet Frame Organization

---

[7] PCnet-ISA Preliminary Information Data Sheet, Publication #16907A.

[8] The Ethernet: A Local Area Network Data Link Layer and Physical Layer Specification, Version 2.0, Nov 1982, DEC, Intel, Xerox.

[9] ISO/IEC 8802-3 :1990 (E); ANSI/IEEE Std 802.3-1990 Edition, Section 8.

The primary difference between the 802.3 and Ethernet frames are described below.

- The start frame delimiter of 802.3 is defined as a byte containing the 10101011 pattern, whereas Ethernet's synch bits have a 11 sequence. However, in both cases the preamble plus the start of frame indication is 64 bits long.

- Ethernet and 802.3 both specify that a packet must range from 64 to 1518 bytes. However, the actual data field in 802.3 may be smaller than the 46-byte value needed to ensure this minimum size. As a result, 802.3 requires the Logical Link Control (LLC) layer to append pad characters to the data field before passing the data to the Media Access Control (MAC) layer. Ethernet assumes that the upper layers ensure that the minimum data field is 46 bytes before passing the data to the MAC, and the concept of pad characters (although they may have effectively been inserted by the upper-layer software) does not exist.

- IEEE 802.3 uses a length field to indicate the number of data bytes (excluding pad characters) that are in the data field only (excluding the frame check sequence (FCS) field). Ethernet, on the other hand, uses a type field in the same 2 bytes to identify the message protocol. Because valid Ethernet type fields are always assigned above the maximum 802.3 packet size, Ethernet and 802.3 packets can coexist on the same network.

Note that the packet size of 64 to 1518 bytes refers to the number of bytes after the start of frame, up to and including the FCS. The preamble is used only by the Manchester encoder/decoder to lock-on to the incoming receive bit stream. The preamble is not passed through the MAC to the host system.

2.1.1 Collision Fragment and Runt Packet Rejection

The receive FIFO in the PCnet-ISA is 128 bytes deep. After power-up and initialization, the PCnet-ISA always waits for the receive FIFO to fill to a minimum of 64 bytes, before it requests the ISA bus by asserting the DMA request (DRQ) pin. In a correctly configured 802.3/Ethernet network, all packets contain a minimum of 64 bytes of data, and all collisions are guaranteed to occur within 512-bit times (64 bytes) from the start of the packet. This means that a receive message that terminates before 64 bytes (due to either a collision or the packet completing) is flushed from the receive FIFO with no host interaction.

This flushing has particular benefits in heavily loaded networks, where collisions can occur frequently. Because the packet is flushed from the FIFO prior to the assertion of DRQ, there is no impact on ISA bus bandwidth utilization.

Passing of runt packets (less than 64 bytes) can optionally be permitted by the PCnet-ISA. This feature is enabled by setting the runt packet accept bit (RPA, CSR124). In normal operation, runt packets are rejected. Some non-compliant network protocols may send runt packets, however.

Note that the 512-bit time (64-byte time) period (the slot time) is the time during which a collision occurs on a normally configured network. The slot time is the worst-case end-to-end delay of the network, so that a transmitting node guarantees to see a collision within this period if another node simultaneously starts to transmit elsewhere. The slot time starts with the first bit of preamble, whereas the 64-byte minimum packet size refers to the the actual packet data excluding the preamble.

2.1.2 Auto Retransmit or Deletion on Collision

Automatic retry of collisions within the slot time. The PCnet-ISA ensures that collisions that occur within 512-bit times from the start of transmission (including the preamble) are automatically retried with no host intervention or system bus impact. The transmit FIFO guarantees that data contained within the FIFO are not overwritten until at least 64 bytes (512 bits) of data have been successfully transmitted onto the network. These criteria will be met, regardless of whether the transmit frame was the first (or only) frame in the transmit FIFO or ithe transmit frame was queued pending completion of the preceding frame.

Deletion of packets due to excessive transmission attempts. If 16 total attempts (initial attempt plus 15 retries) are made to transmit the frame, the PCnet-ISA abandons the transmit process for the particular frame, reports a retry error (RTRY) in the transmit message descriptor (TMD3), gives up ownership (sets the OWN bit to zero) of the transmit descriptor, and sets the transmit interrupt (TINT) bit (CSR0), activating the external interrupt request (IRQ) pin provided the interrupt is unmasked. If the disable retry (DRTY) bit in the mode register (CSR15) is set, RTRY will set after one failed transmission attempt. Internally, the PCnet-ISA updates the transmit FIFO read and write pointers. If no end-of-frame marker exists in the transmit FIFO, the entire FIFO is reset. If a whole frame does reside in the transmit FIFO, the read pointer is moved to the start of the next frame or free location in the FIFO, and the write pointer is unaffected.

Packets experiencing 16 unsuccessful attempts to transmit are not retried. Recovery from this condition must be performed by upper-layer software.

No minimum transmit buffer length. A particularly important aspect of this recovery mechanism is that the PCnet-ISA has no requirement for a minimum transmit buffer size. This is not the case in the original LANCE, which requires a minimum transmit buffer length to ensure that ownership of the transmit descriptor is maintained until after the slot time expires. This enables the LANCE to reload the transmit data into its FIFO for the retransmission attempt. Substantial ISA bus bandwidth savings can be observed in a PCnet-ISA implementation over a LANCE-based PC adaptor design (such as the Novell NE1500T or NE2100) in heavily loaded networks where significant collision activity may be present. The LANCE-based design must reload the transmit data into its FIFO when a collision is detected during transmission. The PCnet-ISA does not require this reload because the transmit FIFO data are protected.

Elimination of the minimum transmit buffer size restriction simplifies device driver writing, although existing LANCE drivers that obey the restriction function correctly with the PCnet-ISA.

Even though the PCnet-ISA requires no minimum transmit buffer size, using extremely small buffers increases the overall bus occupancy on the ISA bus because the descriptor fetch and look-ahead operations occur much more frequently (in relation to the number of data bytes transferred). In addition, if the ISA bus is heavily utilized by other devices, a transmit underflow condition could result if there is insufficient bus bandwidth to allow the the PCnet-ISA to provide sufficient data to the transmit FIFO to keep up with the network data rate.

2.1.3 Programmable Transmit Start Threshold
The transmit start point (XMTSP, CSR80) controls the point at which preamble transmission commences on the network (in relation to the number of bytes written into the transmit FIFO) for the current frame. When the entire frame is in the FIFO, transmission starts regardless of the value in XMTSP. In addition, transmission does not start until the inter-packet gap (IPG) timer expires or the random backoff interval is completed. The XMTSP is given a value of 10 (64 bytes) after hardware or software reset. Regardless of XMTSP, the FIFO will not internally over write its data until at least 64 bytes or the entire frame is transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be rewritten to the transmit FIFO and retries are handled autonomously by the MAC.

CSR80[11-10]	Bytes Written
00	4
01	16
10	64
11	112

The XMTSP default of 64 bytes is logically chosen based on the slot time (64 bytes) and the size of the transmit FIFO (128 bytes). If alternative programming of the XMTSP is used, consideration should be given to system performance aspects. If the PCnet-ISA is the only bus master device in the system, there is little effect other than the lower values (4 or 16 bytes) allowing the device to contend for the network at an earlier point. If the PCnet-ISA is in a multi-master environment, consideration should be given to the likelihood of transmit underflow conditions occurring if the transmit FIFO cannot be filled because of ISA-bus bandwidth restrictions. With the 4-byte threshold, the transmit FIFO must receive additional data before the preamble/start fram delimiter (64 bits) and first 4 bytes (32 bits) have been transmitted (within 9.6 µs). Because the PCnet-ISA normally reads data from the host memory in 32-byte blocks, the preamble will start as more data are written into the transmit FIFO during the rest of the burst. However, caution should be used when other options have been programmed to limit the time or number of transfers on the ISA bus (using the bus activity timer register (CSR82) or burst register (CSR80) or very small transmit buffers are being used.

In the default case, the PCnet-ISA requires two separate bursts of 32 bytes each to be executed before the XMTSP is reached. These yield a latency of 57.6 µs before the transmit FIFO needs additional data.

2.1.4 Programmable Burst Size
The PCnet-ISA moves transmit and receive data across the ISA bus in bursts under bus mastership control. In the default mode, these bursts are 16 read or write cycles corresponding to 32 bytes of data, assuming the data buffer is aligned on a word (16-bit) boundary.

The burst size can be modified using the DMA burst register (DMABR) field within the burst and FIFO threshold control register (CSR80). Any value from 1 to 255 transfers can be programmed. Regardless of the programmed burst size (default or user-programmed), the burst terminates once the programmed high watermark is reached on the transmit FIFO or the low watermark on the receive FIFO.

Programming larger values than the default allows the arbitration time for mastership of the ISA bus (which may be significant) to be amortized over a greater number of data transfer cycles, thereby increasing the overall efficiency of the bus. However, care should be taken to avoid creating a "bus hog" that may cause other devices to experience problems if they are prevented from accessing the ISA bus for significant periods of time. The LANCE uses an 8-cycle fixed burst size (16 bytes maximum), so the PCnet-ISA already makes better utilization of the bus for each arbitration cycle. The PCnet-ISA takes slightly longer on the bus to transfer 32 bytes (slightly over 5 µs) than the LANCE-based NE2100 and NE1500T cards to transfer 16 bytes.

The burst size should be constrained to allow normal memory refresh approximately every 15 µs.

2.1.5 DMAPLUS
As explained in the previous section, in the default operation mode DMA cycles for transmit and receive data movement occur in 16-cycle bursts or until the FIFO high or low watermark is reached. The DMAPLUS bit (CSR4) allows the PCnet-ISA to fill or empty the transmit or receive FIFO in one burst during a single arbitration cycle.

If DMAPLUS = 0, a maximum of 16 transfers is performed. This may be modified by writing to the burst register (CSR80). If DMAPLUS = 1, a burst will continue until the transmit FIFO is filled to its high threshold (32 bytes default) or the receive FIFO emptied to its low threshold (16 bytes default). The exact number of cycles during this burst is dependent on the latency of the ISA bus to the PCnet-ISA's mastership request and the speed of bus operation.

2.1.6 Bus Activity Timer

The bus activity timer allows timed DMA burst activity. It is useful when other devices are sensitive to being prevented access to the ISA bus. In the default configuration this feature is disabled. To activate the feature, the timer bit (CSR4) must be set and the appropriate value programmed into the bus activity timer register (CSR82).

The bus activity timer is gradually decreased every 100 ns while the PCnet-ISA asserts the master transfer in progress ($\overline{\text{MASTER}}$) signal. It must, therefore, be programmed with a value that gives the correct bus activity duration using the 100 ns multiplier. If a value of zero is programmed, the PCnet-ISA performs a single cycle. Care should be taken in the use of this register. While programming a reasonable value may allow more deterministic allocation of ISA bus bandwidth between devices, large values will certainly lead to problems. A maximum value of 6.55 ms can be programmed.

One useful way of using this timer is to ensure that refresh cycles can occur (approximately every 15 μs) even if the memory that the PCnet-ISA is reading or writng is adding additional wait states.

2.1.7 Programmable Transmit DMA Request

The transmit FIFO watermark (XMTFW, CSR80) sets the point at which the buffer management unit (BMU) requests the transfer of more data from the transmit buffers to the transmit FIFO. It specifies the point at which transmit DMA starts and stops, based upon the number of write cycles that could be performed to the transmit FIFO. Transmit DMA is allowed any time that the number of write cycles specified by XMTFW can be executed without causing transmit FIFO overflow. The XMTFW is set to a value of 00 after hardware reset, meaning 16 empty bytes (8 cycles) are needed to initiate DMA.

CSR80[9-8]	Write Cycles
00	8
01	16
10	32
11	Reserved

The PCnet-ISA BMU will not permit the transmit FIFO to overflow. In the default configuration, the burst size will be set to 32 bytes. If a DMA burst cycle is initiated when 16 bytes of space are available, the full 32 bytes may not be moved in the burst. The actual amount of data that will be moved is dependent on the delay in acquiring bus mastership and the number of bytes that have been transmitted on the network during the acquisition time. The PCnet-ISA suspends the DMA burst before the transmit FIFO overflows.

This feature can be used in conjunction with the programmable burst size or DMAPLUS to enlarge the size of the burst transfers and hence minimize the number of overall arbitration cycles for mastership. By programming the PCnet-ISA to wait longer (for more space to be available in the transmit FIFO), the burst size can be increased before the FIFO's high threshold is reached and the burst is suspended. However, it will also increase the exposure of the PCnet-ISA to underflow conditions if it cannot gain access to the ISA-bus. While the burst size is increased, the bus acquisition latency is reduced.

2.1.8 Programmable Receive DMA Request

The receive FIFO watermark (RCVFW, CSR80) sets the point at which the buffer management unit (BMU) requests the transfer of more data from the receive FIFO to the receive buffers. It specifies the point at which receive DMA is requested (in relation to the number of received bytes in the receive FIFO). The RCVFW specifies the number of bytes that must be present (once the packet is verified as a non-runt) before the receive DMA is requested. However, for receive DMA to be performed for a new frame, at least 64 bytes must be received. This effectively prevents runt receive frames and collisions during the slot time (512-bit times). If the runt packet accept feature is enabled, receive DMA is requested as soon as either the RCVFW threshold is reached or a complete valid receive frame is detected (regardless of length). The RCVFW is set to a value of 10 after hardware reset, meaning 64 bytes must be received to initiate DMA.

CSR80[13-12]	Bytes Received
00	16
01	32
10	64
11	Reserved

The PCnet-ISA BMU will not permit the receive FIFO to underflow. In the default configuration, the burst size is set to 32 bytes. If a DMA burst cycle is initiated when 16 bytes of received data are available, the full 32 bytes may not be moved in the burst. The actual amount of data that will be moved is dependent on the delay in acquiring bus mastership and the number of bytes that have been received from the network during the acquisition time. The PCnet-ISA will suspend the DMA burst before the receive FIFO underflows.

Programming the RCVFW to a lower value provides additional latency to acquire the bus before the receive FIFO can reach an overflow situation. However, even when the RCVFW is programmed below the 64-byte limit, each new packet is qualified to be a non-runt before the receive DMA request is issued. If the programmable burst size or DMAPLUS features are used in conjunction with the 64-byte threshold, the size of the receive burst transfers can be enlarged, thus minimizing the number of overall arbitration cycles for mastership. By keeping the default RCVFW, the PCnet-ISA waits until 64-bytes are available in the receive FIFO. The burst size can be increased before the low threshold of the FIFO is reached, and the burst is suspended.

2.1.9 Programmable Memory Read and Write Active Time

The PCnet-ISA allows the bus master read and write cycle times to be programmed. After the power-up and initialization process, the default timing for the $\overline{\text{MEMR}}/\overline{\text{MEMW}}$ command active time is 250 ns. This can be increased or reduced in 50 ns increments from 50 ns to 750 ns. Since the $\overline{\text{MEMR}}/\overline{\text{MEMW}}$ command inactive time is fixed at 100 ns, the read or write cycles can be programmed from 150 ns to 850 ns (350 ns default).

The feature is programmed through the ISA Bus Data Port (IDP) located in the I/O space of the device. The register address port (RAP) pointer must be initialized with a value of 0 to access the mastermode read (MMRD) location, and 1 to access the master mode write (MMWR) location.

MMRD/MMWR	Command Active Time (ns)	Command Inactive Time (ns)	Total Cycle Time (ns)
0000h	50	100	150
0001h	50	100	150
0002h	100	100	200
0003h	150	100	250
0004h	200	100	300
0005h	250	100	350
0006h	300	100	400
0007h	350	100	450
0008h	400	100	500
0009h	450	100	550
000Ah	500	100	600
000Bh	550	100	650
000Ch	600	100	700
000Dh	650	100	750
000Eh	700	100	800
000Fh	750	100	850

This feature is specifically advantageous in motherboard designs, where the ISA bus interface is captive and interoperability is not required over multiple-host systems, unlike an adaptor card. Reducing the command active time by just 50 ns increases the performance during burst operations by approximately 14%. The feature is generally to be avoided in adaptor card designs unless the driver setup procedure is written specifically to allow the installer or user to easily adjust the cycle time to his specific machine. The 350 ns default time was been chosen as a compromise to ensure maximum interoperability across ISA bus-compatible machines, while minimizing the bus occupancy of the device.

2.1.10 Pin-selectable 8- or 16-bit I/O

While transferring packet data, the PCnet-ISA has bus mastership, and hence, full control of the ISA bus timing. However, when being accessed as a slave device, to read or write internal CSRs the chip must react correctly to a host-initiated ISA bus cycle. For this reason hardware compatibility issues relating to bus mastership type devices are almost exclusively related to the use of slave cycles. Most of these incompatibility issues relate to the timing of 16-bit I/O cycles.

To avoid compatibility issues such as this, the PCnet-ISA chip supports 8-bit I/O cycles. These cycles have relaxed timing constraints due to the backward compatibility issue with older PC/XT type machines. The object of this feature is to maximize compatibility with minimal throughput impact. While the chip supports full 16-bit I/O cycles, it is recommended that the 8-bit mode be used by simply not connecting the $\overline{\text{IOCS16}}$ line to the ISA bus, but grounding it instead. If the $\overline{\text{IOCS16}}$ line is connected to the ISA bus, the PCnet-ISA will monitor it for activity. If it detects $\overline{\text{IOCS16}}$ go active, it will assume the system can support 16-bit I/O cycles and will respond accordingly. Detailed descriptions of the 8-bit and 16-bit cycles are available in the data sheet, so they are not reproduced here.

Once the PCnet-ISA has been initialized and the driver is running, very few accesses are required to the internal CSRs. If the driver was originally written for the LANCE, there are no accesses to the CSRs except CSR0 because all other CSRs are not readable unless the stop bit was set. The PCnet-ISA does allow access to additional internal CSRs, even when running. However, these are generally limited to network statistic counters, which can be configured to provide an interrupt when roll over occurs. Hence, the performance impact of running 8-bit I/O (in comparison with the large amount of data transferred using 16-bit bus mastership cycles) will not be measurable for most types of LAN traffic activity.

2.1.11 Transmit Padding

During transmission of a frame, the pad field can be inserted automatically. The auto pad transmit bit enables the automatic pad insertion feature (APAD XMT = 1). The pad field is inserted as the frame leaves the transmit FIFO, thus preserving FIFO space for additional frames. In addition, allowing the PCnet-ISA to perform the padding relieves the software of this overhead task and minimizes ISA bus bandwidth by avoiding the transfer of the pad characters.

Transmit frames are automatically padded to ensue 64 bytes of information, including destination, source, length, type, message data, and FCS fields (excluding preamble). This allows the minimum frame size of 64 bytes (512 bits) for 802.3/Ethernet to be guaranteed with no software intervention from the host system. The pad is placed between the LLC data field and FCS field in the 802.3 frame. The FCS is always added if the frame is padded, regardless of the state of the disable transmit FCS (DXMTFCS) bit. The transmit frame will be padded by bytes with the value of 00h.

Upper-layer software must correctly define the actual length field contained in the message to correspond to the total number of LLC data bytes encapsulated in the packet (length field as defined in the IEEE 802.3 standard). The length value contained in the message is not used by the PCnet-ISA to compute the actual number of pad bytes to be inserted. The PCnet-ISA appends pad bytes depending on the actual number of bits transmitted onto the network. Once the last data byte of the frame has completed and prior to appending the FCS, the PCnet-ISA checks to ensure that 544 bits have been transmitted. If not, pad bytes are added to extend the frame size to this value, and the FCS is then added.

The 544-bit count is derived from the following information.

Minimum frame size (excluding preamble, including FCS)	64 bytes	512 bits
Preamble/SFD size	8 bytes	64 bits
FCS size	4 bytes	32 bits

To be classed as a minimum-size frame at the receiver, the transmitted frame must contain the following bits.

Preamble + (Min Frame Size + FCS) bits

At the point that FCS is to be appended, the transmitted frame should contain the following bits.

Preamble + (Min Frame Size - FCS) bits

64   + (512 - 32) bits

A minimum-length transmit frame from the PCnet-ISA will, therefore, be 576 bits after the FCS is appended.

The Ethernet specification makes no use of the LLC pad field and assumes that minimum-length messages will be at least 64 bytes in length.

The APAD XMT (CSR4) controls programming of the automatic padding feature and must be static when the transmit function is enabled (TXON = 1, CSR0). The transmitter should be disabled before programming this option. The default value of APAD XMT will disable auto pad generation after hardware or software reset.

2.1.12 Receive Frame Stripping

During reception of a frame, the pad field can be stripped automatically. The auto strip receive bit enables the automatic pad stripping feature (ASTRP_RCV = 1). The pad field is stripped before the frame is passed to the receive FIFO, thus preserving FIFO space for additional frames. In addition, allowing the PCnet-ISA to perform the stripping relieves the software of this overhead task and minimizes ISA bus bandwidth by avoiding the transfer of the pad characters. The FCS field is also stripped because it is computed at the transmitting station based on the data and pad field characters and will be invalid for a receive frame that has the pad characters stripped.

The number of bytes to be stripped is calculated from the embedded length field (as defined in the IEEE 802.3 definition) contained in the packet. The length indicates the actual number of LLC data bytes contained in the message. Any received frame that contains a length field of less than 46 bytes will have the pad field stripped. Receive frames that have a length field of 46 bytes or greater will be passed to the host unmodified.

Because any valid Ethernet type field value is always greater than an normal 802.3 length field, the PCnet-ISA does not attempt to strip valid Ethernet frames.

For some network protocols the value passed in the Ethernet type and 802.3 length fields is not compliant with either standard and may cause problems.

The diagram below shows the byte and bit ordering of the received length field for an 802.3-compatible frame format.
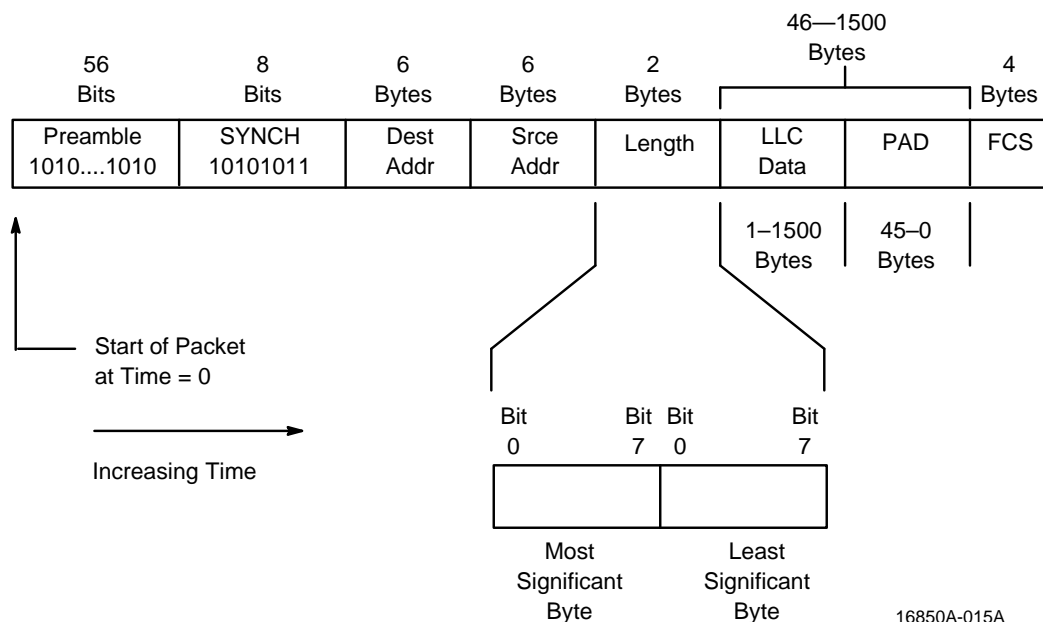

Figure 15. IEEE 802.3-Compatible Frame Format

The ASTRP_RCV in CSR4 controls programming of the automatic pad field stripping feature. The ASTRP_RCV must be static when the receive function is enabled (RXON = 1, CSR0). The receiver should be disabled before programming this option. The default value of ASTRP_RCV will disable auto pad stripping after hardware or software reset.

2.2 Network Interfaces

The following diagrams represent two typical applications of the PCnet-ISA. The first (see Figure 16) shows a 10BASE-T and AUI implementation; the second (see Figure 17) shows a 10BASE-T and 10BASE2 configuration. The PCnet-ISA provides flexibility in the selection of the network interface. After reset, the default configuration sets the external MAU select (XMAUSEL, ISACSR2), so the MAUSEL pin can be used to configure the port (MAUSEL = HIGH to select 10BASE-T). If the XMAUSEL bit is cleared, the selection of the port is controlled either automatically or under software control. If the auto select (ASEL, ISACSR2) bit is set, the active network interface is automatically determined. If the 10BASE-T port is in the link pass state ($\overline{\text{LED0}}$ is low, Link OK is on), the 10BASE-T port will be selected. If the 10BASE-T port is in the link fail state, the AUI port is selected. For forced software selection, the port selection (PORTSEL, CSR15) bits override the ASEL bit and forces the selection to the programmed choice.

In most cases the physical dimensions of the board and connector space dictate how many interfaces can be supported. However, because the 10BASE-T transceiver is integral to the part and the RJ45 connector is very small footprint, this offers the most economical interface in terms of space, power, and cost.

Most PC workgroup installations are either 10BASE2 (Cheapernet) or 10BASE-T (twisted-pair) based. Few PC installations use an AUI connected remote MAU if these interfaces are provided, because 10BASE5 installations are not commonly used in PC-based LANs. Because the AUI connector is the largest in terms of back panel footprint, providing 10BASE-T and 10BASE2 will serve a large proportion of the PC LAN community.

2.2.1 Twisted-Pair (10BASE-T) Interface

The integrated 10BASE-T transceiver makes the implementation of this interface extremely simple. The only components required are the transmit and receive termination resistors, a filter and transformer module, an RJ45 connector, and some additional decoupling capacitors. The filter and transformer module minimizes any potential electromagnetic interference and radio frequency interference (EMI/RFI) problems. The performance of these modules is governed to some extent by the 10BASE-T standard, but the overall EMI/RFI system performance is the responsibility of the manufacturer. Common-mode noise is the primary contributor to radiated energy from the twisted-pair interface. Additionally, there may often be significant common-mode power supply noise generated on the motherboard or adaptor by other devices, so filter and transformer modules that incorporate common mode chokes are recommended. In addition, some modules offer integrated transmit and receive termination resistors. A list of vendors and products is included in the Appendix.

The primary concerns are to minimize the length and cross-over connections between the PCnet-ISA chip, the filter and transformer module, and the RJ45 connector. The PCnet-ISA layout has been optimized to minimize any cross-over requirements with most types of filter and transformer module. Additional information is included in the Chapter 4.

The LED interface drivers provided by the PCnet-ISA allow up to four status indicators. For a 10BASE-T implementation, the link status (LNKST) provided by $\overline{\text{LED0}}$ is mandatory. The 10BASE-T standard recommends that the indicator be green. No other LEDs are mandatory, and use of additional indicators is largely dependent on mounting space and the perceived value that the indicator provides.

The NE1500T provides two LEDs, Link OK (LNKST, $\overline{\text{LED0}}$ default), and Polarity OK (RCVPOL, $\overline{\text{LED2}}$ default). The10BASE-T receiver of the PCnet-ISA has the ability to automatically detect and correct reversed polarity. This is the default configuration, which can be disabled by setting the disable automatic polarity correction (DAPC, CSR15) bit. The default configuration of $\overline{\text{LED1}}$ is to display Receive Activity (RCV) and $\overline{\text{LED3}}$ displays Transmit Activity (XMT). The default configuration of $\overline{\text{LED1}}$–3 can be reprogrammed.
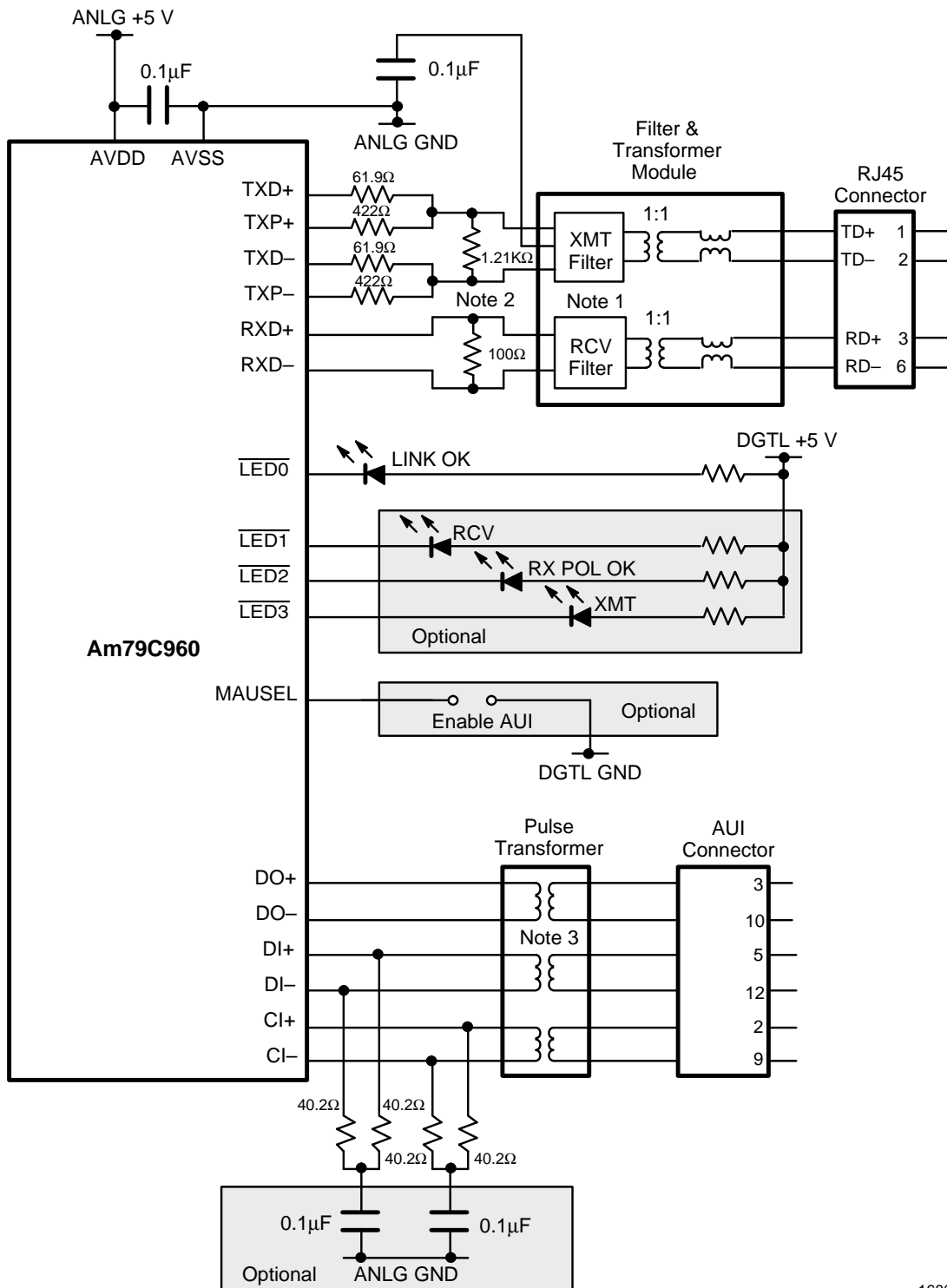
16805A-016A

Notes:

1. Compatible filter modules, with a brief description of package type and features are included in the Appendix.

2. The resistor values are recommended for general purpose use and should allow compliance to the 10BASE-T specification for template fit and jitter performance. However, the overall performance of the transmitter is also affected by the transmit filter configuration.

3. Compatible AUI transformer modules, with a brief description of package type and features are included in the Appendix.

Figure 16. 10BASE-T and AUI Implementation of PCnet-ISA
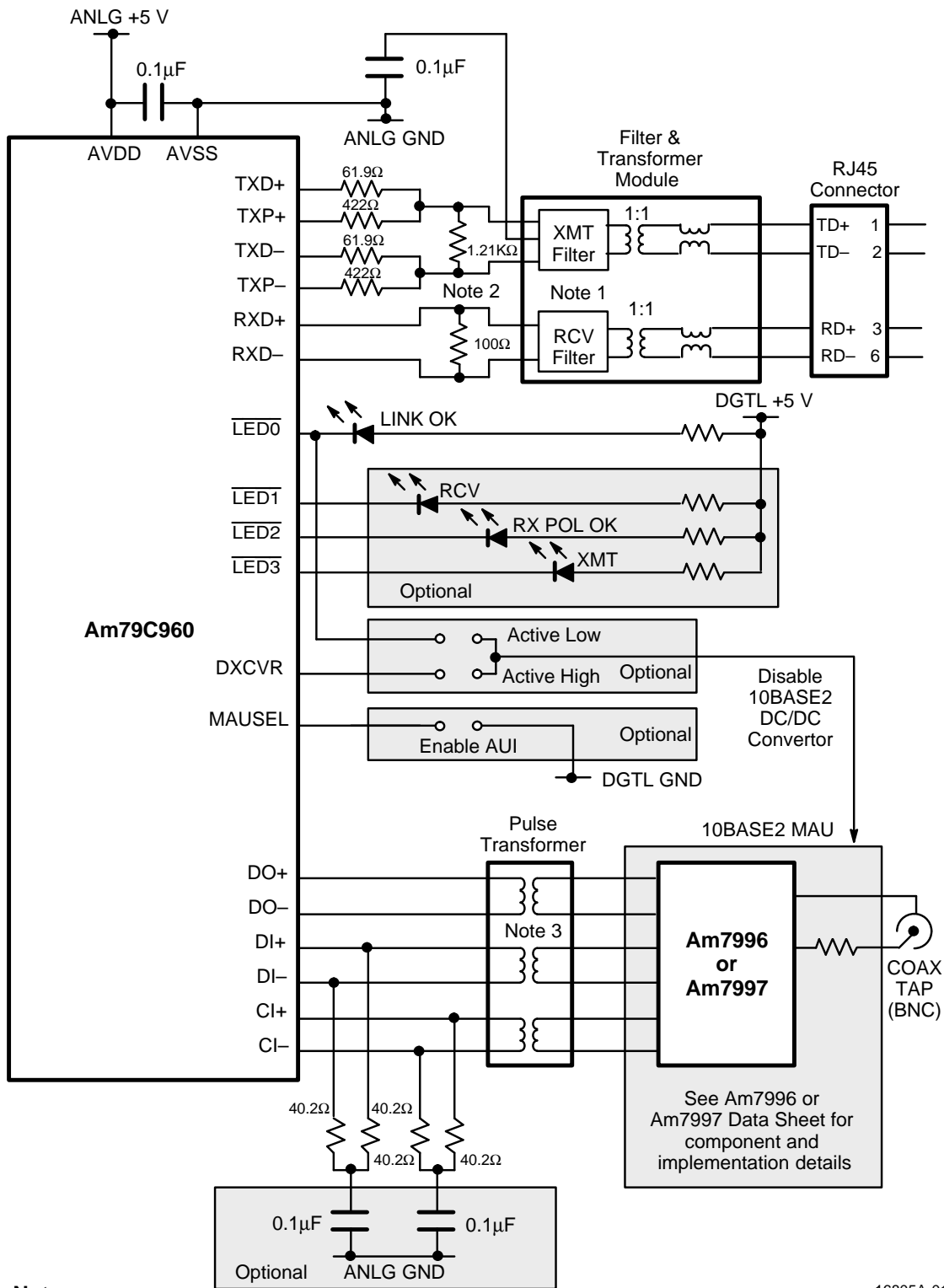
Notes:

1. Compatible filter modules, with a brief description of package type and features are included in the Appendix.

2. The resistor values are recommended for general purpose use and should allow compliance to the 10BASE-T specification for template fit and jitter performance. However, the overall performance of the transmitter is also affected by the transmit filter configuration.

3. Compatible AUI transformer modules, with a brief description of package type and features are included in the Appendix.

Figure 17. 10BASE-T and 10BASE2 Configuration of PCnet-ISA

2.2.2 Attachment Unit Interface

The PCnet-ISA chip directly supports a fully compliant AUI. The AUI can be used to interface to an external MAU and to allow connection to alternate media such as 10BASE2 (Cheapernet), 10BASE5 (Ethernet), or 10BASE-FL (fiber link). The circuit implementation is extremely simple, requiring a pulse transformer, the AUI connector, and some additional terminating and decoupling components. However, the AUI specification optionally requires that the external MAU be powered from the DTE (host) device. Although this is optional, virtually all AUI implementations provide the power source because of the cost and complexity of providing a separate supply to the MAU externally. The voltage plus pin (VP, pin 13) must provide between +12 V DC -6% to +15 V DC +5% at 500 mA. This frequently makes the AUI an unattractive interface to support in power-sensitive applications. For specific details of the power requirements, pin designations, and connector type, consult Section 7.5.2.5 to 7.6.3 in ISO 8802-3: 1990 (ANSI/IEEE Std 802.3).

The AUI isolation transformer must be provided to protect the PCnet-ISA chip from external fault conditions. The AUI specification requires that all drivers and receivers must tolerate a 16 V applied voltage between any of the inputs, outputs, and ground with no damage. The 5 V logic of the PCnet-ISA must be protected against this test voltage. A list of vendors and products is included in the Appendix.

2.2.3 Embedded 10BASE2 Medium Attachment Unit (MAU)

The PCnet-ISA can support an embedded 10BASE2 (Cheapernet) MAU using the AUI port. The same concerns apply to this application, namely power consumption, board and connector space, and cost. In this configuration the AUI is not externally exposed, so the power requiremnt to allow a remote MAU to operate from the AUI is reduced.

Typically, a 10BASE2 transceiver requires a –9 V power supply. Suitable +5 V DC to –9 V DC convertors are available from several vendors. Some of these provide an input pin that allows the convertor to be remotely powered down using a logic signal. This allows the convertor and the transceiver to be disabled when 10BASE-T is the active network port, which substantially reduces the overall system power requirements. The PCnet-ISA can provide either a logic high or low output to disable the convertor, as indicated in the previous diagram and as described below.

- Logic high indicates the external convertor should be turned off. The disable transceiver (DXCVR) output is used to indicate the active network port. A high level indicates the 10BASE-T port is selected and the AUI port is disabled (or $\overline{\text{SLEEP}}$ is active). A low level indicates the AUI port is selected and the twisted-pair interface is disabled.

- Logic low indicates the external convertor should be turned off. The LNKST ($\overline{\text{LED0}}$) output can be used to indicate the active network port. A high level indicates the 10BASE-T port is in the link fail state and the external convertor should be on. A low level indicates the 10BASE-T port is in the link pass state and the external convertor should be off.

An AUI isolation transformer must still be provided to protect the rest of the devices on the adaptor or motherboard from potential network faults on the coax. For full details of how to implement a 10BASE2 MAU with the Am7996/7 transceiver, consult the appropriate data sheet (Am7996, Publication #07506; Am7997, Publication #12473).

Because space is not usually critical in an adaptor design, the 10BASE2 MAU can easily be accommodated on an ISA bus form-factor card. However, space is extremely critical in a motherboard design, so it may be advantageous to keep the 10BASE2 MAU as a separate module. The MAU can be implemented on a very small add-in card, which is effectively mounted only by the bracket. The MAU would include the DC-to-DC convertor, the transceiver chip (Am7996/7), all additional external components, and the BNC connector. This also eliminates the need for a special "knock-out" in the PC case to allow mounting of the 10BASE2 BNC connector. The AUI signals and power to the MAU can be provided from the motherboard by a jumper cable that plugs into a header. Care should be taken to ensure that the cable used to connect the motherboard to the MAU provides adequate shielding of the AUI signals from external noise.

2.2.4 Automatic Selection

A highly desireable feature that is integrated into the PCnet-ISA is the ability to automatically determine which network connection is in use. When the auto select bit (ASEL, ISACSR3) is set, the 10BASE-T port is selected if it is in the link pass state or if the disable link test (DLNKTST, CSR15) bit has been set (effectively forcing the 10BASE-T port into link pass). If the 10BASE-T port enters the link fail state because of receive inactivity, the PCnet-ISA automatically switches to the AUI port for activity.

CHAPTER 3
Power Saving and Security Features

3.1 Power Saving

The PCnet-ISA supports two hardware power-savings modes. Both are entered by driving the $\overline{\text{SLEEP}}$ pin LOW. In addition, the PCnet-ISA offers a third, reduced-power mode of operation using the External Address Detection Interface™ (EADI), although this is not the primary purpose of the feature. See the Security section later in this chapter for details on EADI operation and its use.

3.1.1 Sleep

In **sleep** mode the PCnet-ISA goes into a deep sleep with no support to automatically wake itself up. Sleep mode is enabled when the AWAKE bit in the ISA bus configuration register (ISACSR2) is reset. This mode is the default power-down mode.

3.1.2 Auto-Wake

In **auto-wake** mode, enabled by setting the AWAKE bit in ISACSR2 and driving the $\overline{\text{SLEEP}}$ pin LOW, the twisted-pair media attachment unit receive circuitry remains enabled, even while the $\overline{\text{SLEEP}}$ pin is driven LOW. The $\overline{\text{LED0}}$ output continues to function, indicating a good 10BASE-T link if link beat pulses or valid packet data are present. This $\overline{\text{LED0}}$ pin can be used to drive a LED and external hardware that directly controls the deactivation of the $\overline{\text{SLEEP}}$ pin. This configuration can be used to auto-wake the PCnet-ISA or a part or all of the host system. Control of reactivation of the various system parts (when there is any activity on the 10BASE-T link) are left to the designer, which provides the maximum flexibility for product differentiation. Auto-wake mode can be used only if the T-MAU is the selected network port.

3.2 Remote Wake (Security)

The External Address Detection Interface (EADI) can be used to implement alternative address recognition schemes outside the PCnet-ISA to complement the physical, logical, and promiscuous detection supported internally. This interface is usually provided to allow external perfect address filtering. This feature is typically utilized for bridge, router, or terminal server-type products. However, in the case of the PCnet-ISA it is intended to be used primarily a network security feature.

External logic is required to capture the serial bit stream from the PCnet-ISA and compare it with a table of stored addresses or identifiers. The address matching and the support logic necessary to capture and present the relevant data to the external table of address is application-specific.

As the entire 802.3 packet after start frame delimiter (SFD) is made available, recognition is not limited to the destination address, type (Ethernet), or length (802.3) fields. Internetworking protocol recognition can be performed on specific header or LLC information fields. The diagram below shows the block diagram of an implementation to store and detect multiple addresses in an external content-addressable memory (CAM) device.

3.2.1 EADI - Standard Operation

The EADI interface operates directly from the non-return to zero (NRZ) decoded data and clock recovered by the Manchester decoder. This allows external address detection to be performed in parallel with frame reception and address comparison in the MAC station address detection (SAD) block.

The serial receive data clock (SRDCLK) pin allows clocking of the receive bit stream from the PCnet-ISA into the external address detection logic. SRDCLK runs only during frame reception activity. Once a received packet commences and data and clock are available from the decoder, the EADI logic monitors the alternating (1,0) preamble pattern until the two ones of the start frame delimiter (1,0,1,0,1,0,1,1) are detected, at which point the start frame/byte delimeter (SF/BD) output is driven high.

After SF/BD is asserted, data from the serial receive data (SRD) pin should be deserialized and sent to the CAM or other address detection device.
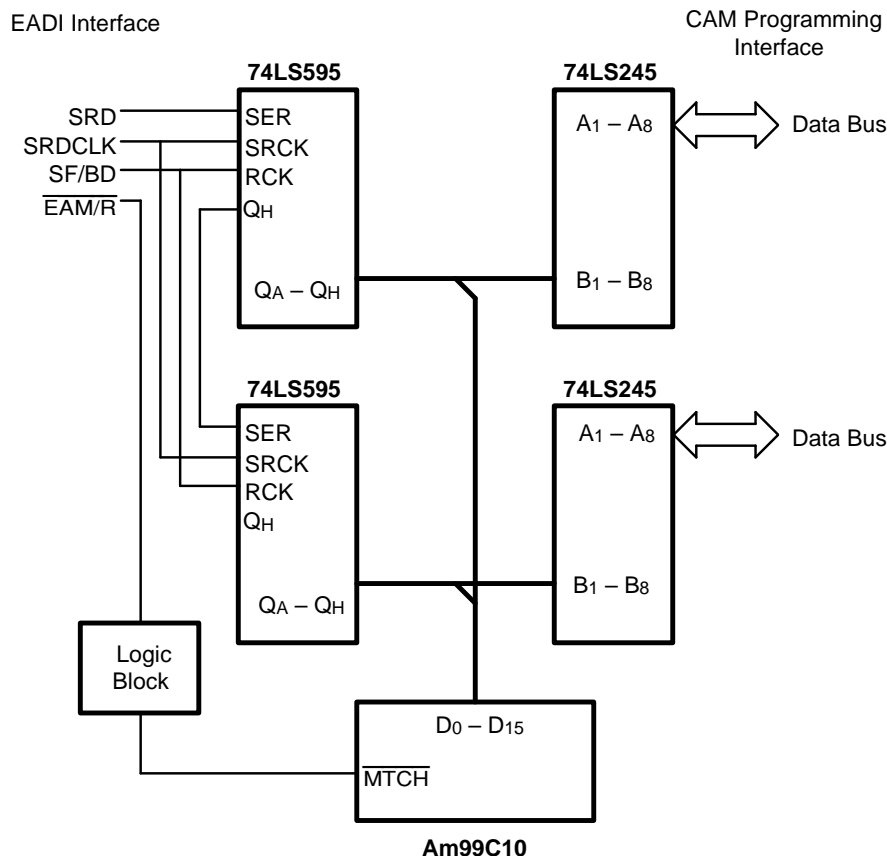
Figure 18. EADI - Simple External CAM Interface

To allow simple serial-to-parallel conversion, SF/BD serves as a strobe or marker to indicate the delineation of bytes subsequent to the SFD. This provides a mechanism that allows not only capture and decoding of the physical or logical (group) address, but also facilitates the capture of header information to determine protocol and internetworking information. The external address reject ($\overline{\text{EAR}}$) pin is driven by the external address comparison logic to reject the incoming packet.

The PCnet-ISA can be configured with both physical and logical address fields operational. If an internal address match is detected by comparison with either the physical or logical address field, the packet is accepted regardless of the condition of $\overline{\text{EAR}}$. Incoming packets that do not pass the internal address comparison continue to be received by the PCnet-ISA. The $\overline{\text{EAR}}$ signal must be externally presented to the PCnet-ISA prior to the first assertion of DMA request (DRQ) signal to guarantee rejection of unwanted packets. This allows approximately 58-byte times after the last destination address bit is available to generate the $\overline{\text{EAR}}$ signal, assuming the PCnet-ISA is not configured to accept runt packets. The $\overline{\text{EAR}}$ signal will be ignored by the PCnet-ISA from 64-byte times after the SFD, and the packet will be accepted if $\overline{\text{EAR}}$ has not been asserted before this time. If the

PCnet-ISA is configured to accept runt packets, the $\overline{EAR}$ signal must be generated prior to the receive message completion, which could be as short as 12-byte times (assuming 6 bytes for source address, 2 bytes for length, no data, and 4 bytes for FCS) after the last bit of the destination address is available. The $\overline{EAR}$ signal must have a pulse width of at least 200 ns.

Setting the PROM bit will cause all receive packets to be received, regardless of the programming of the $\overline{EAR}$ input. The table below summarizes the operation of the EADI features.

PROM	$\overline{EAR}$	Required Timing	Received Messages
1	X	No timing requirements	All Received Frames
0	H	No timing requirements	All Received Frames
0	↓	Low for 200 ns within 512 bits after SFD	Physical/Logical Matches

Internal/External Address Recognition Capabilities

3.2.2 EADI - Security Operation

The EADI will operate as long as the start (STRT) bit in CSR0 is set, even if the receiver or transmitter is disabled by software (disable transmit (DTX) and disable receive (DRX) bits in CSR15 set). This can be useful as a power-down mode in that the PCnet-ISA will not perform any DMA operations, thus saving power by not utilizing the ISA bus drivers. However, the internal operation of the PCnet-ISA continues normally so packet reception will occur, as will serial repeating of the packet data and receive clock over the EADI. External circuitry can be implemented to interface to EADI pins and wait for a match based on specific receive frame contents (versus address alone). This allows the PCnet-ISA and associated external logic to respond to specific frames on the network and thereby facilitate remote node and controlled node operation.

Some emerging network management protocols such as the 3Com/IBM HLM Specification[10] are already implementing the requirement to enable remote control of the node. For applications requiring network-controlled power or security management, the EADI of the PCnet-ISA offers a unique solution. Not only can the station be remotely powered-down, but segregated power supply distribution, the PCnet-ISA, and its associated EADI logic, can remain active, effectively waiting for a command to reactivate the station.

Because the logic for "snooping" of the receive packet and the power management distribution are under the control of the system designer, the end station can implement security by disabling the keyboard entry or display system or powering-down of non-critical functions.

---

[10] IBM/3Com Heterogeneous LAN Management: Application Program Interface: Technical Reference, version 1.0, June 1991.

CHAPTER 4
Considerations for
PC Motherboard Layout

The Am79C960 PCnet-ISA device implements the functions of an Ethernet node controller, a serial interface adapter, a 10BASE-T transceiver, and an Industry Standard Architecture (ISA) bus interface, leaving the designer relatively few tasks to complete an Ethernet design. Still, there are some precautions the designer should consider when designing Ethernet on the motherboard. These guidelines will help speed the process of creating a functional node that complies with electromagnetic interference and radio frequency interference (EMI/RFI) standards.

The most important issue to consider when adding Ethernet to a PC motherboard is noise. Any noise that is capacitively coupled to the PCnet-ISA circuitry has the potential to be transmitted onto the network. When this happens, the network media becomes an antenna, radiating high-frequency noise that may violate FCC (Federal Communications Commission), VDE, or other regulatory specifications. Whether or not Ethernet functionality is disrupted by this radiation, a design that violates EMI/RFI emission regulations should be avoided. Therefore, the purpose of most layout precautions is to isolate the Ethernet circuitry from noise.

4.1 Signal Routing and Ethernet Placement

The area of Ethernet circuitry on the motherboard should be kept free of interference from unrelated signal traces. Thus, the space surrounding the grouped Ethernet components should be considered off limits when routing other signal paths. To minimize the burden this places on the rest of the motherboard layout, Ethernet circuitry should be placed together at the perimeter of the board, ideally near a corner. This makes sense because the PCnet-ISA device should reside near its physical media connector on the backplane.

Consideration should also be given to the placement of Ethernet circuitry relative to other noisy devices on the board. The Am79C960 and the crystal oscillator, in particular, house critical timing mechanisms that are sensitive to noise. Obvious sources of noise such as the CPU, clock, cache controller, and numeric coprocessor should be placed some distance away from these components. This restriction should not cause serious complications because these devices are typically located elsewhere on the motherboard.

4.2 Ground and Power Planes

Once the Ethernet functional block has been placed, it should be isolated from noise within the ground and power planes. These layers of the motherboard should be specially etched, creating noise "fences" around some areas and noise-free "islands" beneath others (see Figure 19).

Ground
& Power
Planes

Am79C960
PCnet-ISA

Etched Areas
of Ground
& Power
Planes

Crystal
Oscillator

Transformer/
Filter Module

10BASE-T
Connector

16850A-019A
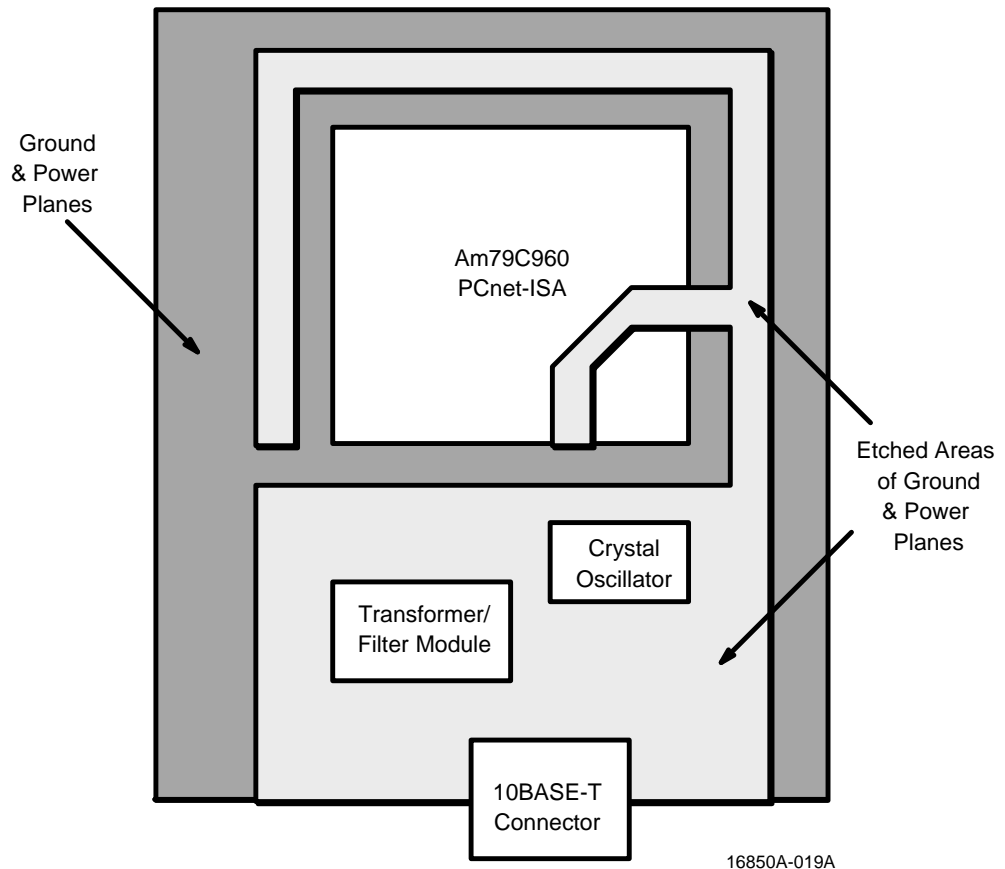
**Figure 19. Recommended Ground and Power Plane Etching Scheme
for 10BASE-T Components**

An etch that almost completely surrounds the group of Ethernet components should be created in the ground and power planes. The opening in this barrier should be oriented in such a way that the noise produced by the rest of the board does not have a straight path into the isolated area. This barrier effectively prevents noise on the motherboard from being transmitted to the network.

Due to the modular architecture of the Am79C960, all analog functionality is contained on one corner of the device. As a precaution against cross-coupled noise between the analog and digital circuitry, another gap in the ground and power planes should be etched to isolate this area from the rest of the chip. Approximately one quarter of the chip should be delineated across the corner nearest pins X1 and X2. This trace will protect the analog circuitry from digital interference.

Additionally, an island should be etched into the ground and power planes to provide a space for the media-specific interface components. For 10BASE-T media access, these components include the filter and transformer module and twisted-pair wire jack. For 10BASE2 access, they inlude the DC-to-DC converter, isolation transformer, Am7996 tap tranceiver, and coax connector. For 10BASE5, this includes only the AUI connector. It is important that these devices not be placed over a ground or power plane, as noise is easily coupled from the ground and power planes through these components and onto the network.

4.3 Dedicated Ethernet Components

Because real estate on the motherboard is at a premium, designers might be tempted to economize by using some components for dual purposes. While it might be obvious that this is not a good idea, the message bears repeating.

4.3.1 Crystal Oscillator

Crystal oscillators are common elements of a PC motherboard. However, using a crystal that already fulfills another purpose to implement the Ethernet timing circuitry is not a good idea. Timing on the network is strictly regimented by the Ethernet/IEEE 802.3 standard. The crystal required for Ethernet must be 20 MHz and must not vary by more than 100 parts per million, or 0.01%. Such a crystal would probably not be cost-effective for other motherboard applications. In addition, a crystal specified to perform at this level may not achieve this degree of accuracy in the actual circuit. Routing long, multi-purpose signal traces to and from such a critical circuit would likely disrupt its performance.

It is highly recommended that a high-performance crystal be dedicated to the isolated Ethernet circuitry. Crystals from different vendors should be tested in the actual circuit, as models from some manufacturers vary in frequency more than others. Different crystal device models may require different tuning among the surrounding capacitors. However, once the capacitors have been tuned for the specific crystals used for production, the design does not need to be altered on a board-by-board basis.

4.3.2 Ethernet Address PROM

Another recommended element for designing Ethernet on the motherboard is a stand-alone address PROM. It is possible to store the unique address of the Ethernet node in other memory devices such as the BIOS PROM or system memory. However, to be compatible with Novell NetWare, the Ethernet address must be stored in I/O addressing space rather than memory addressing space. Thus, unless a designer chooses to write a driver specific to his or her hardware, it is recommended that the stand-alone address PROM be used to ensure Novell NE2100 software compatibility.

4.4 Bypass Capacitors

Bypass capacitors should accompany all power and ground pins on the Am79C960 device. However, the common practice of connecting these capacitors to the power and ground planes near the pins should be avoided, as it introduces unwanted inductance that decreases the effectiveness of the capacitors. Instead, the bypass capacitors should be connected directly to the Am79C960 pins. The pins should then be connected to power and ground with feed-through holes (see Figure 20). Ideally, two feed-hrough connections should be used if space permits. The feed-through holes may be placed either at the pin or behind the capacitor.

Connection   Feed-through holes

16850A-020A

Correct   **Incorrect**
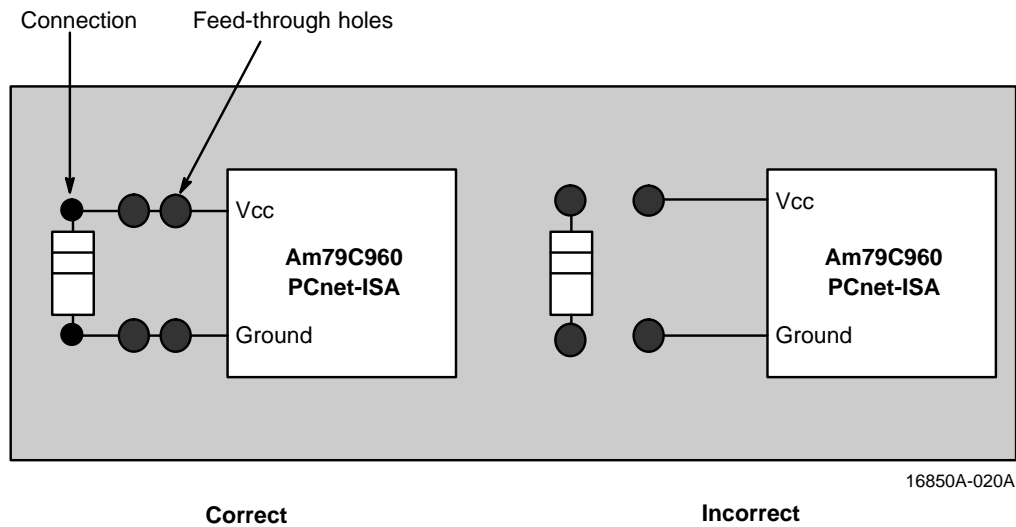
Figure 20. Recommended Connections for Bypass Capacitors

4.5 Other Considerations

When implementing any design, the engineer should employ good board layout practices. Keep all signal traces as short and as straight as possible. In 802.3/Ethernet, nodes may be separated by up to 100 m on 10BASE-T. The signals received from the network travel a long, difficult path before reaching the motherboard, and so they may be noisy and low in amplitude. To avoid jeopardizing the data further, these signals should be routed by the most direct path from the system's physical connector to the Am79C960 PCnet-ISA device.

CHAPTER 5
Software Considerations

5.1 NE2100 Compatibility

Network software written for the LANCE-based NE2100 board should run on a PCnet-ISA–based board without modifications with one exception. Loopback diagnostics written for the NE2100 may fail because, by default, the PCnet-ISA rejects packets shorter than 64 bytes, even in loopback mode.

The following is a list of differences between the PCnet-ISA and the NE2100 boards from a programmer's viewpoint. Subsequent paragraphs give more details about the items in this list.

- External loopback rejects packets smaller than 64 bytes unless runt packet accept is enabled.
- The byte swap (BSWP), ALE control (ACON), and byte control (BCON) bits (bits 2, 1, and 0, respectively) in the control and status register (CSR3) are not used.
- CSR3 is always available, not just when the stop bit is set.
- Several bits that are reserved in the LANCE are used in the PCnet-ISA.
- Setting the stop bit in the PCnet-ISA is not equivalent to hardware reset.
- Reset does not clear the register access port.
- Memory error (MERR) does not turn off the transmitter and receiver in the PCnet-ISA.
- The PCnet-ISA does not require an I/O write after an I/O read to finish a software reset.
- All internal registers can be accessed by software.
- There are no restrictions on the minimum size of transmit or receive buffers.
- Interrupt bits can be masked independently.
- Cyclic reduncy check (CRC) generation can be omitted for individual frames.
- Pad bytes for short packets can be added and stripped automatically.
- There is a test mode option to accept runt packets.
- There is a chip ID register.
- The widths of the read and write signals are programmable.
- The PCnet-ISA contains an automatic wake up circuit that is enabled by software.
- The idle state of the attachment unit interface (AUI) drivers is programmable by software.
- The choice of AUI or twisted-pair media attachment unit (T-MAU) is programmed or determined automatically.
- The 10BASE-T link test can be disabled by software.
- The automatic polarity correction feature of the T-MAU can be disabled by software.
- The software can select one of two T-MAU receiver threshold voltages.
- The meaning of three LED driver outputs is programmable by software.
- External address detection must be enabled by software.
- The PCnet-ISA can be initialized by writing directly to registers.
- The number of descriptors can be anything between 1 and 65535.
- The PCnet-ISA has two groups of registers that are accessed through two different ports.
- The initialize (INIT) and start (STRT) bits in CSR0 can be set at the same time.
- The PCnet-ISA can interrupt the processor when it starts to transmit a packet.
- The reception of broadcast packets can be disabled.

- Physical address detection can be disabled.
- Internal loopback can include or exclude the Manchester encoder/decoder (MENDEC).
- FIFO watermarks can be changed.
- The maximum length of direct memory access (DMA) bursts is programmable.
- The polling of transmit descriptors can be disabled.
- The transmission two-part deferral algorithm can be disabled.
- An alternate backoff algorithm can be selected.

5.2 PCnet-ISA Changes

The following describes functions that are available in the LANCE but behave differently in the PCnet-ISA without affecting the software.

5.2.1 Loopback

Loopback software written for the LANCE will fail on the PCnet-ISA unless runt packet accept can be enabled before the software is run. If runt packet accept, the default, is disabled, all packets of fewer than 64 bytes will be rejected, even in loopback mode.

The runt packet accept mode can be set up by a separate program that is executed before the LANCE software is loaded. This should make the LANCE loopback software work as long as the LANCE software does not cause a hardware reset. A hardware reset puts the PCnet-ISA back into its default state, in which the runt packet accept mode is disabled. Setting the stop bit does not affect the runt packet accept mode.

5.2.2 Register Differences

5.2.2.1 CSR3

The BSWP, ACON, and BCON bits are not used. Writing to these positions has no effect. These bits in the LANCE are used for configuring the bus interface. In the PCnet-ISA hardware reset makes the device compatible with the NE2100.

In the PCnet-ISA CSR3 is always available, not just when the stop bit is set.

5.2.2.2 Reserved Bits

Several bit positions reserved in the LANCE are used in the PCnet-ISA to enable features not implemented in the LANCE. NE2100 software that writes zeros to these positions will run properly on the PCnet-ISA without modifications.

The following bits have been added to CSR3: babble mask (BABLM), missed-frame mask (MISSM), memory error mask (MERRM), receive interrupt mask (RINTM), transmit interrupt mask (TINTM), initialization done mask (IDONM), disable transmit two-part deferral (DXMT2PD), and enable modified back-off algorithm (EMBA).

The following bits have been added to the mode register: disable receive broadcast (DRCVBC), disable receive physical address (DRCVPA), disable link status (DLNKTST), disable automatic polarity correction (DAPC), MENDEC loopback mode (MENDECL), low receive threshold (LRT), transmit mode select (TSEL), and port select (PORTSEL [1:0]).

To add frame check sequence to an existing setup, the ADD_FCS bit has been added to the TMD1 field of the transmit descriptor.

5.2.3 Stopping the PCnet-ISA

Setting the stop bit in the PCnet-ISA is not equivalent to hardware reset, as it is with the LANCE. Setting the stop bit clears the appropriate status bits, but it does not change any bits or registers that affect the configuration of the device.

The stop bit resets and disables the transmit and receive state machines. It resets the internal FIFO and descriptor pointers, so any data queued up in the FIFOs are lost and the internal descriptor pointers point to the first descriptors in the transmit and receive descriptor rings.

If the stop bit is set in the middle of a transmit or receive operation, the PCnet-ISA will complete the current DMA burst, but it will not issue a TINT or RINT interrupt before it turns off the receiver and transmitter. The PCnet-ISA does not violate the OWN bit convention when the stop bit is set. If an OWN is set, the corresponding buffer has not been completely filled by the receiver or transmitted by the transmitter.

Before the software sets the STRT bit, it should finish processing any packets that it has already received, reinitialize its descriptors, and queue up for transmission any outgoing packets that did not get sent before stop was set. However, the software does not have to reinitialize any configuration registers in the PCnet-ISA.

5.2.3.1 Register Access Port

The contents of the register address port (RAP) are not changed by setting the stop bit or by hardware reset. This register powers up in an unknown state. Any LANCE software that expects the register access port to be cleared by reset must be modified to work with PCnet-ISA.

5.2.4 Errors

The MERR bit does not turn off the transmitter and receiver in the PCnet-ISA. This is unlikely to be a problem with software written for the LANCE.

5.2.5 Software Reset

The method of generating a hardware reset for the PCnet-ISA is simpler than that used for the NE2100 board. For the NE2100 a read from the I/O port at I/O base address + 20 (hex) starts the reset pulse, and a write to any I/O port ends the pulse. For the PCnet-ISA it is only necessary to read from the port. No I/O write is needed.

5.3 PCnet-ISA Enhancements

This section describes programmable functions of the PCnet-ISA that are not available in the LANCE. The software controls these new functions by writing to a collection of registers that are not present in the LANCE and by writing to certain bits that are not used in LANCE registers.

5.3.1 General

5.3.1.1 Accessing Internal Registers

Compared to the LANCE the PCnet-ISA has many more registers that can be accessed by software–more than 70, in fact. Fortunately, most of these registers are never touched by normal networking software. They are used for debugging and production testing only.

The PCnet-ISA registers are divided into two groups: control and status registers (CSRs) and ISA Configuration Registers (ISACSRs). Accessing either group is a two step procedure. First, the software must load the register address port (RAP) with the number of the desired register. Then it can read from or write to the register data port (RDP) to access one of the CSRs or the ISA bus data port (IDP) to access one of the ISACSRs.

The register access port is located at the I/O base address + 12 (hex), the register data port is at I/O base + 10 (hex), and the ISA bus data port is at I/O base + 16 (hex).

5.3.1.2 Buffer Sizes

The PCnet-ISA places no restrictions on the minimum size of transmit or receive buffers. This means that a packet can be made up of one or more small buffers containing protocol headers plus one or more large buffers containing data. These buffers are linked together via descriptors in the transmit descriptor ring.

5.3.1.3 Masking Interrupt Bits

The PCnet-ISA adds the capability of masking out individual sources of interrupt. By setting the appropriate mask bits, the software can allow some conditions to cause interrupts while disabling other classes of interrupt.

5.3.1.4 Dynamic FCS Generation

Frame check sequence (FCS) generation can be enabled or disabled on a packet-by-packet basis. This feature is useful for bridge applications in which a packet should be passed from one network to another while preserving the original FCS.

To enable this feature, set the disable transmit FCS (DXMTFCS) bit in the mode register, then for each packet that should be sent with FCS generation, set the ADD_FCS bit in the TMD1 field of the first descriptor that points to that packet . (If the packet occupies more than one buffer, the ADD_FCS bit need only be set in the first descriptor.) The DXMTFCS bit disables FCS generation in general, while the ADD_FCS bit overrides DXMTFCS for an individual packet.

5.3.1.5 Automatic Pad Generation and Stripping

To simplify driver software and to save some bus bandwidth, the PCnet-ISA can be programmed to add pad bytes to all outgoing packets shorter than 64 bytes and to remove pad bytes from incoming packets. Automatic pad generation is enabled by setting the automatic pad transmit (APAD_XMT) bit in CSR0, and automatic pad stripping is enabled by setting the automatic strip receive (ASTRP_RCV) bit in the same register.

The PCnet-ISA uses the length field of the IEEE 802.3 header to determine how many bytes to add or strip. A major difference between the Ethernet and IEEE 802.3 specifications is that Ethernet uses the two bytes after the source address field for a type field rather than for a length field. For this reason a driver that handles Ethernet headers should not use the automatic padding or stripping features.

5.3.1.6 Missed Packet Count

A missed-packet counter (MPC) has been added to the PCnet-ISA to record the number of packets addressed to this node that were not saved because the receiver was disabled or because the FIFO overflowed. This 16-bit counter is cleared by setting the stop bit or by using the hardware reset. This counter can be read at any time, but it cannot be cleared by any software action other than setting the stop bit. When the MPC rolls over from 65535 to 0, the PCnet-ISA sets the missed-packet counter overflow (MPCOM) bit in CSR4 and issues an interrupt signal if this interrupt is enabled (IENA = 1 in CSR0 and MPCOM = 0 in CSR4).

This counter eliminates the need for the missed-packet (MISS) bit in CSR0, but it is included for compatibility with the LANCE. The MISS bit causes an interrupt every time a packet is missed, while the MPC causes an interrupt only after 65536 packets have been missed.

The software can extend the effective length of this counter by maintaining the high-order bytes of an extended counter in computer memory. The high-order bytes should be incremented by the interrupt service routine when an MPCO interrupt occurs.

Because the MPC cannot be reset by software, the software can calculate the number of packets missed because the last time the counter was read by recording the initial value of the counter and subtracting this number from the current count.

The MPC continues to count missed packets while the receiver is disabled. The user can use this feature to create a network traffic monitor. If the promiscuous and disable receiver bits are set in the mode register, the missed-packet counter will keep an accurate count of the total number of packets transmitted over the network.

5.3.1.7 Receive Collision Count

The PCnet-ISA contains a receive collision counter that counts all collisions detected while the device is not transmitting. This 16-bit counter (CSR114) is cleared by setting the stop bit or by using the hardware reset. The counter is never disabled, regardless of the state of the receiver. It is read only and can be read at any time, but it cannot be changed by any software action other than setting the stop bit.

5.3.1.8 Accepting Runt Packets

The PCnet-ISA can be made to accept properly addressed runt packets through the use of a test mode. To enable this feature, first write a one to the enable test (ENTST) bit of CSR4 to enable the buffer management unit (BMU) test register (CSR124). Then set the runt packet accept (RPA) bit in the BMU test register. Finally, write a zero back to the ENTST bit of CSR4 to disable further accesses to the BMU test register. Networking software should never set any bits in the BMU test register other than the RPA bit, because these other bits put the device into modes that are useful only in an IC tester.

5.3.1.9 Chip ID

The PCnet-ISA has a 32-bit chip ID register (CSR88) that contains the manufacturer's JEDEC ID code, the part number, and the version number. The JEDEC manufacturer's code for Advanced Micro Devices Inc. is 00000000001 (binary), the part number for the PCnet-ISA is 0003 (hex), and the version number is 0. Therefore the contents of this 32-bit register are fixed at 0000 3003 (hex). The software can use the information in this register to verify that it is working with the right part.

5.3.2 Bus Interface

5.3.2.1 Bus Timing

The widths of the read and write signals put out by the PCnet-ISA during a DMA transfer are programmable via the ISA bus configuration registers. The value in the master mode read active (MSRDA) register controls the width of the active portion of the read signal, while the value in the master mode write active (MSWRA) controls the width of the active portion of the write signal. Both values are interpreted as 50 ns increments.

5.3.2.2 Base Address Programming

The base I/O address of the PCnet-ISA registers is selected by the two I/O address map (IOAM0-1) pins. The four possible I/O base addresses are (in hex) 300, 320, 340, and 360. The software cannot read the values of these pins. However, it can find the I/O base address indirectly by using each possible value in turn as the base address, then trying to read the chip ID register. If one of these addresses results in the expected chip ID number, then it is extremely likely that that address is the right one to use.

5.3.2.3 Automatic Wake-Up Support

The PCnet-ISA contains an auto-wake circuit that can be programmed to remain active while the PCnet-ISA is in the low-power sleep mode. When enabled via the AWAKE bit in the ISA bus miscellaneous configuration register, this circuit monitors the 10BASE-T receiver for link test pulses. When it detects a valid link test sequence, it turns on the $\overline{\text{LED0}}$ pin, which with external logic can be used to wake up the rest of the chip.

5.3.3 Network Interface

5.3.3.1 TSEL

The state of the attachment unit interface (AUI) drivers during idle is programmable through the transmit mode selection (TSEL) bit in the mode register. TSEL should be set to zero (the default) for IEEE 802.3 or Ethernet 2 transformer coupled networks. It should be set to one for direct-coupled Ethernet 1 networks.

5.3.3.2 AUI or T-MAU

The PCnet-ISA offers three ways of selecting between the two available network interfaces (AUI or T-MAU). The choice can be made by software or by hardware, or it can be made automatically by the link pulse detection logic. The external MAU selection (XMAUSEL) bit in the ISA miscellaneous configuration register selects between hardware and software. If this bit is set, the selection is made by the MAU selection (MAUSEL) pin. Otherwise the two port selection (PORTSEL) bits in the mode register make the choice. The only valid choices for the PORTSEL bits are 00 to select the AUI or 01 to select the T-MAU.

However, if the auto select (ASEL) bit in the ISA miscellaneous configuration register is set, the state of the XMAUSEL bit is ignored, and the selection is made automatically. If the T-MAU detects valid link pulses on its receive data (RXD) inputs, the T-MAU is selected. Otherwise, the AUI is selected.

5.3.3.3 T-MAU Control

Jabber
If the T-MAU detects a jabber condition, it will set the JAB bit in CSR4, thus causing an interrupt. Setting the jabber mask (JABM) bit in CSR4 will mask out this interrupt.

Link Test
Setting the DLNKTST bit in the mode register (CSR15) will prevent the PCnet-ISA from monitoring link pulses. This feature is used for twisted-pair networks that do not use link pulses.

Polarity
Setting the disable automatic polarity correction (DAPC) bit in the mode register (CSR15) disables the automatic polarity correction. The actual polarity as detected by the T-MAU receiver regardless of the state of the DAPC bit can be read in the following rather roundabout way: One of the three LED drivers can be programmed to indicate the actual polarity detected by the T-MAU. The processor can test the state of this LED driver by reading the LED output (LEDOUT) bit of the corresponding LED register.

Receive Threshold
The receiver threshold voltage can be lowered so that the PCnet-ISA can be used with twisted-pair cables that are longer than the maximum specified in the 10BASE-T standard. To enable this feature set the LRT bit in the mode register (CSR15).

5.3.3.4 Programmable LEDs
The PCnet-ISA has four LED driver output pins. One of these ($\overline{LED0}$) displays the T-MAU link status. $\overline{LED0}$ is not programmable. The other three LED drivers can be programmed to display the status of one or more of the following internal signals from the T-MAU: collision, jabber, link, receive status, receive polarity, and transmit status.

If one LED driver is programmed to display more than one signal, the output will be the logical OR of the selected signals. For example, if the receive (RCVE) bit is set in the LED1 (ISACSR5) register, the $\overline{LED1}$ pin will be active whenever the MAU is receiving a packet. If the RCVE and transmit (XMTE) bits in the LED1 register are both set, the $\overline{LED1}$ pin is active whenever the MAU is receiving or transmitting.

Since network events are extremely short, each of the three programmable LED driver circuits can be connected to pulse stretcher circuits that will cause the LEDs to be turned on long enough for humans to be able to notice them. Each pulse stretcher is enabled by setting the pulse stretcher (PSE) bit in the appropriate LED register (ISACSR 5-7).

The state of an LED driver signal before the pulse stretcher can be read at any time by examining the LEDOUT bit of the appropriate LED register. This technique can be used to test the polarity of the twisted-pair receiver inputs (normal or reversed).

5.3.3.5 External Address Detection
If external hardware is used to make the PCnet-ISA respond to more than one physical address, the EADI™ selection (EADISEL) bit in the ISA bus miscellaneous configuration register must be set. This bit causes the $\overline{LED1-3}$ and MAUSEL pins to be converted to EADI pins. When EADISEL is set, the XMAUSEL bit in the same register must be zero.

When external address detection hardware is used, it may be convenient to set the disable receive broadcast (DRCVBC) bit in the mode register, thereby disabling the automatic reception of broadcast messages. The internal physical address detection logic can be turned off by setting the disable receive physical address (DRCVPA) bit in the

mode register. Also, the internal logical address filtering can be disabled by filling the logical address filter (LADRF) register with zeros.

5.3.4 Network Software Procedures

5.3.4.1 Initialization

Automatic Initialization
The PCnet-ISA can be initialized just as the LANCE by setting up an initialization block in memory, then setting the INIT bit in CSR0. A minor difference is that with the PCnet-ISA, the INIT and STRT bits can be set at the same time, whereas with the LANCE these bits should be set in two separate I/O operations.

Direct Initialization
As an alternative to the LANCE-type automatic initialization, the PCnet-ISA can be initialized directly by writing to the appropriate control and status registers. This alternate method simplifies the software somewhat. If this method is used, the INIT bit in CSR0 should not be set.

Descriptor Ring Lengths
The LANCE restricts the sizes of descriptor rings to integer powers of 2. The PCnet-ISA removes this restriction. The software can set the number of transmit or receive descriptors to any number between 1 and 65535 by loading the two's complement of the desired value into CSR76 for receive descriptors or CSR78 for transmit descriptors.

The automatic initialization procedure using the initialization block and the INIT bit in CSR0 can only set descriptor ring lengths to integer powers of two. However, the software can bypass this procedure altogether by writing directly to control and status registers, or it can use the automatic process first, then write directly to CSR76 and CSR78 to override the values set up by the automatic process.

5.3.4.2 Transmission and Reception

Start of Transmit Interrupt
The PCnet-ISA can interrupt the processor when it starts to transmit a packet. This interrupt occurs if the corresponding mask bit (TXSTRTM in CSR4) is cleared. This interrupt is masked out by default.

Disable Receive Broadcast
The reception of broadcast packets can be disabled by setting the DRCVBC bit in the mode register (CSR15). This reduces overhead for network protocols that do not use broadcast messages.

Disable Physical Address Detection
If the DRCVPA bit is set in the mode register (CSR15), the PCnet-ISA will not save packets addressed to the node's physical address.

5.3.4.3 Changing Modes
To change any bit in the initialization address register (IADR), LADRF, physical address register (PADR), Mode, receive ring length register (RCVRL), transmit ring length register (XMTRL), FIFO threshold register, or DMA bus activity timer register (DMATBAT), the software must first set the stop bit in CSR0. While it is not necessary to do a LANCE-type automatic initialization after the stop bit has been set, it is necessary to clean up the transmit and receive queues as described earlier in this chapter (Stopping the PCnet-ISA).

5.3.4.4 Loopback
The size of loopback packets are not limited to 32 bytes as with the LANCE. In fact, unless the runt packet accept mode is enabled, all loopback packets must be legal IEEE 802.3 or Ethernet packets of 64 bytes or more. This means that loopback tests written for

the LANCE should be modified either to increase the packet size or to enable runt packet accept.

The MENDECL bit in CSR15 determines whether internal loopback includes or excludes the MENDEC.

5.3.5 Network Performance Tuning

5.3.5.1 FIFO Watermark Selection

To match the PCnet-ISA with the characteristics of the computer's bus, the software can change the transmit and receive FIFO watermarks and the number of bytes that must be loaded into the transmit FIFO before the device starts to transmit. The FIFO watermarks are the points at which DMA transfers start.

For the receive FIFO, the watermark is the number of bytes that must be present in the FIFO to trigger a DMA burst. For the transmit FIFO, the watermark is measured in write cycles. The FIFO must have enough empty space to accommodate that number of write cycles before the DMA transfer starts.

The transmit start point is the number of bytes that must be loaded into the FIFO before the PCnet-ISA starts transmitting.

These three parameters are programmed via fields of CSR80, the FIFO threshold, and DMA burst control register. The watermarks can each have one of three possible values, while the transmit start point has four possibilities.

5.3.5.2 DMA Burst Length Control

The maximum time that the PCnet-ISA keeps control of the system bus during a DMA transfer burst can be controlled in two ways. The maximum number of DMA cycles the device uses before giving up the bus is programmable via the DMA burst counter in CSR80. The effect of this register can be disabled by setting the DMAPLUS bit in CSR4. When DMAPLUS is set, the DMA transfer continues until the FIFO is empty or full.

Alternatively, the maximum time that the device can hold the bus for a DMA burst is programmable via CSR82, the bus timer register. This timer is enabled or disabled via the timer bit in CSR4.

Both techniques can be enabled at the same time. In this case the PCnet-ISA releases the bus whenever either counter times out.

5.3.5.3 Disabling Descriptor Polling

The polling of transmit descriptors can be disabled by setting the disable polling (DPOLL) bit in CSR4. When this bit is set, the software must set the transmit demand (TDMD) bit in CSR0 each time it has a packet to send. Disabling the polling frees some bus bandwidth.

5.3.5.4 Disabling Transmission Two-part Deferral

The transmission two-part deferral algorithm can be disabled by setting the disable transmit two-part deferral (DXMT2PD) bit in CSR3.

5.3.5.5 Alternate Backoff Algorithm

An alternate backoff algorithm can be selected by setting the EMBA bit in CSR3.

5.4 Programmable Registers

Although the PCnet-ISA has more than 70 registers that can be accessed by software, most of these registers are intended for debugging and production testing purposes only. The following are the only registers that should be accessed by network software.

Control and Status Registers

Register	Contents
CSR0	Status and control bits (DEFAULT = 0004) 8000 ERR 0800 MERR 0080 INTR 0008 TDMD 4000 BABL 0400 RINT 0040 IENA 0004 STOP 2000 CERR 0200 TINT 0020 RXON 0002 STRT 1000 MISS 0100 IDON 0010 TXON 0001 INIT
CSR1	Lower IADR (maps to port 16)
CSR2	Upper IADR (maps to port 17)
CSR3	Interrupt masks and configuration bits (DEFAULT = 0) 8000 – 0800 MERM 0080 – 0008 EMBA 4000 BABLM 0400 RINTM 0040 – 0004 – 2000 – 0200 TINTM 0020 – 0002 – 1000 MISSM 0100 IDONM 0010 DXMT2PD 0001 –
CSR4	Interrupt masks; configuration and status bits (DEFAULT = 0115) 8000 ENTST 0800 APADXMT 0080 – 0008 TXSTRT 4000 DMAPLUS 0400 ASTRPRCV 0040 – 0004 TXSTRTM 2000 TIMER 0200 MPCO 0020 RCVCCO 0002 JAB 1000 DPOLL 0100 MPCOM 0010 RCVCCOM 0001 JABM
CSR8– CSR11	Logical address filter
CSR12– CSR14	Physical address register
CSR15	MODE: (DEFAULT = 0) 8000 PROM 0800 DAPC 0080 PORTSEL0 0008 DXMTFCS 4000 DRCVBC 0400 MENDECL 0040 INTL 0004 LOOP 2000 DRCVPA 0200 LRT/TSEL 0020 DRTY 0002 DTX 1000 DLNKTST 0100 PORTSEL1 0010 FCOLL 0001 DRX
CSR76	RCVRL, RCV descriptor ring length
CSR78	XMTRL, XMT descriptor ring length
CSR80	FIFO threshold and DMA burst control. (DEFAULT = 2810) bits [13:12] = RCVFW, receive FIFO watermark 0000 Request DMA when 16 bytes are present 1000 Request DMA when 32 bytes are present 2000 Request DMA when 64 bytes are present* bits [11:10] = XMTSP, transmit start point 0000 Start transmission after 4 bytes have been written 0400 Start transmission after 16 bytes have been written 0800 Start transmission after 64 bytes have been written* 0C00 Start transmission after 112 bytes have been written bits [9:8] = XMTFW, transmit FIFO watermark 0000 Start DMA when 8 write cycles can be made* 0100 Start DMA when 16 write cycles can be made 0200 Start DMA when 32 write cycles can be made bits [7:0] = DMA Burst Register

*default value

Control and Status Registers (Continued)

Register	Contents
CSR82	DMATR, bus timer
CSR88	Chip ID (contents = 00003003)
CSR112	Missed-packet count
CSR114	Receive collision count
CSR124	BMU test register: Used only for runt packet accept bit (0004)

ISA Bus Configuration Registers

RAP Addr	Register	Contents
0	MSRDA	Programs width of DMA read signal (DEFAULT = 5)
1	MSWRA	Programs width of DMA write signal (DEFAULT = 5)
2	MC	ISA bus configuration bits (DEFAULT = 0001) 0008 EADISEL 0004 AWAKE 0002 ASEL 0001 XMAUSEL
5	LED1	Programs the function and width of the LED1 signal (DEFAULT = 0084) 8000 LEDOUT 0008 RVPE 0080 PSE 0004 RCVE 0010 XMTE 0002 JABE 0001 COLE
6	LED2	Programs the function and width of the LED2 signal (DEFAULT = 0008) (Same format as LED1)
7	LED3	Programs the function and width of the LED3 signal (DEFAULT = 0090) (Same format as LED1)

CHAPTER 6
Jumperless Solution

6.1 Description

The following design implements a PCnet-ISA based PC/AT board (bus master mode) that is automatically software configured, requiring no hardware adjustments or settings. While this implementation specifically addresses an add-in board application, an identical solution is equally applicable to a motherboard design. This jumperless hardware configures the I/O address space, the direct memory access (DMA) channel, and the interrupt number used. The board configuration is stored on the PCnet-ISA board in an EEPROM device (93C46), enabling the board to remember its configuration even after loss of power. The stored configuration is installed automatically upon power-up, using the reset signal on the Industry Standard Architecture (ISA) bus. The configuration data can also be installed via software control. The configuration data can be written and read via software control. The design presented here is implemented using external hardware that intercepts I/O accesses of the ISA bus configuration register (ISACSR3); future versions of the PCnet-ISA could integrate most of this design while maintaining 100% software compatibility.

The PCnet-ISA board must implement the former jumpered functions, namely address decoding, interrupt selection, and DMA channel selection. In the jumperless design, the jumper positions are replaced with configuration bits stored in the EEPROM. Loaded into the hardware, these bits define the functionality of the address decoder and the interrupt and DMA multiplexers. In this design external hardware catches the last seven select bits as they are clocked out of the EEPROM. Future PCnet-ISA silicon could allocate additional select bits for additional functions. In this design, a total of 32 select bits are defined, although this external implementation uses only 16. Externally accessible for self-configuration, these 16 bits are stored in address locations 8 and 9 of the EEPROM and are defined in Figures 21 and 22 and the tables that follow.

16850A-021A
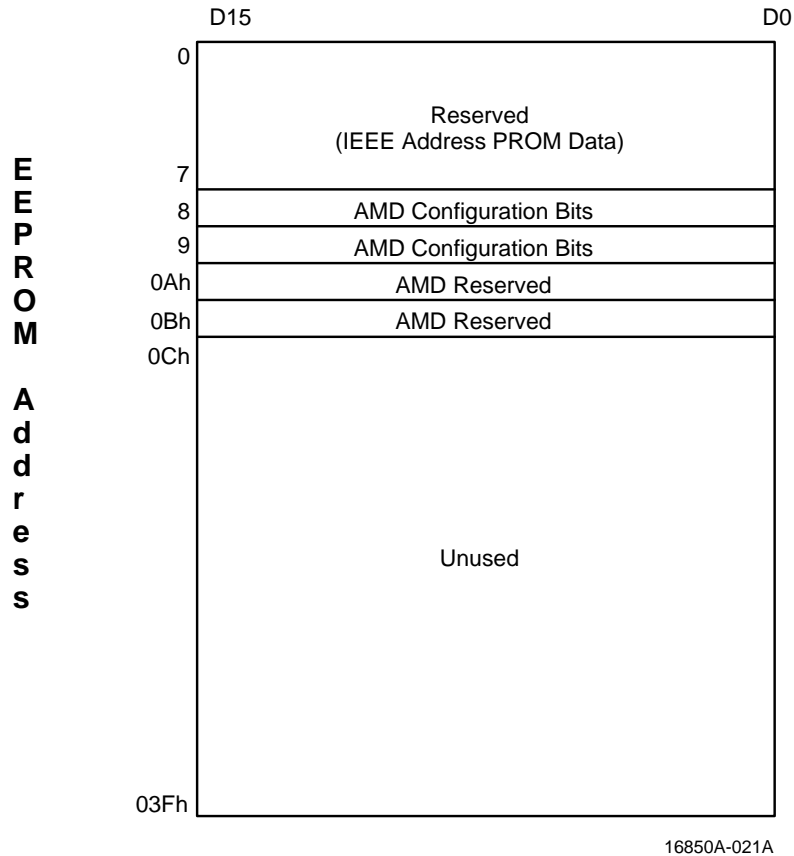
Figure 21. EEPROM Address Space Map

Location 9

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES	RES

Location 8

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RES	RES	RES	RES	SMAP4	SMAP3	SMAP2	SMAP1	SMAP0	BOOT	IOAM1	IOAM0	DMA SEL1	DMA SEL0	IRQ SEL1	IRQ SEL0

RES: Reserved for future use by AMD or user.

16850A-022A

Figure 22. AMD Configuration Bits

SMAP4-SMAP0: (Shared-memory mode only) shared RAM address map (8K blocks).

(Reserved bits in bus master mode)

SMAP [4:0]	Address	SMAP [4:0]	Address	SMAP [4:0]	Address	SMAP [4:0]	Address
00000	C0000-C1FFF	00000	D0000-D1FFF	00000	E0000-E1FFF	00000	F0000-F1FFF
00001	C2000-C3FFF	00001	D2000-D3FFF	00001	E2000-E3FFF	00001	F2000-F3FFF
00010	C4000-C5FFF	00010	D4000-D5FFF	00010	E4000-E5FFF	00010	F4000-F5FFF
00011	C6000-C7FFF	00011	D6000-D7FFF	00011	E6000-E7FFF	00011	F6000-F7FFF
00100	C8000-C9FFF	00100	D8000-D9FFF	00100	E8000-E9FFF	00100	F8000-F9FFF
00101	CA000-CBFFF	00101	DA000-DBFFF	00101	EA000-EBFFF	00101	FA000-FBFFF
00110	CC000-CDFFF	00110	DC000-DDFFF	00110	EC000-EDFFF	00110	FC000-FDFFF
00111	CE000-CFFFF	00111	DE000-DFFFF	00111	EE000-EFFFF	00111	FE000-FFFFF

BOOT: Boot PROM enable. When set, the boot PROM address space is defined by the I/O address map (IOAM[1:0]).

IOAM1-IOAM0: Define PCnet-ISA I/O base address and boot PROM address

IOAM[1:0]	I/O Base Address	Boot PROM (if enabled by BOOT bit)
00	300h	C8000-CBFFFh
01	320h	CC000-CFFFFh
10	340h	D0000-D3FFFh
11	360h	D4000-D7FFFh

DMASEL3-DMASEL2: Define DMA channel used

DMASEL[3:2]	DMA Channel
00	3
01	5
10	6
11	7

IRQSEL1-IRQSEL0: Define interrupt request used

IRQSEL[1:0]	Interrupt
00	3
01	4
10	5
11	9

EEPROM locations 0-7 (16 bytes) are reserved for future use to hold the Novell Ethernet address PROM data. This address PROM space in the EEPROM can be programmed during production-board test. This external design does not use this address PROM space and requires a separate address PROM, although future PCnet-ISA silicon may emulate the address PROM internally, loaded with data read from the EEPROM.

As previously mentioned, the PCnet-ISA board can be configured (loading the select bits from the EEPROM into the jumperless hardware) via software control by writing 52h to ISACSR3. The initialization hardware then automatically loads all select bits into the jumperless hardware.

As with the auto configuration command, direct EEPROM accesses are mapped through ISACSR3 on the PCnet-ISA, although the mechanism is slightly more complicated. This complicated sequence is intentional in an effort to virtually eliminate unintentional corruption of the EEPROM data. Because EEPROM writes must be executed only during initial board installation and reconfiguration, the speed of execution is of minor importance. The EEPROM used on the PCnet-ISA board is a serial device, and the user read and write hardware used here directly accesses the control pins on the EEPROM. Therefore, a series of vectors must be written to ISACSR3 to access the EEPROM. The EEPROM control lines are chip select (CS), serial clock (SK), data in (DI), and data out (DO). Please refer to the 93C46 EEPROM (National Semiconductors, Samsung, Hyundai) datasheet for details. Basically, an opcode and address is clocked into the DO pin of the EEPROM, and the associated data are then either clocked into DI (write) or clocked out of the DO pin (read). In this design, the CS line is set and latched with a write of 69h to ISACSR3 and reset with a write of 68h to ISACSR3. Once the CS line is set, the states of the SK and DI lines can be controlled through writes to ISACSR3, and the state of DO can be accessed through any read of ISACSR3.

The 8-bit data pattern written to ISACSR3 for the EEPROM control writes is defined in Figure 23.

MSB   LSB

1	0	0	1	0	1	SK	DI

The state of the DO pin of the EEPROM is reflected in the least significant bit of any read of ISACSR3 of the PCnet-ISA (provided the EEPROM CS line has been set).

MSB   LSB

X	X	X	X	X	X	X	DO
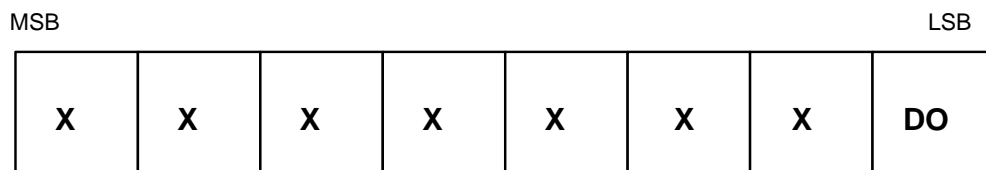
16850A-023A

Figure 23. The 8-Bit Pattern for EEPROM Control Writes

Any pattern written to ISACSR3 other than 52h (software configuration), 69h (CS enable), 68h (CS disable), or the above EEPROM control pattern (100101xx) has no affect on the operation of the jumperless board. With the exception of the EEPROM control register (ISACSR3), the PCnet-ISA and boot PROM does not respond to any ISA bus commands while the EEPROM CS is enabled by the user software. No resource (PCnet-ISA, boot PROM, ISACSR3) will respond while the jumperless hardware is executing the configuration cycle.

A typical software EEPROM access would consist of operations described below.

- Set the CS line on the EEPROM by writing 69h to ISACSR3.
- Clock EEPROM opcode and address into the EEPROM by writing a series of patterns (100101xx) to ISACSR3.

- (EEPROM write) Clock data word into EEPROM by writing a series of patterns (100101xx) to ISACSR3.

- (EEPROM read) Clock data word out of EEPROM one bit (D) at a time by performing alternate writes (100101xx) and reads (xxxxxxxD) of ISACSR3.

- Reset the CS line on the EEPROM by writing 68h to ISACSR3.

These operations must be done sequentially as the EEPROM command cycle is reset once the EEPROM CS is deasserted.

The initialization of the select bits in the EEPROM normally occurs when the PCnet-ISA board is installed in a workstation. The user configures the board with a software utility that proceeds automatically or allows the user to manually select the hardware settings.

A typical configuration of the jumperless board consists of the following operations:

- Locate the board at its current location (one of four possible address spaces).

- Write new bit values for the address space, DMA channel, and interrupt number directly to the appropriate locations in the EEPROM (see programming examples later in this Chapter).

- Issue the reconfigure command (write 52h to ISACSR3). This loads the new configuration bits from the EEPROM into the configuration hardware. All subsequent operation of the jumperless board are determined by the newly loaded configuration. Reconfiguration is now complete.

6.2 Logic Equations

Bus Master Mode

The following connections and equations are written with no implied logic device or partitioning information.

Input Connections

```
AEN        FROM THE ISA BUS
DACK3      FROM THE ISA BUS
DACK5      FROM THE ISA BUS
DACK6      FROM THE ISA BUS
DACK7      FROM THE ISA BUS
DRQ        FROM THE PCnet-ISA DRQ PIN
EEPROM_DO  FROM THE 93C46 EEPROM DO PIN
IOR        FROM THE ISA BUS
IOW        FROM THE ISA BUS
IRQ        FROM THE PCnet-ISA IRQ PIN
SA1-SA9    FROM THE ISA BUS
SD0-SD7    FROM THE ISA BUS
SMEMR      FROM THE ISA BUS
```

Output Connections

```
DACK            TO THE PCnet-ISA DACK PIN
DRQ3            TO THE ISA BUS
DRQ5            TO THE ISA BUS
DRQ6            TO THE ISA BUS
DRQ7            TO THE ISA BUS
EE_CS1          TO THE 93C46 EEPROM CS PIN
EE_CS2          TO THE 93C46 EEPROM CS PIN
EE_DI1          TO THE 93C46 EEPROM DI PIN
EE_DI2          TO THE 93C46 EEPROM DI PIN
EE_SK1          TO THE 93C46 EEPROM SK PIN
```

```
IEE_SK2            TO THE 93C46 EEPROM SK PIN

IOAM_0             TO THE PCnet-ISA IOAM0 PIN

IOAM_1             TO THE PCnet-ISA IOAM1 PIN

IOR_PCNET_ISA      TO THE PCnet-ISA IOR PIN

IOW_PCNET_ISA      TO THE PCnet-ISA IOW PIN

IRQ3               TO THE ISA BUS

IRQ4               TO THE ISA BUS

IRQ5               TO THE ISA BUS

IRQ9               TO THE ISA BUS

SMEMR_PCNET_ISA    TO THE PCnet-ISA SMEMR PIN
```

Bi-directional Connections

```
SD0        TO THE ISA BUS
```

Internal Terms

```
BOOT_EN    LATCHED VALUE OF EEPROM DATA

BUSY       ASSERTED DURING LOGIC (RE)CONFIGURATION CYCLE

CONFIGCLK  CLOCK THAT SHIFTS CONFIGURATION DATA INTO
           JUMPERLESS LOGIC

DMASEL_0   LATCHED VALUE OF EEPROM DATA

DMASEL_1   LATCHED VALUE OF EEPROM DATA

EEQ1       LOGIC CONFIGURATION SEQUENCER STATE VARIABLE

EEQ2       LOGIC CONFIGURATION SEQUENCER STATE VARIABLE

EEQ3       LOGIC CONFIGURATION SEQUENCER STATE VARIABLE

EEQ4       LOGIC CONFIGURATION SEQUENCER STATE VARIABLE

EEQ5       LOGIC CONFIGURATION SEQUENCER STATE VARIABLE

EERAPADD   SET IF RAP WAS LAST WRITTEN WITH 0003
           (POTENTIAL POINTER TO ISACSR3)

IRQSEL_0   LATCHED VALUE OF EEPROM DATA

IRQSEL_1   LATCHED VALUE OF EEPROM DATA

ISACSRACC  ASSERTED WHEN THE ISACSR IS BEING ACCESSED
           (ADDRESS DECODE)

RAPACC     ASSERTED WHEN RAP IS BEING ACCESSED (ADDRESS
           DECODE)

SWCONFIG   ASSERTED WHEN THE (SOFTWARE) CONFIG
           COMMAND DETECTED
```

The following terms are combinatorial.

```
DACK     = /DMASEL_1 * /DMASEL_0 * DACK3

       + /DMASEL_1 * DMASEL_0 * DACK5

       + DMASEL_1 * /DMASEL_0 * DACK6

       + DMASEL_1 * DMASEL_0 * DACK7

DRQ3.TRST = /DMASEL_1 * /DMASEL_0 * /BUSY

DRQ3     = /DMASEL_1 * /DMASEL_0 * DRQ

DRQ5.TRST = /DMASEL_1 * DMASEL_0 * /BUSY

DRQ5     = /DMASEL_1 * DMASEL_0 * DRQ
```

```
DRQ6.TRST = DMASEL_1 * /DMASEL_0 * /BUSY
DRQ6     = DMASEL_1 * /DMASEL_0 * DRQ
DRQ7.TRST = DMASEL_1 * DMASEL_0 * /BUSY
DRQ7     = DMASEL_1 * DMASEL_0 * DRQ
IOR_PCNET_ISA = IOR * /EE_CS1 * /BUSY * (/EERAPADD + /ISACSRACC)
IOW_PCNET_ISA = IOW * /EE_CS1 * /BUSY * (/EERAPADD + /ISACSRACC)
IRQ3.TRST = /IRQSEL_1 * /IRQSEL_0 * /IRQ * /BUSY
IRQ3     = /IRQSEL_1 * /IRQSEL_0 * IRQ
IRQ4.TRST = /IRQSEL_1 * IRQSEL_0 * /IRQ * /BUSY
IRQ4     = /IRQSEL_1 * IRQSEL_0 * IRQ
IRQ5.TRST = IRQSEL_1 * /IRQSEL_0 * /IRQ * /BUSY
IRQ5     = IRQSEL_1 * /IRQSEL_0 * IRQ
IRQ9.TRST = IRQSEL_1 * IRQSEL_0 * /IRQ * /BUSY
IRQ9     = IRQSEL_1 * IRQSEL_0 * IRQ
ISACSRACC = /IOAM_1 * /IOAM_0 * SA9 * SA8 * /SA7 * /SA6 * /SA5 * SA4 *
/SA3 * SA2 * SA1 * /SA0 * /AEN
    + /IOAM_1 * IOAM_0 * SA9 * SA8 * /SA7 * /SA6 * SA5 * SA4 * /SA3 *
SA2 * SA1 * /SA0 * /AEN
    + IOAM_1 * /IOAM_0 * SA9 * SA8 * /SA7 * SA6 * /SA5 * SA4 * /SA3 *
SA2 * SA1 * /SA0 * /AEN
    + IOAM_1 * IOAM_0 * SA9 * SA8 * /SA7 * SA6 * SA5 * SA4 * /SA3 * SA2
* SA1 * /SA0 * /AEN
RAPACC  = /IOAM_1 * /IOAM_0 * SA9 * SA8 * /SA7 * /SA6 * /SA5 * SA4 *
/SA3 * /SA2 * SA1 * /SA0 * /AEN
    + /IOAM_1 * IOAM_0 * SA9 * SA8 * /SA7 * /SA6 * SA5 * SA4 * /SA3 *
/SA2 * SA1 * /SA0 * /AEN
    + IOAM_1 * /IOAM_0 * SA9 * SA8 * /SA7 * SA6 * /SA5 * SA4 * /SA3 *
/SA2 * SA1 * /SA0 * /AEN
    + IOAM_1 * IOAM_0 * SA9 * SA8 * /SA7 * SA6 * SA5 * SA4 * /SA3 * /SA2
* SA1 * /SA0 * /AEN
SD0.TRST = IOR * EE_CS1 * /BUSY * EERAPADD * ISACSRACC
SD0      = EEPROM_DO
SMEMR_PCNET_ISA    = BOOT_EN * SMEMR * /BUSY
```

The following terms are clocked by CONFIGCLK.

```
IRQSEL_0  := EEPROM_DO
IRQSEL_1  := IRQSEL_0
DMASEL_0  := IRQSEL_1
DMASEL_1  := DMASEL_0
IOAM_0    := DMASEL_1
IOAM_1    := IOAM_0
BOOT_EN   := IOAM_1
```

The following terms are clocked by $\overline{\text{IOW}}$ on the ISA bus.

```
EE_CS1.TRST = /BUSY
EE_CS1   := EERAPADD * /BUSY * ISACSRACC * /SD7 * SD6 * SD5 * /SD4 * SD3
            * /SD2 * /SD1 * SD0 + EE_CS1 * /BUSY * (/EERAPADD + SD7 + /SD6
            + /SD5 + SD4 + /SD3 + SD2 + SD1)
EE_DI1.TRST = /BUSY
EE_DI1   := SD0 * /BUSY * EERAPADD * ISACSRACC * EE_CS1 * SD7 * /SD6
            * /SD5 * SD4 * /SD3 * SD2 + EE_DI1 * /BUSY * (/EERAPADD
            + /ISACSRACC + /EE_CS1 + /SD7 + SD6 + SD5 + /SD4 + SD3 + /SD2)
EERAPADD := RAPACC * /SD7 * /SD6 * /SD5 * /SD4 * /SD3 * /SD2 * SD1
            * SD0 + EERAPADD * (/RAPACC + SD7 + SD6 + SD5 + SD4 + SD3
            + SD2 + /SD1 + /SD0)
EE_SK1.TRST = /BUSY
EE_SK1   := SD1 * /BUSY * EERAPADD * ISACSRACC * EE_CS1 * SD7 * /SD6
            * /SD5 * SD4 * /SD3 * SD2 + EE_SK1 * /BUSY * (/EERAPADD
            + /ISACSRACC + /EE_CS1 + /SD7 + SD6 + SD5 + /SD4 + SD3 + /SD2)
SWCONFIG := EERAPADD * ISACSRACC * /EE_CS1 * /BUSY * /SD7 * SD6 * /SD5
            * SD4 * /SD3 * /SD2 * SD1 * /SD0
SWCONFIG.RSTF = BUSY
```

The following terms are clocked by a 500 kHz (max frequency) clock, which may be derived from the oscillator (OSC) signal of the ISA bus (÷32).

```
BUSY       := /EEQ5 * /EEQ4 * /EEQ3 * /EEQ2 * /EEQ1 * /EE_SK2 * (RESET
    + SWCONFIG) + BUSY * (/EEQ5 + /EEQ4 + EEQ3 + /EEQ2 + /EEQ1 + EE_SK2)
CONFIGCLK   :=
     /EEQ5 * EEQ4 * /EEQ3 * EEQ2 * /EEQ1 * EE_SK2
   + /EEQ5 * EEQ4 * /EEQ3 * EEQ2 * EEQ1 * EE_SK2
   + /EEQ5 * EEQ4 * EEQ3 * /EEQ2 * /EEQ1 * EE_SK2
   + /EEQ5 * EEQ4 * EEQ3 * /EEQ2 * EEQ1 * EE_SK2
   + /EEQ5 * EEQ4 * EEQ3 * EEQ2 * /EEQ1 * EE_SK2
   + /EEQ5 * EEQ4 * EEQ3 * EEQ2 * EEQ1 * EE_SK2
   + EEQ5 * /EEQ4 * /EEQ3 * /EEQ2 * /EEQ1 * EE_SK2
   + EEQ5 * /EEQ4 * /EEQ3 * /EEQ2 * EEQ1 * EE_SK2
   + EEQ5 * /EEQ4 * /EEQ3 * EEQ2 * /EEQ1 * EE_SK2
   + EEQ5 * /EEQ4 * /EEQ3 * EEQ2 * EEQ1 * EE_SK2
   + EEQ5 * /EEQ4 * EEQ3 * /EEQ2 * /EEQ1 * EE_SK2
   + EEQ5 * /EEQ4 * EEQ3 * /EEQ2 * EEQ1 * EE_SK2
   + EEQ5 * /EEQ4 * EEQ3 * EEQ2 * /EEQ1 * EE_SK2
   + EEQ5 * /EEQ4 * EEQ3 * EEQ2 * EEQ1 * EE_SK2
   + EEQ5 * EEQ4 * /EEQ3 * /EEQ2 * /EEQ1 * EE_SK2
   + EEQ5 * EEQ4 * /EEQ3 * /EEQ2 * EEQ1 * EE_SK2
EE_CS2.TRST = BUSY
EE_CS2 := /EEQ5 * /EEQ4 * /EEQ3 * /EEQ2 * /EEQ1 * EE_SK2 * BUSY
    + EE_CS2 * (/EEQ5 + /EEQ4 + EEQ3 + /EEQ2 + EEQ1 + /EE_SK2) * BUSY
EE_DI2.TRST = BUSY
EE_DI2     := /EEQ5 * /EEQ4 * /EEQ3 * /EEQ2 * /EEQ1 * EE_SK2
    + /EEQ5 * /EEQ4 * /EEQ3 * /EEQ2 * EEQ1
    + /EEQ5 * /EEQ4 * /EEQ3 * EEQ2 * /EEQ1 * /EE_SK2
    + /EEQ5 * /EEQ4 * EEQ3 * /EEQ2 * EEQ1 * EE_SK2
    + /EEQ5 * /EEQ4 * EEQ3 * EEQ2 * /EEQ1 * /EE_SK2
```

```
EE_SK2.TRST = BUSY
EE_SK2 := /EE_SK2 * BUSY * /RESET * (/EEQ5 + /EEQ4 + EEQ3 + /EEQ2 + /Q1)


EEQ1    := /EEQ1 * EE_SK2 * BUSY * /RESET
         + EEQ1 * /EE_SK2 * BUSY * /RESET


EEQ2    := /EEQ2 * EEQ1 * EE_SK2 * BUSY * /RESET
         + EEQ2 * (/EEQ1 + /EE_SK2) * BUSY * /RESET


EEQ3    := /EEQ3 * EEQ2 * EEQ1 * EE_SK2 * BUSY * /RESET
         + EEQ3 * (/EEQ2 + /EEQ1 + /EE_SK2) * BUSY * /RESET


EEQ4    := /EEQ4 * EEQ3 * EEQ2 * EEQ1 * EE_SK2 * BUSY * /RESET
         + EEQ4 * (/EEQ3 + /EEQ2 + /EEQ1 + /EE_SK2) * BUSY * /RESET


EEQ5    := /EEQ5 * EEQ4 * EEQ3 * EEQ2 * EEQ1 * EE_SK2 * BUSY * /RESET
         + EEQ5 * (/EEQ4 + /EEQ3 + /EEQ2 + /EEQ1 + /EE_SK2) * BUSY
           * /RESET
```

6.3 Code Examples

```
CHIP_SELECT_ENABLE      EQU     0069H
CHIP_SELECT_DISABLE     EQU     0068H
CONFIGURE_COMMAND       EQU     0052H
EE_ACCESS_PATTERN       EQU     0094H   ;THE TWO LSBITS CORRESPOND TO
                                        ;EEPROM CONTROL PINS SK & DI
RAP_OFFSET   EQU    12H;OFFSET OF REGISTER ADDRESS PORT FROM I/O BASE
RDP_OFFSET   EQU    10H;OFFSET OF REGISTER DATA PORT FROM I/O BASE
IDP_OFFSET   EQU    16H;OFFSET OF ISA CONFIGURATION REGISTER DATA PORT
                      ;(IDP) FROM I/O BASE
;    THIS ROUTINE READS A 16-BIT QUANTITY FROM THE 93C46 EEPROM.         ;
;    CALL THIS ROUTINE WITH THE EEPROM ADDRESS STORED IN BL.             ;
;    VALID VALUES IN BL ARE 0 TO 63; THE TWO MSBITS ARE IGNORED.         ;
;    THE 16-BIT DATA IS RETURNED IN BX                                   ;
;    THE LEAST-SIGNIFICANT BIT IN AX IS THE LAST BIT CLOCKED OUT         ;
;    OF THE EEPROM                                                       ;
```

```
            EEPROM_READ  PROC  NEAR
        PUSH     AX
        PUSH     CX
        PUSH     DX
        MOV      DX,IO_BASE_ADDRESS     ;DEFINED BY IOAM1,0 PINS
        ADD      DX,RAP_OFFSET
        MOV      AX,3                   ;ISACSR3 POINTER
        OUT      DX,AX                  ;RAP = 3
        MOV      DX,IO_BASE_ADDRESS
        ADD      DX,IDP_OFFSET          ;POINTS TO IDP
        MOV      AX,CHIP_SELECT_ENABLE
        OUT      DX,AX                  ;ENABLE EEPROM CHIP SELECT (CS PIN)
        STC                             ;BIT TO BE CLOCKED IS A 1
        CALL     BIT_CLOCK              ;START BIT
        OR       BL,080H                ;2 MSBITS OF BL= OPCODE=10 FOR READ
        CALL     BYTE_CLOCK             ;CLOCKS OPCODE & ADDRESS INTO
                                        ;EEPROM
        MOV      CL,16                  ;16 BITS TO READ OUT OF EEPROM
    READ_BITS:
        CLC
        CALL     BIT_CLOCK              ;CLOCK WITH DI=0
        IN       AX,DX                  ;DO IS LSBIT OF THIS I/O READ
        ROR      AX,1                   ;SHIFT INTO CARRY FLAG
        RCL      BX,1                   ;SHIFT CARRY FLAG INTO LSBIT OF BX
        LOOP     READ_BITS
        CLC
        CALL     BIT_CLOCK              ;LAST EEPROM SK EDGE
        MOV      AX,CHIP_SELECT_DISABLE
        OUT      DX,AX                  ;DISABLE EEPROM CHIP SELECT
        CALL     DELAY1                 ;1 MICROSECOND DELAY ROUTINE
        POP      DX
        POP      CX
        POP      AX
        RET
    EEPROM_READ  ENDP
    ;   THIS ROUTINE WRITES THE 16-BIT QUANTITY IN AX TO THE 93C46 EEPROM. ;
    ;   CALL THIS ROUTINE WITH THE EEPROM ADDRESS STORED IN BL (0-63).     ;
    ;   THE LEAST-SIGNIFICANT BIT IN AX IS THE LAST BIT CLOCKED INTO THE EEPROM ;

    EEPROM_WRITE PROC  NEAR
        PUSH     AX
        PUSH     BX
        PUSH     CX
        PUSH     DX
        MOV      CX,AX                  ;SAVE DATA IN CX
        MOV      DX,IO_BASE_ADDRESS
        ADD      DX,RAP_OFFSET
        MOV      AX,3                   ;ISACSR3 POINTER
        OUT      DX,AX                  ;RAP = 3
```

```
        MOV     DX,IO_BASE_ADDRESS
        ADD     DX,IDP_OFFSET           ;POINTS TO IDP
        MOV     AX,CHIP_SELECT_ENABLE
        OUT     DX,AX                   ;ENABLE EEPROM CHIP SELECT (CS PIN)
        STC                             ;BIT TO BE CLOCKED IS A 1
        CALL    BIT_CLOCK               ;START BIT
        MOV     BH,BL                   ;SAVE ADDRESS IN BH
        MOV     BL,30H                  ;4 MSBITS OF BL= OPCODE=0011 FOR
                                        ;EEPROM WRITE ENABLE
        CALL    BYTE_CLOCK              ;CLOCKS WRITE ENABLE OPCODE INTO
                                        ;EEPROM
        MOV     AX,CHIP_SELECT_DISABLE
        OUT     DX,AX                   ;DISABLE EEPROM CS PIN
        CALL    DELAY1                  ;1 MICROSECOND DELAY
        MOV     AX,CHIP_SELECT_ENABLE
        OUT     DX,AX
        STC
        CALL    BIT_CLOCK               ;START BIT
        MOV     BL,BH                   ;RETRIEVE EEPROM ADDRESS
        AND     BL,3FH
        OR      BL,40H                  ;2 MSBITS OF BL= OPCODE=01 FOR WRITE
        CALL    BYTE_CLOCK              ;CLOCKS OPCODE AND ADDRESS INTO
                                        ;EEPROM
        MOV     BX,CX                   ;PUT WRITE DATA INTO BX
        MOV     CL,16                   ;16 BITS TO READ OUT OF EEPROM
WRITE_BITS:
        ROL     BX,1                    ;COPY OF MSBIT TO CARRY FLAG
        CALL    BIT_CLOCK               ;CLOCK WITH DI=DATA BIT
        LOOP    WRITE_BITS              ;CLOCK ALL DATA BITS INTO EEPROM
        MOV     AX,CHIP_SELECT_DISABLE
        OUT     DX,AX                   ;DISABLE EEPROM CHIP SELECT
        CALL    DELAY1                  ;1 MICROSECOND DELAY
        MOV     AX,CHIP_SELECT_ENABLE
        OUT     DX,AX                   ;ENABLE EEPROM CHIP SELECT
BUSY_CHECK:
        CLC
        CALL    BIT_CLOCK               ;CLOCK EEPROM WITH DI=0
        IN      AX,DX                   ;CHECK BUSY STATUS OF EEPROM
        AND     AL,1                    ; BY LOOKING AT THE EEPROM DO PIN
        JZ      BUSY_CHECK              ;KEEP LOOKING UNTIL DO <> 0
        MOV     AX,CHIP_SELECT_DISABLE
        OUT     DX,AX                   ;DISABLE EEPROM CHIP SELECT
        CALL    DELAY1                  ;1 MICROSECOND DELAY
        MOV     AX,CHIP_SELECT_ENABLE
        OUT     DX,AX                   ;ENABLE EEPROM CHIP SELECT
        STC
        CALL    BIT_CLOCK               ;START BIT
        XOR     BL,BL                   ;4 MSBITS OF BL= OPCODE= 0000 FOR
                                        ;EEPROM WRITE DISABLE
```

```
            CALL        BYTE_CLOCK              ;CLOCKS WRITE DISABLE OPCODE INTO EEPROM
            MOV         AX,CHIP_SELECT_DISABLE
            OUT         DX,AX                   ;DISABLE EEPROM CHIP SELECT
            CALL        DELAY1                  ;1 MICROSECOND DELAY
            POP         DX
            POP         CX
            POP         BX
            POP         AX
            RET
EEPROM_WRITE ENDP
;    THIS ROUTINE CLOCKS THE VALUE OF THE CARRY BIT INTO THE 93C46     ;
;    EEPROM. WHEN THE CARRY BIT IS RESET, THIS ROUTINE FUNCTIONS       ;
;    AS THE EEPROM DATA OUT CLOCK                                      ;
BIT_CLOCK PROC   NEAR
            PUSH        AX
            PUSH        DX
            PUSHF                               ;SAVE STATE OF THE CARRY BIT
            MOV        DX,IO_BASE_ADDRESS
            ADD        DX,IDP_OFFSET
            MOV        AX,EE_ACCESS_PATTERN
            POPF                                ;RETREIVE STATE OF THE CARRY BIT
            JNC        FALLING_EDGE
            OR         AX,1                     ;SET THE DI LINE IF CARRY SET.
FALLING_EDGE:
            OUT        DX,AX                    ;FALLING EDGE OF SK
                                                ;DI = DATA BIT
            CALL       DELAY2                   ;2 MICROSECOND DELAY ROUTINE
            OR         AX,2
            OUT        DX,AX                    ;RISING EDGE OF SK
                                                ;DI = DATA BIT
            CALL       DELAY2                   ;2 MICROSECOND DELAY ROUTINE
            POP        DX
            POP        AX
            RET
BIT_CLOCK ENDP
;    THIS ROUTINE CLOCKS THE BITS IN BL INTO THE 93C46 EEPROM    ;
;    THE BITS ARE CLOCKED MSBIT FIRST, LSBIT LAST                ;
BYTE_CLOCK    PROC    NEAR
            PUSH        BX
            PUSH        CX
            MOV CL,8                ;8 BITS TO CLOCK INTO EEPROM
SHIFT_BIT:
            ROL                     ;COPY OF MSBIT INTO CARRY FLAG
            CALL       EEPROM_CLOCK
            LOOP       SHIFT_BIT
            POP        CX
            POP        BX
            RET
BYTE_CLOCK    ENDP
```

Glossary

AUI
Attachment Unit Interface. IEEE specification for a node- or repeater-connection interface to an external medium attachment unit (MAU). The AUI cable between the DTE/repeater and the MAU may be up to 50 m in length. In systems where the MAU is embedded into the DTE or repeater (such as 10BASE-T or 10BASE2) a physical implementation of the AUI may not be present. Defined in Section 7 of ISO/IEC 8802-3: 1990 (ANSI/IEEE Std 802.3).

CI
Control In. AUI differential pair circuit, operating at pseudo-ECL levels. The MAU drives a 10 MHz signal on the CI circuit to indicate to the DTE or repeater that a collision has been detected on the network and/or an SQE Test from the MAU to the DTE is in progress.

Concentrator
A general term frequently used for repeater. Typically, a concentrator supports more than one network protocol, such as 802.3/Ethernet as well as 802.5/Token Ring. The terms hub, concentrator, and intelligent hub are frequently used interchangeably to reference a multiport, multiprotocol device, capable of statistics gathering, fault monitoring, and network management activities.

CRC
Cyclic Redundancy Check.

CSMA/CD
Carrier Sense Multiple Access/Collision Detect.

DI
Data In. AUI differential pair circuit, operating at pseudo-ECL levels. Data received by the MAU from either the media or the DO circuit is driven onto the DI circuit for use by the DTE or repeater.

DO
Data Out. AUI differential pair circuit, operating at pseudo-ECL levels. The DTE or repeater drives Manchester-encoded data out on the DO circuit; the data is transmitted by the MAU over the physical media and the DI circuit.

DTE
Data Terminal Equipment. Communication station (or node) capable of reception and transmission of data. Generally, it includes the MAC and PLS sublayer functions, but may also include an embedded MAU.

ENDEC
Encoder/Decoder.

FOIRL
Fiber Optic Inter Repeater Link. IEEE specification for inter repeater communications for repeaters. Defined in Section 9.9 of ISO/IEC 8802-3:1990 (ANSI/IEEE Std 802.3).

FCS
Frame Check Sequence.

Hub
A general term frequently used instead for repeater. See "concentrator."

IMR
Integrated Multiport Repeater (Am79C980). A single-chip repeater that incorporates eight 10BASE-T and one AUI-compatible ports, all 802.3 repeater requirements, management, diagnostics, and port expansion facilities.

IPG
Inter Packet Gap. The minimum time permitted between back-to-back packets on the 802.3 network. The gap time is specified as 96 bits (9.6 µs) minimum. A phenomenon known as IPG shrinkage can cause the IPG to be reduced below 96 bits.

LANCE
Local Area Network Controller for Ethernet (Am7990).

MAC
Media Access Control.

Manchester Encoding
Method for transmitting binary data on the medium.

MAU
Medium Attachment Unit. The physical and electrical interface between a DTE or repeater and the actual medium. The MAU is connected to the DTE by an attachment unit interface (see *AUI*), although this may not be visible if the MAU is embedded within the DTE or repeater. A different MAU is required to support each different type of medium (cable type).

PLS
Physical Layer Signalling.

PMA
Physical Medium Attachment.

Repeater
An 802.3/Ethernet repeater in its most generic form is an n-port device that supports 802.3 protocol only. A repeater is used to extend the physical topology of the network, allowing two or more cable segments to be coupled together. No more than four repeaters are permitted between the path of any two stations. When data are received on a single port, the repeater retransmits the incoming bit stream to all other ports, performing signal retiming and amplitude restoration. When data appears simultaneously on more than one port, the repeater transmits a collision to all ports, including the receiving ports. In addition, the repeater can isolate a port if it detects faults, such as excessive number or duration of collisions, to prevent disruption of the rest of the network. In a 10BASE-T network, the repeater provides a central point of connectivity, ideally suited to the incorporation of statistics gathering and network administration functions. Covered by Section 9 of ISO/IEC 8802-3 :1990 (ANSI/IEEE Std 802.3).

SIA
Serial Interface Adaptor (Am7992). A Manchester encoder/decoder chip that performs the physical layer signalling (PLS) sublayer functions of the IEEE 802.3 standard. The device encodes data and clock from the MAC for transmission over the network and drives the DO circuit of the AUI. It receives data from the network via the DI circuit of the AUI, extracts the data and clock into separate paths, and passes these back to the MAC.

SQE
Signal Quality Error. A 10 Mbps pulse train passed from the MAU (using the CI circuit) to a DTE or repeater to indicate an error condition on the network, such as collision or excessive transmit duration (jabber).

SQE Test
Signal Quality Error Test.

TPEX
Twisted-pair Ethernet transceiver (Am79C98 or Am79C100). A transceiver chip that converts the electrical signals of the AUI to those of the 10BASE-T standard.

10BASE-T

10 Mbps Baseband Twisted-Pair. Covered by Section 14 of IEEE 802.3. Uses 802.3 protocol, point-to-point twisted-pair cabling and repeaters to provide network services. No defined maximum node count; maximum cable distance is 100m. Defined in Section 13 and 14 of IEEE Std 802.3i-1990 (Supplement to ISO/IEC 8802-3: 1990 (ANSI/IEEE Std 802.3)).

10BASE-FL

10 Mbps Baseband Fiber Optic Link. Covered by draft of IEEE 802.3. Uses 802.3 protocol, dual fiber point-to-point cabling and repeaters to provide the network architecture. No defined maximum node count; maximum fiber distance is 1–2 km, depending on system configuration.

10BASE2

10 Mbps Baseband 200 m (frequently referred to as Cheapernet). A low-cost version of 10BASE5. It eliminates the external AUI requirement, relaxes the network electrical interfaces and allows use of thin, 75 Ω coaxial cable. Maximum 30 nodes (or mating connectors) on cable segment, 185 m per segment. Defined in Section 10 of ISO/IEC 8802-3: 1990 (ANSI/IEEE Std 802.3).

10BASE5

10 Mbps Baseband 500 m (commonly known as Ethernet). Based on the original Ethernet specification proposed by DEC, Intel, and Xerox, for multidrop communication scheme using the CSMA/CD access protocol, over thick, 75 Ω coaxial cable. 802.3 is the corresponding IEEE standard which varies in minor electrical and protocol specifications. Maximum 100 nodes on cable segment. Defined in Section 8 of ISO/IEC 8802-3: 1990 (ANSI/IEEE Std 802.3).

Appendix
PCnet-ISA Compatible Media
Interface Modules

PCnet-ISA Compatible 10BASE-T Media Interface Modules

Manufacturer	Part #	Package	Description
Bel Fuse	A556-2006-DE	16-pin 0.3" DIL	Transmit and receive filters and transformers
Bel Fuse	0556-2006-00	14-pin SIP	Transmit and receive filters and transformers
Bel Fuse	0556-2006-01	14-pin SIP	Transmit and receive filters, transformers, and common mode chokes
Bel Fuse	0556-6392-00	16-pin 0.5" DIL	Transmit and receive filters, transformers, and common mode chokes
Halo Electronics	FD02-101G	16-pin 0.3" DIL	Transmit and receive filters and transformers
Halo Electronics	FD12-101G	16-pin 0.3" DIL	Transmit and receive filters and transformers; transmit common mode choke
Halo Electronics	FD22-101G	16-pin 0.3" DIL	Transmit and receive filters, transformers, and common mode chokes
Nano pulse	NP6612	16-pin 0.3" DIL	Transmit and receive filters, transformers, and common mode chokes
Nano pulse	NP6581	8-pin 0.3" DIL	Transmit and receive common mode chokes
Nano pulse	NP6696	24-pin 0.6" DIL	Transmit and receive filters, transformers, and common mode chokes
PCA Electronics	EPA1990	16-pin 0.3" DIL	Transmit and receive filters and transformers
PCA Electronics	EPA2013	16-pin 0.3" DIL	Transmit and receive filters and transformers; transmit common mode choke
Pulse Engineering	PE65421	16-pin 0.3" DIL	Transmit and receive filters and transformers
Pulse Engineering	SUPRA 1.1	16-pin 0.5" DIL	Transmit and receive filters and transformers; transmit common mode choke
TDK	TLA 470	14-pin SIP	Transmit and receive filters and transformers
TDK	HIM3000	24-pin 0.6" DIL	Transmit and receive filters, transformers, and common mode chokes
Valor Electronics	PT3877	16-pin 0.3" DIL	Transmit and receive filters and transformers
Valor Electronics	PT3983	8-pin 0.3" DIL	Transmit and receive common mode chokes
Valor Electronics	FL1012	16-pin 0.3" DIL	Transmit and receive filters and transformers; transmit common mode choke

PCnet-ISA Compatible AUI Isolation Transformers

Manufacturer	Part #	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3" DIL	50 µH
PCA Electronics	EP9531-4	16-pin 0.3" DIL	50 µH
Pulse Engineering	PE64106	16-pin 0.3" DIL	50 µH
TDK	TLA 100-3E	16-pin 0.3" DIL	100 µH
Valor Electronics	LT6031	16-pin 0.3" DIL	50 µH

Contact the following people for further information on their products :

Bel Fuse	Tony Imburgia	Phone :	(317) 831-4226
		FAX :	(317) 831-4547
Halo Electronics	Jeffrey Heaton	Phone :	(415) 989-7313
		FAX :	(415) 367-7158
PCA Electronics	TBD	Phone :	TBD
		FAX :	TBD
Pulse Engineering	Mukesh Mehta	Phone :	(619) 268-2520
		FAX :	(619) 268-2515
Valor Electronics	Earnie Jensen	Phone :	(619) 458-1471
		FAX :	(619) 458-0875
Nano Pulse	Michael Schubert	Phone :	(714) 529-2600
		FAX :	(714) 671-7919
TDK	Kevin Umeda	Phone :	(213) 530-9397
		FAX :	(213) 530-8127