PRELIMINARY

AMD

Am79C961A

PCnet™-ISA II Jumperless, Full Duplex Single-Chip Ethernet Controller for ISA

DISTINCTIVE CHARACTERISTICS

- **Single-chip Ethernet controller for the Industry Standard Architecture (ISA) and Extended Industry Standard Architecture (EISA) buses**
- **Supports IEEE 802.3/ANSI 8802-3 and Ethernet standards**
- **Supports full duplex operation on the 10BASE-T, AUI, and GPSI ports**
- **Direct interface to the ISA or EISA bus**
- **Pin compatible to Am79C961 PCnet-ISA$^+$ Jumperless Single-Chip Ethernet Controller**
- **Software compatible with AMD's Am7990 LANCE register and descriptor architecture**
- **Low power, CMOS design with sleep mode allows reduced power consumption for critical battery powered applications**
- **Individual 136-byte transmit and 128-byte receive FIFOs provide packet buffering for increased system latency, and support the following features:**
  - — Automatic retransmission with no FIFO reload
  - — Automatic receive stripping and transmit padding (individually programmable)
  - — Automatic runt packet rejection
  - — Automatic deletion of received collision frames
- **Dynamic transmit FCS generation programmable on a frame-by-frame basis**
- **Single +5 V power supply**
- **Internal/external loopback capabilities**
- **Supports 8K, 16K, 32K, and 64K Boot PROMs or Flash for diskless node applications**
- **Supports Microsoft's Plug and Play System configuration for jumperless designs**
- **Supports staggered AT bus drive for reduced noise and ground bounce**
- **Supports 8 interrupts on chip**
- **Look Ahead Packet Processing (LAPP) allows protocol analysis to begin before end of receive frame**
- **Supports 4 DMA channels on chip**
- **Supports 16 I/O locations**
- **Supports 16 boot PROM locations**
- **Provides integrated Attachment Unit Interface (AUI) and 10BASE-T transceiver with 2 modes of port selection:**
  - — Automatic selection of AUI or 10BASE-T
  - — Software selection of AUI or 10BASE-T
- **Automatic Twisted Pair receive polarity detection and automatic correction of the receive polarity**
- **Supports bus-master, programmed I/O, and shared-memory architectures to fit in any PC application**
- **Supports edge and level-sensitive interrupts**
- **DMA Buffer Management Unit for reduced CPU intervention which allows higher throughput by by-passing the platform DMA**
- **JTAG Boundary Scan (IEEE 1149.1) test access port interface for board level production test**
- **Integrated Manchester Encoder/Decoder**
- **Supports the following types of network interfaces:**
  - — AUI to external 10BASE2, 10BASE5, 10BASE-T or 10BASE-F MAU
  - — Internal 10BASE-T transceiver with Smart Squelch to Twisted Pair medium
- **Supports LANCE General Purpose Serial Interface (GPSI)**
- **132-pin PQFP package**

Publication# **19364** Rev: **A** Amendment/**+3**
Issue Date: **September 1996**

GENERAL DESCRIPTION

The PCnet-ISA II controller, a single-chip Ethernet controller, is a highly integrated system solution for the PC-AT Industry Standard Architecture (ISA) architecture. It is designed to provide flexibility and compatibility with any existing PC application. This highly integrated 132-pin VLSI device is specifically designed to reduce parts count and cost, and addresses applications where higher system throughput is desired. The PCnet-ISA II controller is fabricated with AMD's advanced low-power CMOS process to provide low standby current for power sensitive applications.

The PCnet-ISA II controller can be configured into one of three different architecture modes to suit a particular PC application. In the Bus Master mode, all transfers are performed using the integrated DMA controller. This configuration enhances system performance by allowing the PCnet-ISA II controller to bypass the platform DMA controller and directly address the full 24-bit memory space. The implementation of Bus Master mode allows minimum parts count for the majority of PC applications. The PCnet-ISA II can also be configured as a Bus Slave with either a Shared Memory or Programmed I/O architecture for compatibility with low-end machines, such as PC/XTs that do not support Bus Masters, and high-end machines that require local packet buffering for increased system latency.

The PCnet-ISA II controller is designed to directly interface with the ISA or EISA system bus. It contains an ISA Plug and Play bus interface unit, DMA Buffer Management Unit, 802.3 Media Access Control function, individual 136-byte transmit and 128-byte receive FIFOs, IEEE 802.3 defined Attachment Unit Interface (AUI), and a Twisted Pair Transceiver Media Attachment Unit. Full duplex network operation can be enabled on any of the device's network ports. The PCnet-ISA II controller is also register compatible with the LANCE (Am7990) Ethernet controller and PCnet-ISA. The DMA Buffer Management Unit supports the LANCE descriptor software model. External remote boot and Ethernet physical address PROMs and Electrically Erasable Proms are also supported.

This advanced Ethernet controller has the built-in capability of automatically selecting either the AUI port or the Twisted Pair transceiver. Only one interface is active at any one time. The individual 136-byte transmit and 128-byte receive FIFOs optimize system overhead, providing sufficient latency during packet transmission and reception, and minimizing intervention during normal network error recovery. The integrated Manchester encoder/decoder eliminates the need for an external Serial Interface Adapter (SIA) in the node system. If support for an external encoding/decoding scheme is desired, the embedded General Purpose Serial Interface (GPSI) allows direct access to/from the MAC. In addition, the device provides programmable on-chip LED drivers for transmit, receive, collision, receive polarity, link integrity and activity, or jabber status. The PCnet-ISA II controller also provides an External Address Detection Interface™ (EADI™) to allow external hardware address filtering in internetworking applications.

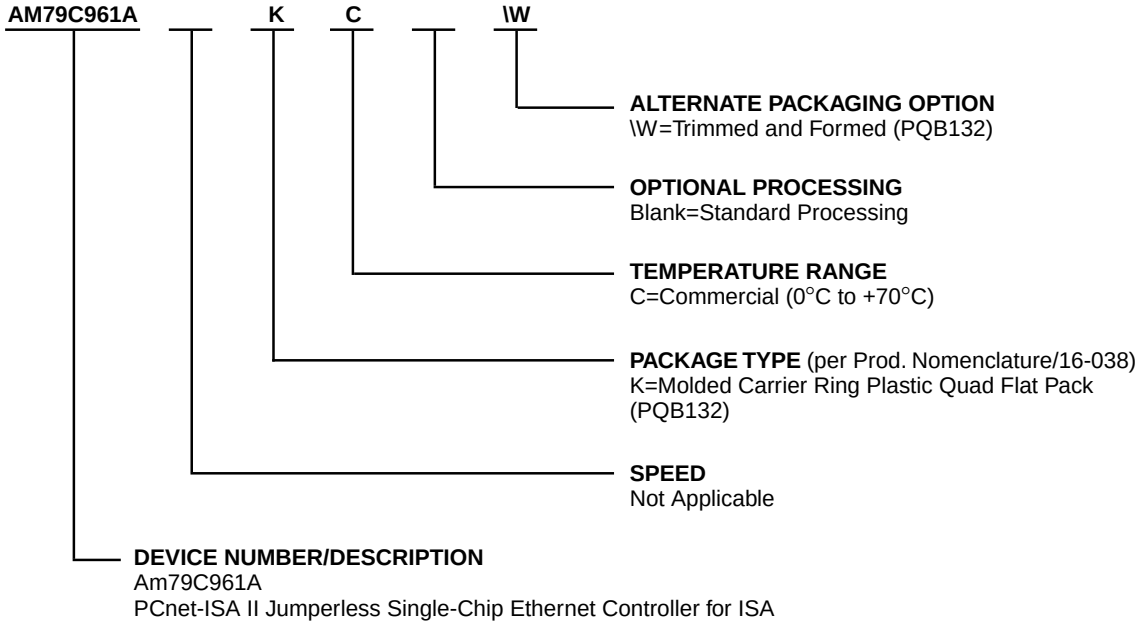
RELATED PRODUCTS

Part No.	Description
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA Jumperless Single-Chip Ethernet Controller (for ISA bus)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller (for 386, 486, VL local buses)
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Single-Chip Ethernet and SCSI Controller (for PCI bus)

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



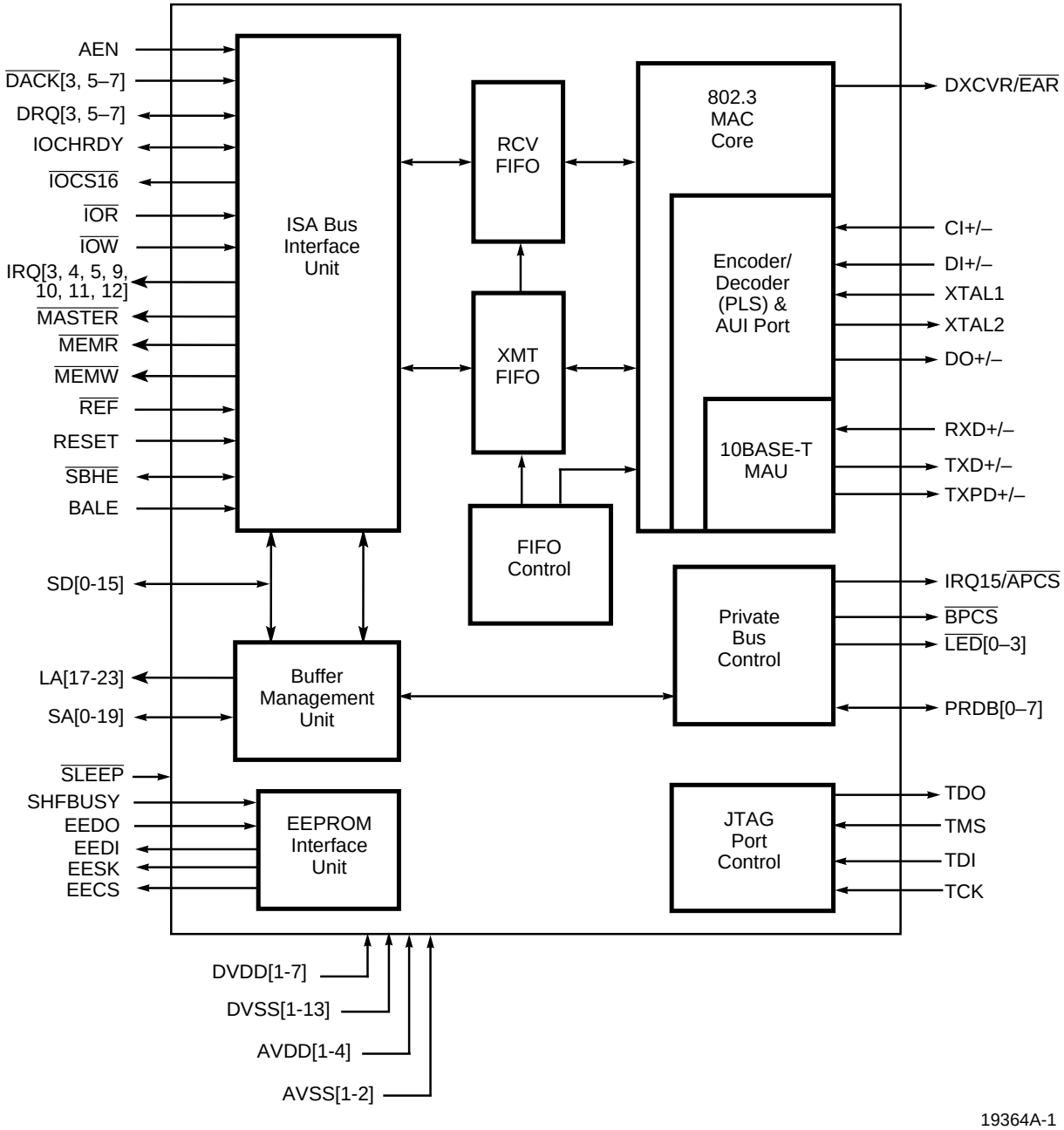
Valid Combinations	
AM79C961A	KC, KC\W

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

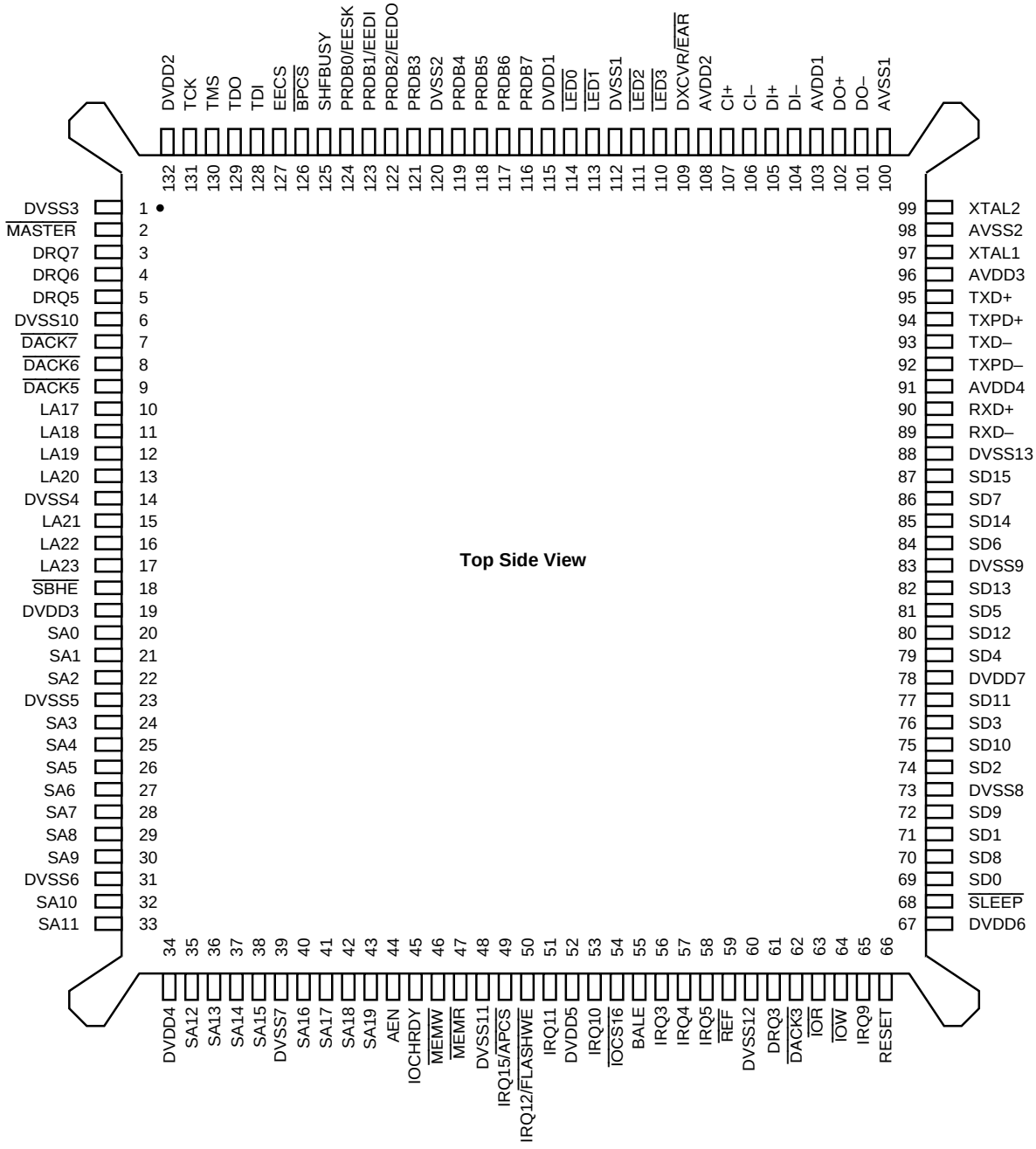
TABLE OF CONTENTS

BLOCK DIAGRAM: BUS MASTER MODE



19364A-1

CONNECTION DIAGRAMS: BUS MASTER MODE



19364A-2

PIN DESIGNATIONS: BUS MASTER MODE

Listed by Pin Number

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	DVSS3	34	DVDD4	67	DVDD6	100	AVSS1
2	$\overline{\text{MASTER}}$	35	SA12	68	$\overline{\text{SLEEP}}$	101	DO–
3	DRQ7	36	SA13	69	SD0	102	DO+
4	DRQ6	37	SA14	70	SD8	103	AVDD1
5	DRQ5	38	SA15	71	SD1	104	DI–
6	DVSS10	39	DVSS7	72	SD9	105	DI+
7	$\overline{\text{DACK7}}$	40	SA16	73	DVSS8	106	CI–
8	$\overline{\text{DACK6}}$	41	SA17	74	SD2	107	CI+
9	$\overline{\text{DACK5}}$	42	SA18	75	SD10	108	AVDD2
10	LA17	43	SA19	76	SD3	109	DXCVR/$\overline{\text{EAR}}$
11	LA18	44	AEN	77	SD11	110	$\overline{\text{LED3}}$
12	LA19	45	IOCHRDY	78	DVDD7	111	$\overline{\text{LED2}}$
13	LA20	46	$\overline{\text{MEMW}}$	79	SD4	112	DVSS1
14	DVSS4	47	$\overline{\text{MEMR}}$	80	SD12	113	$\overline{\text{LED1}}$
15	LA21	48	DVSS11	81	SD5	114	$\overline{\text{LED0}}$
16	LA22	49	IRQ15/$\overline{\text{APCS}}$	82	SD13	115	DVDD1
17	LA23	50	IRQ12/$\overline{\text{FlashWE}}$	83	DVSS9	116	PRDB7
18	$\overline{\text{SBHE}}$	51	IRQ11	84	SD6	117	PRDB6
19	DVDD3	52	DVDD5	85	SD14	118	PRDB5
20	SA0	53	IRQ10	86	SD7	119	PRDB4
21	SA1	54	$\overline{\text{IOCS16}}$	87	SD15	120	DVSS2
22	SA2	55	BALE	88	DVSS13	121	PRDB3
23	DVSS5	56	IRQ3	89	RXD–	122	PRDB2/EEDO
24	SA3	57	IRQ4	90	RXD+	123	PRDB1/EEDI
25	SA4	58	IRQ5	91	AVDD4	124	PRDB0/EESK
26	SA5	59	$\overline{\text{REF}}$	92	TXPD–	125	SHFBUSY
27	SA6	60	DVSS12	93	TXD–	126	$\overline{\text{BPCS}}$
28	SA7	61	DRQ3	94	TXPD+	127	EECS
29	SA8	62	$\overline{\text{DACK3}}$	95	TXD+	128	TDI
30	SA9	63	$\overline{\text{IOR}}$	96	AVDD3	129	TDO
31	DVSS6	64	$\overline{\text{IOW}}$	97	XTAL1	130	TMS
32	SA10	65	IRQ9	98	AVSS2	131	TCK
33	SA11	66	RESET	99	XTAL2	132	DVDD2

PIN DESIGNATIONS: BUS MASTER MODE

Listed by Pin Number

Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
AEN	44	DVSS12	60	LED2	111	SA6	27
AVDD1	103	DVSS13	88	LED3	110	SA7	28
AVDD2	108	DVSS2	120	$\overline{\text{MASTER}}$	2	SA8	29
AVDD3	96	DVSS3	1	$\overline{\text{MEMR}}$	47	SA9	30
AVDD4	91	DVSS4	14	$\overline{\text{MEMW}}$	46	$\overline{\text{SBHE}}$	18
AVSS1	100	DVSS5	23	PRDB0/EESK	124	SD0	69
AVSS2	98	DVSS6	31	PRDB1/EEDI	123	SD1	71
BALE	55	DVSS7	39	PRDB2/EEDO	122	SD10	75
BPCS	126	DVSS8	73	PRDB3	121	SD11	77
CI–	106	DVSS9	83	PRDB4	119	SD12	80
CI+	107	DXCVR/$\overline{\text{EAR}}$	109	PRDB5	118	SD13	82
$\overline{\text{DACK3}}$	62	EECS	127	PRDB6	117	SD14	85
$\overline{\text{DACK5}}$	9	IOCHRDY	45	PRDB7	116	SD15	87
$\overline{\text{DACK6}}$	8	$\overline{\text{IOCS16}}$	54	REF	59	SD2	74
$\overline{\text{DACK7}}$	7	$\overline{\text{IOR}}$	63	RESET	66	SD3	76
DI–	104	$\overline{\text{IOW}}$	64	RXD–	89	SD4	79
DI+	105	IRQ10	53	RXD+	90	SD5	81
DO–	101	IRQ11	51	SA0	20	SD6	84
DO+	102	IRQ12/$\overline{\text{FlashWE}}$	50	SA1	21	SD7	86
DRQ3	61	IRQ15/$\overline{\text{APCS}}$	49	SA10	32	SD8	70
DRQ5	5	IRQ3	56	SA11	33	SD9	72
DRQ6	4	IRQ4	57	SA12	35	SHFBUSY	125
DRQ7	3	IRQ5	58	SA13	36	$\overline{\text{SLEEP}}$	68
DVDD1	115	IRQ9	65	SA14	37	TCK	131
DVDD2	132	LA17	10	SA15	38	TDI	128
DVDD3	19	LA18	11	SA16	40	TDO	129
DVDD4	34	LA19	12	SA17	41	TMS	130
DVDD5	52	LA20	13	SA18	42	TXD–	93
DVDD6	67	LA21	15	SA19	43	TXD+	95
DVDD7	78	LA22	16	SA2	22	TXPD–	92
DVSS1	112	LA23	17	SA3	24	TXPD+	94
DVSS10	6	LED0	114	SA4	25	XTAL1	97
DVSS11	48	LED1	113	SA5	26	XTAL2	99

PIN DESIGNATIONS: BUS MASTER MODE

Listed by Group

Pin Name	Pin Function	I/O	Driver
ISA Bus Interface			
AEN	Address Enable	I	
BALE	Bus Address Latch Enable	I	
$\overline{\text{DACK}}$[3, 5–7]	DMA Acknowledge	I	
DRQ[3, 5–7]	DMA Request	I/O	TS3
IOCHRDY	I/O Channel Ready	I/O	OD3
$\overline{\text{IOCS16}}$	I/O Chip Select 16	O	OD3
$\overline{\text{IOR}}$	I/O Read Select	I	
$\overline{\text{IOW}}$	I/O Write Select	I	
IRQ[3, 4, 5, 9, 10, 11, 12, 15]	Interrupt Request	O	TS3/OD3
LA[17-23]	Unlatched Address Bus	I/O	TS3
$\overline{\text{MASTER}}$	Master Transfer in Progress	O	OD3
$\overline{\text{MEMR}}$	Memory Read Select	O	TS3
$\overline{\text{MEMW}}$	Memory Write Select	O	TS3
$\overline{\text{REF}}$	Memory Refresh Active	I	
RESET	System Reset	I	
SA[0 –19]	System Address Bus	I/O	TS3
$\overline{\text{SBHE}}$	System Byte High Enable	I/O	TS3
SD[0 –15]	System Data Bus	I/O	TS3
Board Interfaces			
IRQ15/$\overline{\text{APCS}}$	IRQ15 or Address PROM Chip Select	O	TS1
$\overline{\text{BPCS}}$	Boot PROM Chip Select	O	TS1
DXCVR/$\overline{\text{EAR}}$	Disable Transceiver	I/O	TS1
$\overline{\text{LED0}}$	$\overline{\text{LED0/LNKST}}$	O	TS2
$\overline{\text{LED1}}$	$\overline{\text{LED1}}$/SFBD/$\overline{\text{RCVACT}}$	O	TS2
$\overline{\text{LED2}}$	$\overline{\text{LED2}}$/SRD/$\overline{\text{RXDATPOL}}$	O	TS2
$\overline{\text{LED3}}$	$\overline{\text{LED3}}$/SRDCLK/$\overline{\text{XMTACT}}$	O	TS2
PRDB[3–7]	PROM Data Bus	I/O	TS1
$\overline{\text{SLEEP}}$	Sleep Mode	I	
XTAL1	Crystal Input	I	
XTAL2	Crystal Output	O	
SHFBUSY	Read access from EEPROM in process	I/O	
PRDB(0)/EESK	Serial Shift Clock	I/O	
PRDB(1)/EEDI	Serial Shift Data In	I/O	
PRDB(2)/EEDO	Serial Shift Data Out	O	
EECS	EEPROM Chip Select		

PIN DESIGNATIONS: BUS MASTER MODE (continued)

Listed by Group

Pin Name	Pin Function	I/O	Driver
Attachment Unit Interface (AUI)			
CI±	Collision Inputs	I	
DI±	Receive Data	I	
DO±	Transmit Data	O	
Twisted Pair Transceiver Interface (10BASE-T)			
RXD±	10BASE-T Receive Data	I	
TXD±	10BASE-T Transmit Data	O	
TXPD±	10BASE-T Predistortion Control	O	
IEEE 1149.1 Test Access Port Interface (JTAG)			
TCK	Test Clock	I	
TDI	Test Data Input	I	
TDO	Test Data Output	O	TS2
TMS	Test Mode Select	I	
Power Supplies			
AVDD	Analog Power [1-4]		
AVSS	Analog Ground [1-2]		
DVDD	Digital Power [1-7]		
DVSS	Digital Ground [1-13]		

Output Driver Types

Name	Type	IOL (mA)	IOH (mA)	pF
TS1	Tri-State	4	–1	50
TS2	Tri-State	12	–4	50
TS3	Tri-State	24	–3	120
OD3	Open Drain	24	–3	120

PIN DESCRIPTION: BUS MASTER MODE

These pins are part of the bus master mode. In order to understand the pin descriptions, definition of some terms from a draft of IEEE P996 are included.

IEEE P996 Terminology

Alternate Master: Any device that can take control of the bus through assertion of the $\overline{\text{MASTER}}$ signal. It has the ability to generate addresses and bus control signals in order to perform bus operations. All Alternate Masters must be 16 bit devices and drive $\overline{\text{SBHE}}$**.**

Bus Ownership: The Current Master possesses bus ownership and can assert any bus control, address and data lines.

Current Master: The Permanent Master, Temporary Master or Alternate Master which currently has ownership of the bus.

Permanent Master: Each P996 bus will have a device known as the Permanent Master that provides certain signals and bus control functions as described in Section 3.5 (of the IEEE P996 spec.), "Permanent Master". The Permanent Master function can reside on a Bus Adapter or on the backplane itself.

Temporary Master: A device that is capable of generating a DMA request to obtain control of the bus and directly asserting only the memory and I/O strobes during bus transfer. Addresses are generated by the DMA device on the Permanent Master.

ISA Interface

AEN

Address Enable ***Input***

This signal must be driven LOW when the bus performs an I/O access to the device.

BALE

Used to latch the LA20–23 address lines.

$\overline{\text{DACK}}$ 3, 5-7

DMA Acknowledge ***Input***

Asserted LOW when the Permanent Master acknowledges a DMA request. When $\overline{\text{DACK}}$ is asserted the PCnet-ISA II controller becomes the Current Master by asserting the $\overline{\text{MASTER}}$ signal.

DRQ 3, 5-7

DMA Request ***Input/Output***

When the PCnet-ISA II controller needs to perform a DMA transfer, it asserts DRQ. The Permanent Master acknowledges DRQ with the assertion of $\overline{\text{DACK}}$. When the PCnet-ISA II does not need the bus it desserts DRQ. The PCnet-ISA II provides for fair bus bandwidth sharing between two bus mastering devices on the ISA bus through an adaptive delay which is inserted between back-to-back DMA requests. See the Back-to-Back DMA Requests section for details.

Because of the operation of the Plug and Play registers, the DMA Channels on the PCnet-ISA II must be attached to the specific DRQ and DACK signals on the PC/AT bus as indicated by the pin names.

IOCHRDY

I/O Channel Ready ***Input/Output***

When the PCnet-ISA II controller is being accessed, IOCHRDY HIGH indicates that valid data exists on the data bus for reads and that data has been latched for writes. When the PCnet-ISA II controller is the Current Master on the ISA bus, it extends the bus cycle as long as IOCHRDY is LOW.

$\overline{\text{IOCS16}}$

I/O Chip Select 16 ***Output***

When an I/O read or write operation is performed, the PCnet-ISA II controller will drive the $\overline{\text{IOCS16}}$ pin LOW to indicate that the chip supports a 16-bit operation at this address. (If the motherboard does not receive this signal, then the motherboard will convert a 16-bit access to two 8-bit accesses).

The PCnet-ISA II controller follows the IEEE P996 specification that recommends this function be implemented as a pure decode of SA0-9 and AEN, with no dependency on $\overline{\text{IOR}}$, or $\overline{\text{IOW}}$; however, some PC/AT clone systems are not compatible with this approach. For this reason, the PCnet-ISA II controller is recommended to be configured to run 8-bit I/O on all machines. Since data is moved by memory cycles there is virtually no performance loss incurred by running 8-bit I/O and compatibility problems are virtually eliminated. The PCnet-ISA II controller can be configured to run 8-bit-only I/O by clearing Bit 0 in Plug and Play register F0.

$\overline{\text{IOR}}$

I/O Read ***Input***

$\overline{\text{IOR}}$ is driven LOW by the host to indicate that an Input/Output Read operation is taking place. $\overline{\text{IOR}}$ is only valid if the AEN signal is LOW and the external address matches the PCnet-ISA II controller's predefined I/O address location. If valid, $\overline{\text{IOR}}$ indicates that a slave read operation is to be performed.

$\overline{\text{IOW}}$

I/O Write ***Input***

$\overline{\text{IOW}}$ is driven LOW by the host to indicate that an Input/Output Write operation is taking place. $\overline{\text{IOW}}$ is only valid if AEN signal is LOW and the external address matches the PCnet-ISA II controller's predefined I/O address location. If valid, $\overline{\text{IOW}}$ indicates that a slave write operation is to be performed.

IRQ 3, 4, 5, 9, 10, 11, 12, 15

Interrupt Request ***Output***

An attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON, RCVCCO, JAB, MPCO, or TXDATSTRT. All status flags have a mask bit which allows for suppression of IRQ assertion. These flags have the following meaning:

BABL	Babble
RCVCCO	Receive Collision Count Overflow
JAB	Jabber
MISS	Missed Frame
MERR	Memory Error
MPCO	Missed Packet Count Overflow
RINT	Receive Interrupt
IDON	Initialization Done
TXDATSTRT	Transmit Start

Because of the operation of the Plug and Play registers, the interrupts on the PCnet-ISA II must be attached to specific IRQ signals on the PC/AT bus.

LA17-23

Unlatched Address Bus ***Input/Output***

The unlatched address bus is driven by the PCnet-ISA II controller during bus master cycle.

The functions of these unlatched address pins will change when GPSI mode is invoked. The following table shows the pin configuration in GPSI mode. Please refer to the section on General Purpose Serial Interface for detailed information on accessing this mode.

Pin Number	Pin Function in Bus Master Mode	Pin Function in GPSI Mode
10	LA17	RXDAT
11	LA18	SRDCLK
12	LA19	RXCRS
13	LA20	CLSN
15	LA21	STDCLK
16	LA22	TXEN
17	LA23	TXDAT

$\overline{\text{MASTER}}$

Master Mode ***Input/Output***

This signal indicates that the PCnet-ISA II controller has become the Current Master of the ISA bus. After the PCnet-ISA II controller has received a DMA Acknowledge ($\overline{\text{DACK}}$) in response to a DMA Request (DRQ), the Ethernet controller asserts the $\overline{\text{MASTER}}$ signal to indicate to the Permanent Master that the PCnet-ISA II controller is becoming the Current Master.

$\overline{\text{MEMR}}$

Memory Read ***Input/Output***

$\overline{\text{MEMR}}$ goes LOW to perform a memory read operation.

$\overline{\text{MEMW}}$

Memory Write ***Input/Output***

$\overline{\text{MEMW}}$ goes LOW to perform a memory write operation.

$\overline{\text{REF}}$

Memory Refresh ***Input***

When $\overline{\text{REF}}$ is asserted, a memory refresh is active. The PCnet-ISA II controller uses this signal to mask inadvertent DMA Acknowledge assertion during memory refresh periods. If $\overline{\text{DACK}}$ is asserted when $\overline{\text{REF}}$ is active, $\overline{\text{DACK}}$ assertion is ignored. $\overline{\text{REF}}$ is monitored to eliminate a bus arbitration problem observed on some ISA platforms.

RESET

Reset ***Input***

When RESET is asserted HIGH the PCnet-ISA II controller performs an internal system reset. RESET must be held for a minimum of 10 XTAL1 periods before being deasserted. While in a reset state, the PCnet-ISA II controller will tristate or deassert all outputs to predefined reset levels. The PCnet-ISA II controller resets itself upon power-up.

SA0-19

System Address Bus ***Input/Output***

This bus contains address information, which is stable during a bus operation, regardless of the source. SA17-19 contain the same values as the unlatched address LA17-19. When the PCnet-ISA II controller is the Current Master, SA0-19 will be driven actively. When the PCnet-ISA II controller is not the Current Master, the SA0-19 lines are continuously monitored to determine if an address match exists for I/O slave transfers or Boot PROM accesses.

$\overline{\text{SBHE}}$

System Byte High Enable ***Input/Output***

This signal indicates the high byte of the system data bus is to be used. $\overline{\text{SBHE}}$ is driven by the PCnet-ISA II controller when performing bus mastering operations.

SD0-15

System Data Bus ***Input/Output***

These pins are used to transfer data to and from the PCnet-ISA II controller to system resources via the ISA data bus. SD0-15 is driven by the PCnet-ISA II control-

ler when performing bus master writes and slave read operations. Likewise, the data on SD0-15 is latched by the PCnet-ISA II controller when performing bus master reads and slave write operations.

Board Interface

IRQ12/$\overline{\text{FlashWE}}$

Flash Write Enable ***Output***

Optional interface to the Flash memory boot PROM Write Enable.

IRQ15/$\overline{\text{APCS}}$

Address PROM Chip Select ***Output***

When programmed as $\overline{\text{APCS}}$ in Plug and Play Register F0, this signal is asserted when the external Address PROM is read. When an I/O read operation is performed on the first 16 bytes in the PCnet-ISA II controller's I/O space, $\overline{\text{APCS}}$ is asserted. The outputs of the external Address PROM drive the PROM Data Bus. The PCnet-ISA II controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus.

When programmed to IRQ15 (default), this pin has the same function as IRQ 3, 4, 5, 9, 10, 11, or 12.

$\overline{\text{BPCS}}$

Boot PROM Chip Select ***Output***

This signal is asserted when the Boot PROM is read. If SA0-19 lines match a predefined address block and $\overline{\text{MEMR}}$ is active and $\overline{\text{REF}}$ inactive, the $\overline{\text{BPCS}}$ signal will be asserted. The outputs of the external Boot PROM drive the PROM Data Bus. The PCnet-ISA II controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus.

DXCVR/$\overline{\text{EAR}}$

Disable Transceiver/
External Address Reject ***Input/Output***

This pin can be used to disable external transceiver circuitry attached to the AUI interface when the internal 10BASE-T port is active. The polarity of this pin is set by the DXCVRP bit (PnP register 0xF0, bit 5). When DXCVRP is cleared (default), the DXCVR pin is driven HIGH when the Twisted Pair port is active or SLEEP mode has been entered and driven LOW when the AUI port is active. When DXCVRP is set, the DXCVR pin is driven LOW when the Twisted Pair port is active or SLEEP mode has been entered and driven HIGH when the AUI port is active.

If EADI mode is selected, this pin becomes the $\overline{\text{EAR}}$ input.

The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be OR'd with the value on the $\overline{\text{EAR}}$ pin. The $\overline{\text{EAR}}$ pin is defined as $\overline{\text{REJECT}}$. (See the EADI section for details regarding the function and timing of this signal).

$\overline{\text{LEDO-3}}$

LED Drivers ***Output***

These pins sink 12 mA each for driving LEDs. Their meaning is software configurable (see section *The ISA Bus Configuration Registers*) and they are active LOW.

When EADI mode is selected, the pins named $\overline{\text{LED1}}$, $\overline{\text{LED2}}$, and $\overline{\text{LED3}}$ change in function while $\overline{\text{LED0}}$ continues to indicate 10BASE-T Link Status.

LED	EADI Function
1	SF/BD
2	SRD
3	SRDCLK

PRDB3-7

Private Data Bus ***Input/Output***

This is the data bus for the Boot PROM and the Address PROM.

PRDB2/EEDO

Private data bus bit 2/Data Out ***Input/Output***

A multifunction pin which serves as PRDB2 of the private data bus and, when ISACSR3 bit 4 is set, changes to become DATA OUT from the EEPROM.

PRDB1/EEDI

Private data bus bit 1/Data In ***Input/Output***

A multifunction pin which serves as PRDB1 of the private data bus and, when ISACSR3 bit 4 is set, changes to become DATA In to the EEPROM.

PRDB0/EESK

Private data bus bit 0/
Serial Clock ***Input/Output***

A multifunction pin which serves as PRDB0 of the private data bus and, when ISACSR3 bit 4 is set, changes to become Serial Clock to the EEPROM.

SHFBUSY

Shift Busy ***Input/Output***

This pin indicates that a read from the external EEPROM is in progress. It is active only when data is being shifted out of the EEPROM due to a hardware RESET or assertion of the EE_LOAD bit (ISACSR3, bit 14). If this pin is left unconnected or pulled low with a pull-down resistor, an EEPROM checksum error is forced. Normally, this pin should be connected to V_{CC} through a 10K Ω pull-up resistor.

EECS

EEPROM CHIP SELECT ***Output***

This signal is asserted when read or write accesses are being performed to the EEPROM. It is controlled by ISACSR3. It is driven at Reset during EEPROM Read.

$\overline{\text{SLEEP}}$

Sleep ***Input***

When $\overline{\text{SLEEP}}$ pin is asserted (active LOW), the PCnet-ISA II controller performs an internal system reset and proceeds into a power savings mode. All outputs will be placed in their normal reset condition. All PCnet-ISA II controller inputs will be ignored except for the $\overline{\text{SLEEP}}$ pin itself. Deassertion of $\overline{\text{SLEEP}}$ results in the device waking up. The system must delay the starting of the network controller by 0.5 seconds to allow internal analog circuits to stabilize.

XTAL1

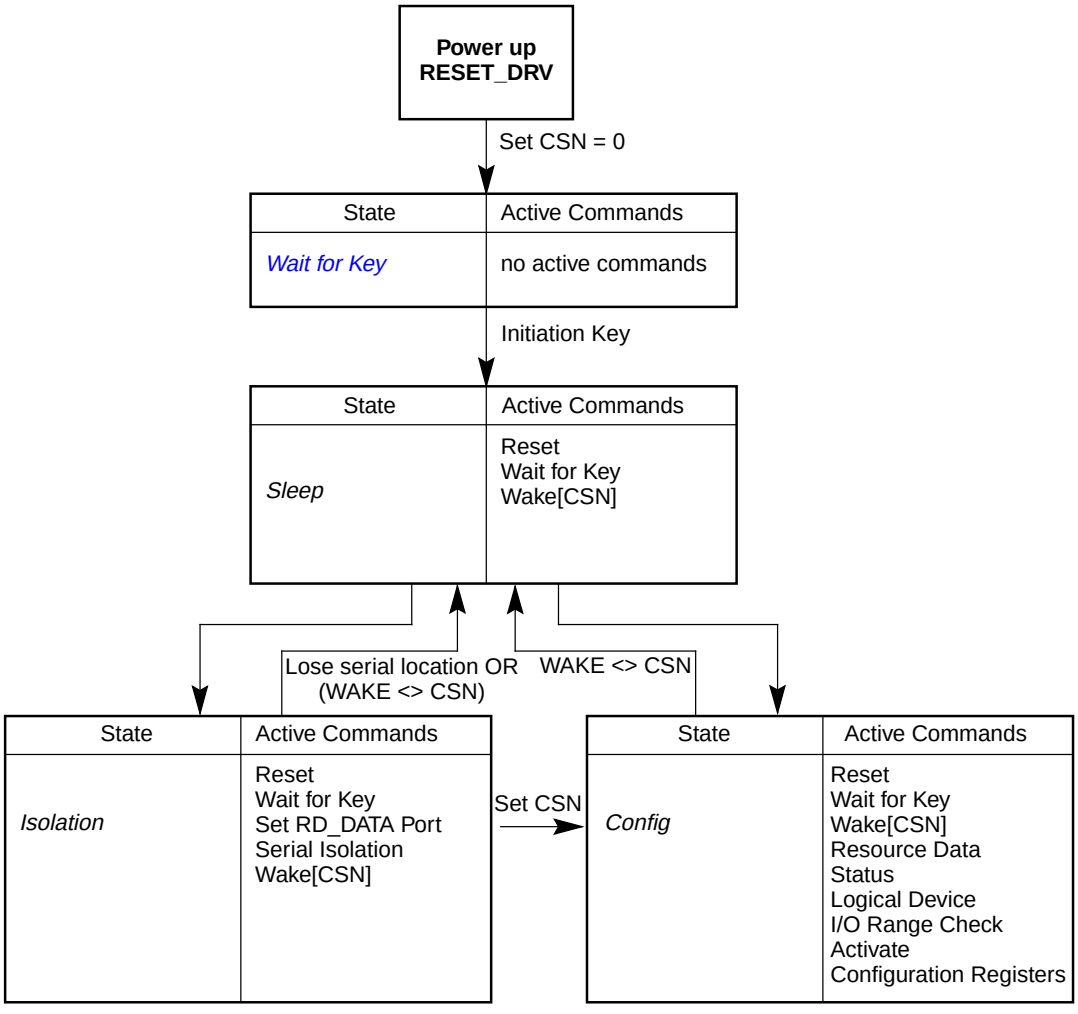
Crystal Connection ***Input***

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. Alternatively, an external 20 MHz CMOS-compatible clock signal can be used to drive this pin. Refer to the section on External Crystal Characteristics for more details.

XTAL2

Crystal Connection ***Output***

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. If an external clock is used, this pin should be left unconnected.

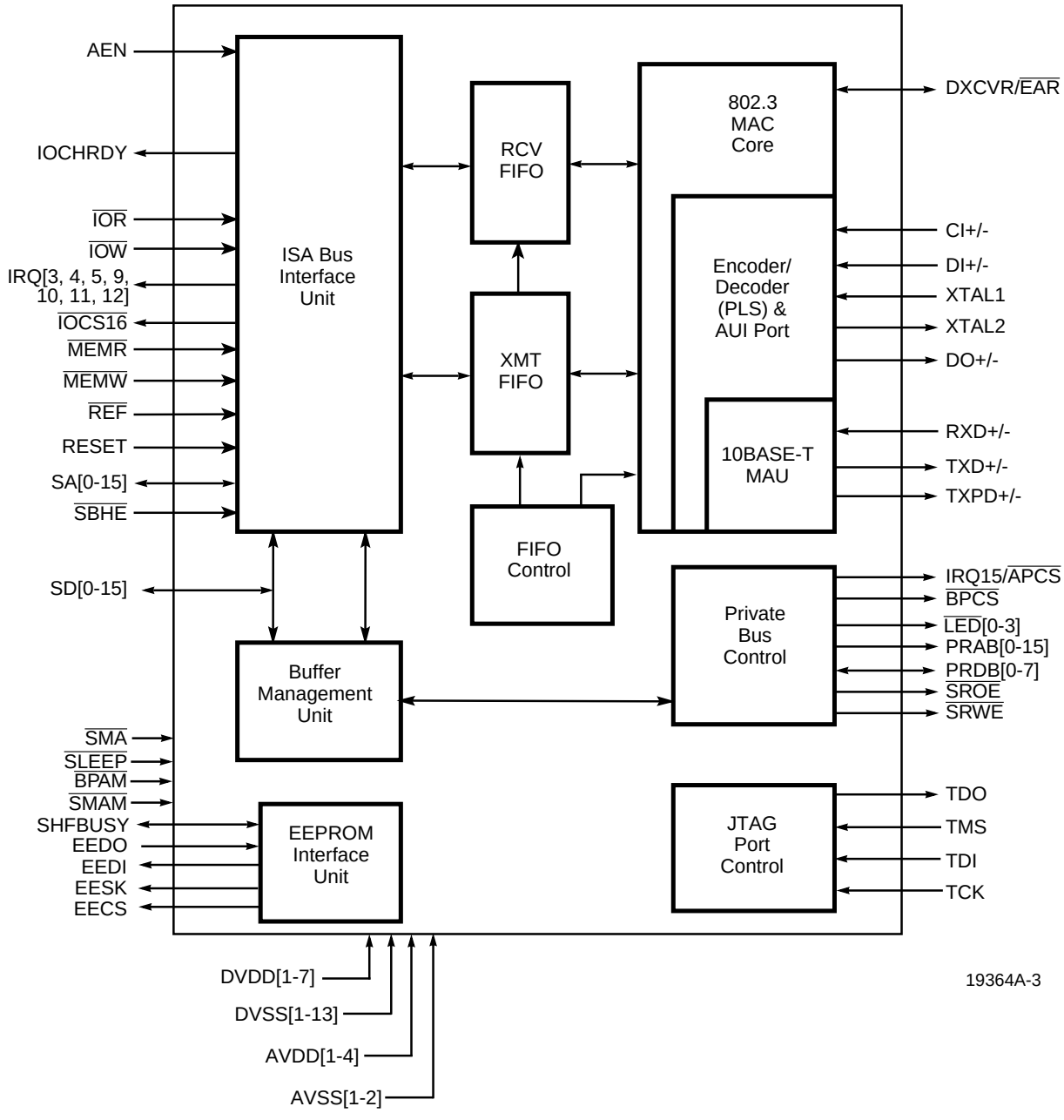


Notes:

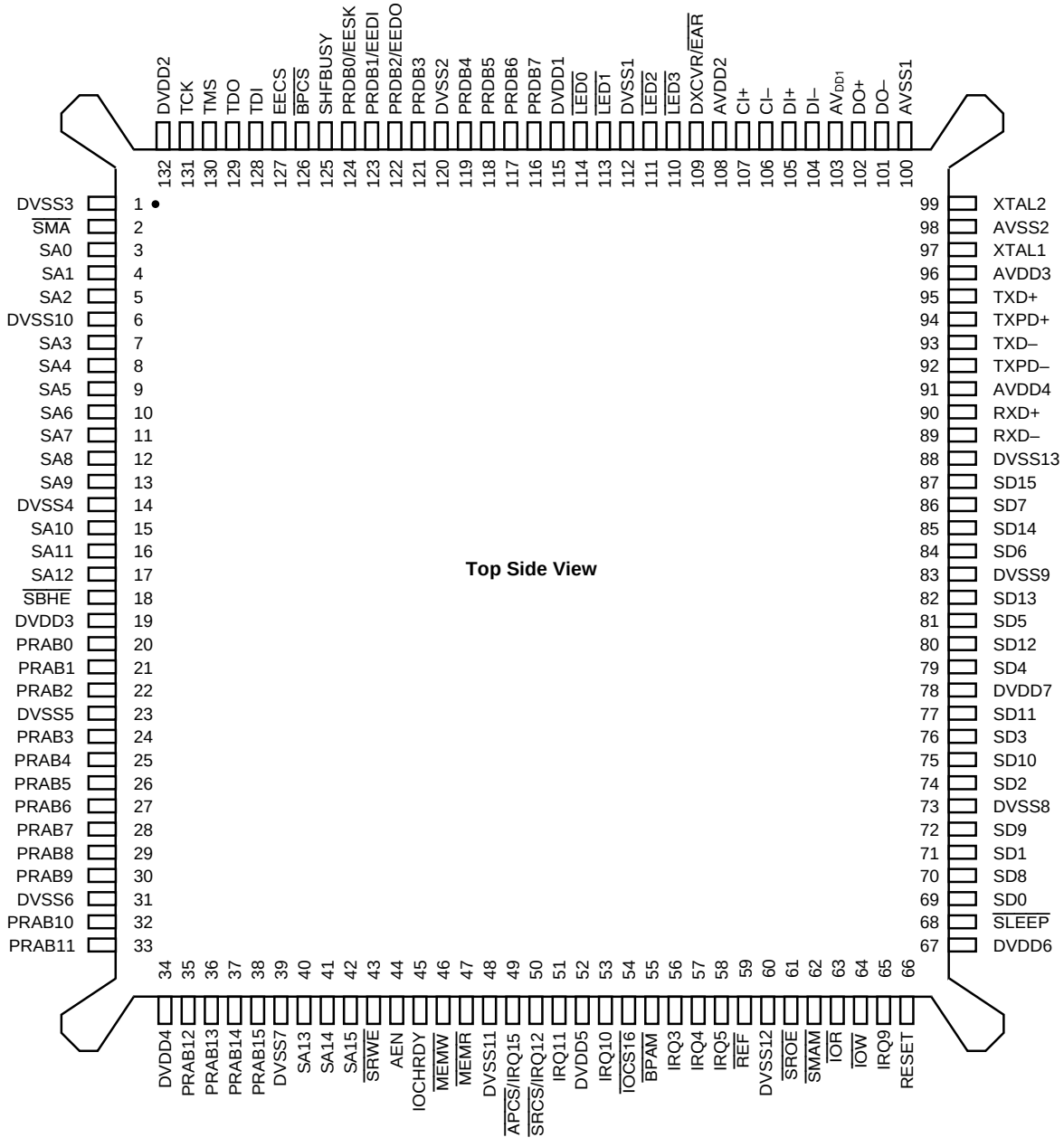
1. *CSN = Card Select Number.*
2. ***RESET_DRV causes a state transition from the current state to Wait for Key and sets all CSNs to zero. All logical devices are set to their power-up configuration values.***
3. *The Wait for Key command causes a state transition from the current state to Wait for Key.*

Plug and Play ISA Card State Transitions

BLOCK DIAGRAM: BUS SLAVE MODE



CONNECTION DIAGRAMS: BUS SLAVE MODE



19364A-4

PIN DESIGNATIONS: BUS SLAVE MODE

Listed by Pin Number

Pin #	Name	Pin #	Name	Pin #	Name
1	DVSS3	45	IOCHRDY	89	RXD-
2	$\overline{\text{SMA}}$	46	$\overline{\text{MEMW}}$	90	RXD+
3	SA0	47	$\overline{\text{MEMR}}$	91	AVDD4
4	SA1	48	DVSS11	92	TXPD-
5	SA2	49	IRQ15	93	TXD-
6	DVSS10	50	IRQ12	94	TXPD+
7	SA3	51	IRQ11	95	TXD+
8	SA4	52	DVDD5	96	AVDD3
9	SA5	53	IRQ10	97	XTAL1
10	SA6	54	$\overline{\text{IOCS16}}$	98	AVSS2
11	SA7	55	$\overline{\text{BPAM}}$	99	XTAL2
12	SA8	56	IRQ3	100	AVSS1
13	SA9	57	IRQ4	101	DO-
14	DVSS4	58	IRQ5	102	DO+
15	SA10	59	$\overline{\text{REF}}$	103	AVDD1
16	SA11	60	DVSS12	104	DI-
17	SA12	61	$\overline{\text{SROE}}$	105	DI+
18	$\overline{\text{SBHE}}$	62	$\overline{\text{SMAM}}$	106	CI-
19	DVDD3	63	$\overline{\text{IOR}}$	107	CI+
20	PRAB0	64	$\overline{\text{IOW}}$	108	AVDD2
21	PRAB1	65	IRQ9	109	DXCVR/$\overline{\text{EAR}}$
22	PRAB2	66	RESET	110	$\overline{\text{LED3}}$
23	DVSS5	67	DVDD6	111	$\overline{\text{LED2}}$
24	PRAB3	68	$\overline{\text{SLEEP}}$	112	DVSS1
25	PRAB4	69	SD0	113	$\overline{\text{LED1}}$
26	PRAB5	70	SD8	114	$\overline{\text{LED0}}$
27	PRAB6	71	SD1	115	DVDD1
28	PRAB7	72	SD9	116	PRDB7
29	PRAB8	73	DVSS8	117	PRDB6
30	PRAB9	74	SD2	118	PRDB5
31	DVSS6	75	SD10	119	PRDB4
32	PRAB10	76	SD3	120	DVSS2
33	PRAB11	77	SD11	121	PRDB3
34	DVDD4	78	DVDD7	122	PRDB2/EEDO
35	PRAB12	79	SD4	123	PRDB1/EEDI
36	PRAB13	80	SD12	124	PRDB0/EESK
37	PRAB14	81	SD5	125	SHFBUSY
38	PRAB15	82	SD13	126	$\overline{\text{BPCS}}$
39	DVSS7	83	DVSS9	127	EECS
40	SA13	84	SD6	128	TDI
41	SA14	85	SD14	129	TDO
42	SA15	86	SD7	130	TMS
43	$\overline{\text{SRWE}}$	87	SD15	131	TCK
44	AEN	88	DVSS13	132	DVDD2

PIN DESIGNATIONS: BUS SLAVE MODE

Listed by Pin Name

Name	Pin#	Name	Pin#	Name	Pin#
AEN	44	IRQ15	49	SA13	40
AVDD1	103	IRQ3	56	SA14	41
AVDD2	108	IRQ4	57	SA15	42
AVDD3	96	IRQ5	58	SA2	5
AVDD4	91	IRQ9	65	SA3	7
AVSS1	100	$\overline{\text{LED0}}$	114	SA4	8
AVSS2	98	$\overline{\text{LED1}}$	113	SA5	9
$\overline{\text{BPAM}}$	55	$\overline{\text{LED2}}$	111	SA6	10
$\overline{\text{BPCS}}$	126	$\overline{\text{LED3}}$	110	SA7	11
CI-	106	$\overline{\text{MEMR}}$	47	SA8	12
CI+	107	$\overline{\text{MEMW}}$	46	SA9	13
DI-	104	PRAB0	20	$\overline{\text{SBHE}}$	18
DI+	105	PRAB1	21	SD0	69
DO-	101	PRAB10	32	SD1	71
DO+	102	PRAB11	33	SD10	75
DVDD1	115	PRAB12	35	SD11	77
DVDD2	132	PRAB13	36	SD12	80
DVDD3	19	PRAB14	37	SD13	82
DVDD4	34	PRAB15	38	SD14	85
DVDD5	52	PRAB2	22	SD15	87
DVDD6	67	PRAB3	24	SD2	74
DVDD7	78	PRAB4	25	SD3	76
DVSS1	112	PRAB5	26	SD4	79
DVSS10	6	PRAB6	27	SD5	81
DVSS11	48	PRAB7	28	SD6	84
DVSS12	60	PRAB8	29	SD7	86
DVSS13	88	PRAB9	30	SD8	70
DVSS2	120	PRDB0/DO	124	SD9	72
DVSS3	1	PRDB0/D1	123	SHFBUSY	125
DVSS4	14	PRDB0/SCLK	122	$\overline{\text{SLEEP}}$	68
DVSS5	23	PRDB3	121	$\overline{\text{SMA}}$	2
DVSS6	31	PRDB4	119	$\overline{\text{SMAM}}$	62
DVSS7	39	PRDB5	118	$\overline{\text{SROE}}$	61
DVSS8	73	PRDB6	117	$\overline{\text{SRWE}}$	43
DVSS9	83	PRDB7	116	TCK	131
DXCVR/$\overline{\text{EAR}}$	109	$\overline{\text{REF}}$	59	TDI	128
EECS	127	RESET	66	TDO	129
IOCHRDY	45	RXD-	89	TMS	130
$\overline{\text{IOCS16}}$	54	RXD+	90	TXD-	93
$\overline{\text{IOR}}$	63	SA0	3	TXD+	95
$\overline{\text{IOW}}$	64	SA1	4	TXPD-	92
IRQ10	53	SA10	15	TXPD+	94
IRQ11	51	SA11	16	XTAL1	97
IRQ12	50	SA12	17	XTAL2	99

PIN DESIGNATIONS: BUS SLAVE MODE

Listed by Group

Pin Name	Pin Function	I/O	Driver
ISA Bus Interface			
AEN	Address Enable	I	
IOCHRDY	I/O Channel Ready	O	OD3
$\overline{\text{IOCS16}}$	I/O Chip Select 16	O	OD3
$\overline{\text{IOR}}$	I/O Read Select	I	
$\overline{\text{IOW}}$	I/O Write Select	I	
IRQ[3, 4, 5, 9, 10, 11, 12, 15]	Interrupt Request	O	TS3/OD3
$\overline{\text{MEMR}}$	Memory Read Select	I	
$\overline{\text{MEMW}}$	Memory Write Select	I	
REF	Memory Refresh Active	I	
RESET	System Reset	I	
SA[0–15]	System Address Bus	I	
$\overline{\text{SBHE}}$	System Byte High Enable	I	
SD[0–15]	System Data Bus	I/O	TS3
Board Interfaces			
IRQ15/$\overline{\text{APCS}}$	IRQ15 or Address PROM Chip Select	O	TS1
$\overline{\text{BPCS}}$	Boot PROM Chip Select	O	TS1
$\overline{\text{BPAM}}$	Boot PROM Address Match	I	
DXCVR/$\overline{\text{EAR}}$	Disable Transceiver	I/O	TS1
$\overline{\text{LED0}}$	$\overline{\text{LED0}}$/$\overline{\text{LNKST}}$	O	TS2
$\overline{\text{LED1}}$	$\overline{\text{LED1}}$/$\overline{\text{SFBD}}$/$\overline{\text{RCVACT}}$	O	TS2
$\overline{\text{LED2}}$	$\overline{\text{LED2}}$/$\overline{\text{SRD}}$/$\overline{\text{RXDATD01}}$	O	TS2
$\overline{\text{LED3}}$	$\overline{\text{LED3}}$/$\overline{\text{SRDCLK}}$/$\overline{\text{XMTACT}}$	O	TS2
PRAB[0–15]	PRivate Address Bus	I/O	TS3
PRDB[3–7]	PRivate Data Bus	I/O	TS1
$\overline{\text{SLEEP}}$	Sleep Mode	I	
$\overline{\text{SMA}}$	**Slave Mode Architecture**	**I**	
$\overline{\text{SMAM}}$	Shared Memory Address Match	I	
$\overline{\text{SROE}}$	Static RAM Output Enable	O	TS3
$\overline{\text{SRWE}}$	Static RAM Write Enable	O	TS1
XTAL1	Crystal Oscillator Input	I	
XTAL2	Crystal Oscillator OUTPUT	O	
SHFBUSY	Read access from EEPROM in process	O	
PRDB(0)/EESK	Serial Shift Clock	I/O	
PRDB(1)/EEDI	Serial Shift Data In	I/O	
PRDB(2)/EEDO	Serial Shift Data Out	I/O	
EECS	EEPROM Chip Select	O	

PIN DESIGNATIONS: BUS SLAVE MODE

Listed by Group

Pin Name	Pin Function	I/O	Driver
Attachment Unit Interface (AUI)			
CI±	Collision Inputs	I	
DI±	Receive Data	I	
DO±	Transmit Data	O	
Twisted Pair Transceiver Interface (10BASE-T)			
RXD±	10BASE-T Receive Data	I	
TXD±	10BASE-T Transmit Data	O	
TXPD±	10BASE-T Predistortion Control	O	
IEEE 1149.1 Test Access Port Interface (JTAG)			
TCK	Test Clock	I	
TDI	Test Data Input	I	
TDO	Test Data Output	O	TS2
TMS	Test Mode Select	I	
Power Supplies			
AVDD	Analog Power [1-4]		
AVSS	Analog Ground [1-2]		
DVDD	Digital Power [1-7]		
DVSS	Digital Ground [1-13]		

Output Driver Types

Name	Type	I_{OL} (mA)	I_{OH} (mA)	pF
TS1	Tri-State	4	–1	50
TS2	Tri-State	12	–4	50
TS3	Tri-State	24	–3	120
OD3	Open Drain	24	–3	120

PIN DESCRIPTION: BUS SLAVE MODE

ISA Interface

AEN

Address Enable ***Input***

This signal must be driven LOW when the bus performs an I/O access to the device.

IOCHRDY

I/O Channel Ready ***Output***

When the PCnet-ISA II controller is being accessed, a HIGH on IOCHRDY indicates that valid data exists on the data bus for reads and that data has been latched for writes.

$\overline{\text{IOCS16}}$

I/O Chip Select 16 ***Input/Output***

When an I/O read or write operation is performed, the PCnet-ISA II controller will drive this pin LOW to indicate that the chip supports a 16-bit operation at this address. (If the motherboard does not receive this signal, then the motherboard will convert a 16-bit access to two 8-bit accesses).

The PCnet-ISA II controller follows the IEEE P996 specification that recommends this function be implemented as a pure decode of SA0-9 and AEN, with no dependency on $\overline{\text{IOR}}$, or $\overline{\text{IOW}}$; however, some PC/AT clone systems are not compatible with this approach. For this reason, the PCnet-ISA II controller is recommended to be configured to run 8-bit I/O on all machines. Since data is moved by memory cycles there is virtually no performance loss incurred by running 8-bit I/O and compatibility problems are virtually eliminated. The PCnet-ISA II controller can be configured to run 8-bit-only I/O by clearing Bit 0 in Plug and Play Register F0.

$\overline{\text{IOR}}$

I/O Read ***Input***

To perform an Input/Output Read operation on the device $\overline{\text{IOR}}$ must be asserted. $\overline{\text{IOR}}$ is only valid if the AEN signal is LOW and the external address matches the PCnet-ISA II controller's predefined I/O address location. If valid, $\overline{\text{IOR}}$ indicates that a slave read operation is to be performed.

$\overline{\text{IOW}}$

I/O Write ***Input***

To perform an Input/Output write operation on the device $\overline{\text{IOW}}$ must be asserted. $\overline{\text{IOW}}$ is only valid if AEN signal is LOW and the external address matches the PCnet-ISA II controller's predefined I/O address location. If valid, $\overline{\text{IOW}}$ indicates that a slave write operation is to be performed.

IRQ3, 4, 5, 9, 10, 11, 12, 15

Interrupt Request ***Output***

An attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON or TXSTRT. All status flags have a mask bit which allows for suppression of IRQ assertion. These flags have the following meaning:

BABL	Babble
RCVCCO	Receive Collision Count Overflow
JAB	Jabber
MISS	Missed Frame
MERR	Memory Error
MPCO	Missed Packet Count Overflow
RINT	Receive Interrupt
IDON	Initialization Done
TXSTRT	Transmit Start

$\overline{\text{MEMR}}$

Memory Read ***Input***

$\overline{\text{MEMR}}$ goes LOW to perform a memory read operation.

$\overline{\text{MEMW}}$

Memory Write ***Input***

$\overline{\text{MEMW}}$ goes LOW to perform a memory write operation.

REF

Memory Refresh ***Input***

When $\overline{\text{REF}}$ is asserted, a memory refresh cycle is in progress. During a refresh cycle, $\overline{\text{MEMR}}$ assertion is ignored.

RESET

Reset ***Input***

When RESET is asserted HIGH, the PCnet-ISA II controller performs an internal system reset. RESET must be held for a minimum of 10 XTAL1 periods before being deasserted. While in a reset state, the PCnet-ISA II controller will tristate or deassert all outputs to predefined reset levels. The PCnet-ISA II controller resets itself upon power-up.

SA0-15

System Address Bus ***Input***

This bus carries the address inputs from the system address bus. Address data is stable during command active cycle.

SBHE

System Bus High Enable ***Input***

This signal indicates the HIGH byte of the system data bus is to be used. There is a weak pull-up resistor on this pin. If the PCnet-ISA II controller is installed in an 8-bit only system like the PC/XT, SBHE will always be HIGH and the PCnet-ISA II controller will perform only 8-bit operations. There must be at least one LOW going edge on this signal before the PCnet-ISA II controller will perform 16-bit operations.

SD0-15

System Data Bus ***Input/Output***

This bus is used to transfer data to and from the PCnet-ISA II controller to system resources via the ISA data bus. SD0-15 is driven by the PCnet-ISA II controller when performing slave read operations.

Likewise, the data on SD0-15 is latched by the PCnet-ISA II controller when performing slave write operations.

Board Interface

APCS/IRQ15

Address PROM Chip Select ***Output***

This signal is asserted when the external Address PROM is read. When an I/O read operation is performed on the first 16 bytes in the PCnet-ISA II controller's I/O space, APCS is asserted. The outputs of the external Address PROM drive the PROM Data Bus. The PCnet-ISA II controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus. IOCS16 is not asserted during this cycle.

BPAM

Boot PROM Address Match ***Input***

This pin indicates a Boot PROM access cycle. If no Boot PROM is installed, this pin has a default value of HIGH and thus may be left connected to V_{DD}.

BPCS

Boot PROM Chip Select ***Output***

This signal is asserted when the Boot PROM is read. If BPAM is active and MEMR is active, the BPCS signal will be asserted. The outputs of the external Boot PROM drive the PROM Data Bus. The PCnet-ISA II controller buffers the contents of the PROM data bus and drives them on the System Data Bus. IOCS16 is not asserted during this cycle. If 16-bit cycles are performed, it is the responsibility of external logic to assert MEMCS16 signal.

DXCVR/EAR

Disable Transceiver/ External Address Reject ***Input/Output***

This pin disables the transceiver. The DXCVR output is configured in the initialization sequence. A high level indicates the Twisted Pair Interface is active and the AUI is inactive, or SLEEP mode has been entered. A low level indicates the AUI is active and the Twisted Pair interface is inactive.

If EADI mode is selected, this pin becomes the EAR input.

The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be OR'd with the value on the EAR pin. The EAR pin is defined as REJECT. (See the EADI section for details regarding the function and timing of this signal).

LED0-3

LED Drivers ***Output***

These pins sink 12 mA each for driving LEDs. Their meaning is software configurable (see section *The ISA Bus Configuration Registers*) and they are active LOW.

When EADI mode is selected, the pins named LED1, LED2, and LED3 change in function while LED0 continues to indicate 10BASE-T Link Status. The DXCVR input becomes the EAR input.

LED	EADI Function
1	SF/BD
2	SRD
3	SRDCLK

PRAB0-15

Private Address Bus ***Input/Output***

The Private Address Bus is the address bus used to drive the Address PROM, Remote Boot PROM, and SRAM. PRAB10-15 are required to be buffered by a Bus Buffer with ABOE as its control and SA10-15 as its inputs.

PRDB3-7

Private Data Bus ***Input/Output***

This is the data bus for the static RAM, the Boot PROM, and the Address PROM.

PRDB2/EEDO

Private Data Bus Bit 2/Data Out ***Input/Output***

A multifunction pin which serves as PRDB2 of the private data bus and, when ISACSR3 bit 4 is set, changes to become DATA OUT from the EEPROM.

PRDB1/EEDI

Private Data Bus Bit 1/Data In ***Input/Output***

A multifunction pin which serves as PRDB1 of the private data bus and, when ISACSR3 bit 4 is set, changes to become DATA In to the EEPROM.

PRDB0/EESK

Private Data Bus Bit 0/ Serial Clock ***Input/Output***

A multifunction pin which serves as PRDB0 of the private data bus and, when ISACSR3 bit 4 is set, changes to become Serial Clock to the EEPROM.

SHFBUSY

Shift Busy ***Input/Output***

This pin indicates that a read from the external EEPROM is in progress. It is active only when data is being shifted out of the EEPROM due to a hardware RESET or assertion of the EE_LOAD bit (ISACSR3, bit 14). If this pin is left unconnected or pulled low with a pull-down resistor, an EEPROM checksum error is forced. Normally, this pin should be connected to V_{CC} through a 10K Ω pull-up resistor.

EECS

EEPROM CHIP SELECT ***Output***

This signal is asserted when read or write accesses are being performed to the EEPROM. It is controlled by ISACSR3. It is driven at Reset during EEPROM Read.

$\overline{\text{SLEEP}}$

Sleep ***Input***

When $\overline{\text{SLEEP}}$ input is asserted (active LOW), the PCnet-ISA II controller performs an internal system reset and proceeds into a power savings mode. All outputs will be placed in their normal reset condition. All PCnet-ISA II controller inputs will be ignored except for the $\overline{\text{SLEEP}}$ pin itself. Deassertion of $\overline{\text{SLEEP}}$ results in the device waking up. The system must delay the starting of the network controller by 0.5 seconds to allow internal analog circuits to stabilize.

$\overline{\text{SMA}}$

Slave Mode Architecture ***Input***

This pin must be permanently pulled LOW for operation in the Bus Slave mode. It is sampled after the hardware RESET sequence. In the Bus Slave mode, the PCnet-ISA II can be programmed for Shared Memory access or Programmed I/O access through the PIOSEL bit (ISACSR2, bit 13).

$\overline{\text{SMAM}}$

Shared Memory Address Match ***Input***

When the Shared Memory architecture is selected (ISACSR2, bit 13), this pin is an input that indicates an access to shared memory when asserted. The type of access is decided by $\overline{\text{MEMR}}$ or $\overline{\text{MEMW}}$.

When the Programmed I/O architecture is selected, this pin should be permanently tied HIGH.

$\overline{\text{SROE}}$

Static RAM Output Enable ***Output***

This pin directly controls the external SRAM's OE pin.

$\overline{\text{SRCS}}$/IRQ12

Static RAM Chip Select ***Output***

This pin directly controls the external SRAM's chip select ($\overline{\text{CS}}$) pin when the Flash boot ROM option is selected.

When Flash boot ROM option is not selected, this pin becomes IRQ12.

$\overline{\text{SRWE}}$/$\overline{\text{WE}}$

Static RAM Write Enable/ Write Enable ***Output***

This pin ($\overline{\text{SRWE}}$) directly controls the external SRAM's $\overline{\text{WE}}$ pin when a Flash memory device is not implemented.

When a Flash memory device is implemented, this pin becomes a global write enable ($\overline{\text{WE}}$) pin.

XTAL1

Crystal Connection ***Input***

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. Alternatively, an external 20 MHz CMOS-compatible clock signal can be used to drive this pin. Refer to the section on External Crystal Characteristics for more details.

XTAL2

Crystal Connection ***Output***

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. If an external clock is used, this pin should be left unconnected.

PIN DESCRIPTION: NETWORK INTERFACES

AUI

CI+, CI–

Control Input ***Input***

This is a differential input pair used to detect Collision (Signal Quality Error Signal).

DI+, DI–

Data In ***Input***

This is a differential receive data input pair to the PCnet-ISA II controller.

DO+, DO–

Data Out ***Output***

This is a differential transmit data output pair from the PCnet-ISA II controller.

Twisted Pair Interface

RXD+, RXD–

Receive Data ***Input***

This is the 10BASE-T port differential receive input pair.

TXD+, TXD–

Transmit Data ***Output***

These are the 10BASE-T port differential transmit drivers.

TXP+, TXP–

Transmit Predistortion Control ***Output***

These are 10BASE-T transmit waveform pre-distortion control differential outputs.

PIN DESCRIPTION: IEEE 1149.1 (JTAG) TEST ACCESS PORT

TCK

Test Clock ***Input***

This is the clock input for the boundary scan test mode operation. TCK can operate up to 10 MHz. TCK does not have an internal pull-up resistor and must be connected to a valid TTL level of high or low. TCK must not be left unconnected.

TDI

Test Data Input ***Input***

This is the test data input path to the PCnet-ISA II controller. If left unconnected, this pin has a default value of HIGH.

TDO

Test Data Output ***Output***

This is the test data output path from the PCnet-ISA II controller. TDO is tri-stated when JTAG port is inactive.

TMS

Test Mode Select ***Input***

This is a serial input bit stream used to define the specific boundary scan test to be executed. If left unconnected, this pin has a default value of HIGH.

PIN DESCRIPTION: POWER SUPPLIES

All power pins with a “D” prefix are digital pins connected to the digital circuitry and digital I/O buffers. All power pins with an “A” prefix are analog power pins connected to the analog circuitry. Not all analog pins are quiet and special precaution must be taken when doing board layout. Some analog pins are more noisy than others and must be separated from the other analog pins.

AVDD1–4

Analog Power (4 Pins) ***Power***

Supplies power to analog portions of the PCnet-ISA II controller. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines.

AVSS1–2

Analog Ground (2 Pins) ***Power***

Supplies ground reference to analog portions of PCnet-ISA II controller. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines.

DVDD1–7

Digital Power (7 Pins) ***Power***

Supplies power to digital portions of PCnet-ISA II controller. Four pins are used by Input/Output buffer drivers and two are used by the internal digital circuitry.

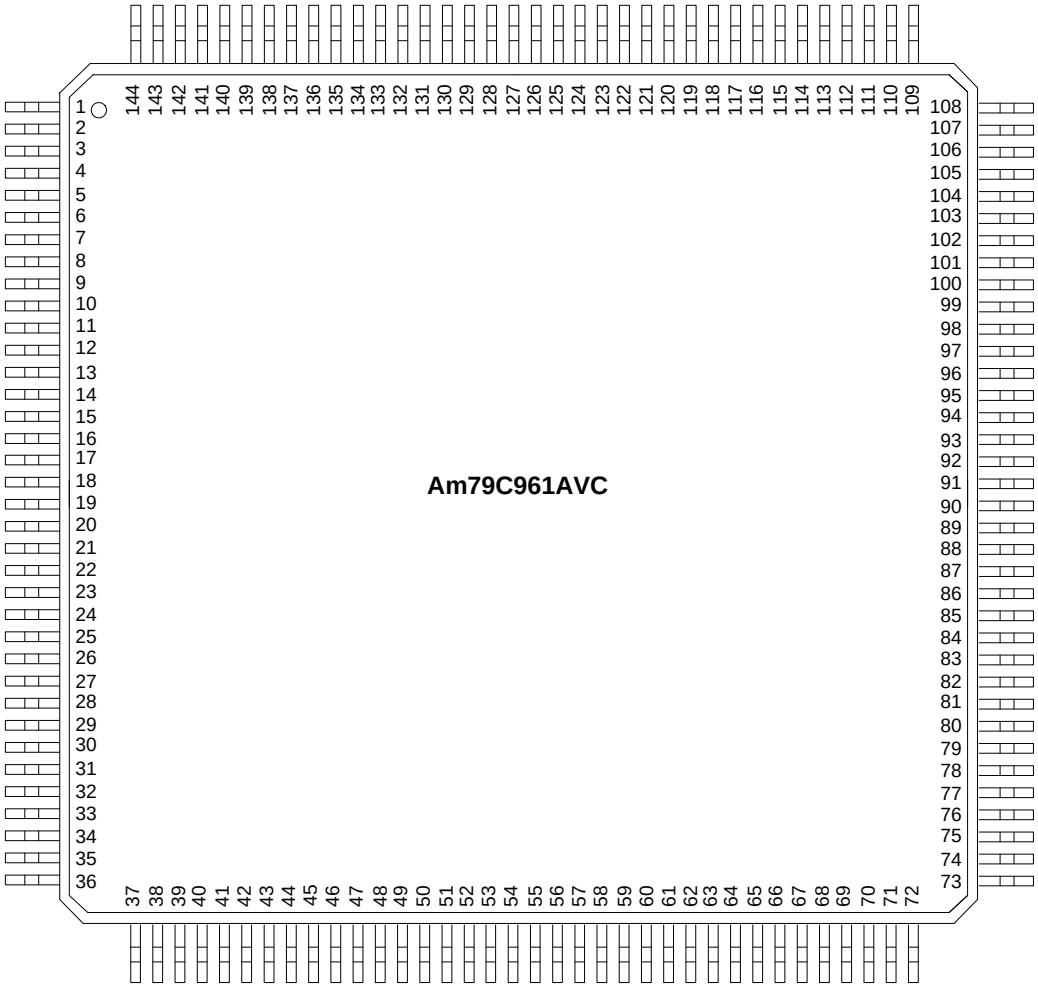
DVSS1–13

Digital Ground (13 Pins) ***Power***

Supplies ground reference to digital portions of PCnet-ISA II controller. Ten pins are used by Input/Output buffer drivers and two are used by the internal digital circuitry.

CONNECTION DIAGRAM

TQFP 144



PIN DESIGNATIONS: BUS MASTER MODE (TQFP 144)

Listed by Pin Number

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	NC	37	NC	73	NC	109	NC
2	DVSS3	38	DVDD4	74	DVDD6	110	AVSS1
3	$\overline{\text{MASTER}}$	39	SA12	75	$\overline{\text{SLEEP}}$	111	DO–
4	DRQ7	40	SA13	76	SD0	112	DO+
5	DRQ6	41	SA14	77	SD8	113	AVDD1
6	DRQ5	42	SA15	78	SD1	114	DI–
7	DVSS10	43	DVSS7	79	SD9	115	DI+
8	$\overline{\text{DACK7}}$	44	SA16	80	DVSS8	116	CI–
9	$\overline{\text{DACK6}}$	45	SA17	81	SD2	117	CI+
10	$\overline{\text{DACK5}}$	46	SA18	82	SD10	118	AVDD2
11	LA17	47	SA19	83	SD3	119	DXCVR/$\overline{\text{EAR}}$
12	LA18	48	AEN	84	SD11	120	$\overline{\text{LED3}}$
13	LA19	49	IOCHRDY	85	DVDD7	121	$\overline{\text{LED2}}$
14	LA20	50	$\overline{\text{MEMW}}$	86	SD4	122	DVSS1
15	DVSS4	51	$\overline{\text{MEMR}}$	87	SD12	123	$\overline{\text{LED1}}$
16	LA21	52	DVSS11	88	SD5	124	$\overline{\text{LED0}}$
17	SA22	53	IRQ15/$\overline{\text{APCS}}$	89	SD13	125	DVDD1
18	SA23	54	IRQ12/Flash$\overline{\text{WE}}$	90	DVSS9	126	PRDB7
19	$\overline{\text{SBHE}}$	55	IRQ11	91	SD6	127	PRDB6
20	DVDD3	56	DVDD5	92	SD14	128	PRDB5
21	SA0	57	IRQ10	93	SD7	129	PRDB4
22	SA1	58	$\overline{\text{IOCS16}}$	94	SD15	130	DVSS2
23	SA2	59	BALE	95	DVSS13	131	PRDB3
24	DVSS5	60	IRQ3	96	RXD–	132	PRDB2/EEDO
25	SA3	61	IRQ4	97	RXD+	133	PRDB1/EEDI
26	SA4	62	IRQ5	98	AVDD4	134	PRDB0/EESK
27	SA5	63	REF	99	TXPD–	135	SHFBUSY
28	SA6	64	DVSS12	100	TXD–	136	$\overline{\text{BPCS}}$
29	SA7	65	DRQ3	101	TXPD+	137	EECS
30	SA8	66	$\overline{\text{DACK3}}$	102	TXD+	138	TDI
31	SA9	67	$\overline{\text{IOR}}$	103	AVDD3	139	TDO
32	DVSS6	68	$\overline{\text{IOW}}$	104	XTAL1	140	TMS
33	SA10	69	IRQ9	105	AVSS2	141	TCK
34	SA11	70	RESET	106	XTAL2	142	DVDD2
35	NC	71	NC	107	NC	143	NC
36	NC	72	NC	108	NC	144	NC

PIN DESIGNATIONS: BUS MASTER MODE (TQFP 144)

Listed by Pin Name

Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
AEN	48	DVSS3	2	NC	37	SA3	25
AVDD1	113	DVSS4	15	NC	71	SA4	26
AVDD2	118	DVSS5	24	NC	72	SA5	27
AVDD3	103	DVSS6	32	NC	73	SA6	28
AVDD4	98	DVSS7	43	NC	107	SA7	29
AVSS1	110	DVSS8	80	NC	108	SA8	30
AVSS2	105	DVSS9	90	NC	109	SA9	31
BALE	59	DXCVR/$\overline{\text{EAR}}$	119	NC	143	$\overline{\text{SBHE}}$	19
$\overline{\text{BPCS}}$	136	EECS	137	NC	144	SD0	76
CI+	117	IOCHRDY	49	PRDB0/EESK	134	SD1	78
CI–	116	$\overline{\text{IOCS16}}$	58	PRDB1/EEDI	133	SD10	82
$\overline{\text{DACK3}}$	66	$\overline{\text{IOR}}$	67	PRDB2/EEDO	132	SD11	84
$\overline{\text{DACK5}}$	10	$\overline{\text{IOW}}$	68	PRDB3	131	SD12	87
$\overline{\text{DACK6}}$	9	IRQ10	57	PRDB4	129	SD13	89
$\overline{\text{DACK7}}$	8	IRQ11	55	PRDB5	128	SD14	92
DI+	115	IRQ12/Flash$\overline{\text{WE}}$	54	PRDB6	127	SD15	94
DI–	114	IRQ15/$\overline{\text{APCS}}$	53	PRDB7	126	SD2	81
DO+	112	IRQ3	60	REF	63	SD3	83
DO–	111	IRQ4	61	RESET	70	SD4	86
DRQ3	65	IRQ5	62	RXD+	97	SD5	88
DRQ5	6	IRQ9	69	RXD–	96	SD6	91
DRQ6	5	LA17	11	SA0	21	SD7	93
DRQ7	4	LA18	12	SA1	22	SD8	77
DVDD1	125	LA19	13	SA10	33	SD9	79
DVDD2	142	LA20	14	SA11	34	SHFBUSY	135
DVDD3	20	LA21	16	SA12	39	$\overline{\text{SLEEP}}$	75
DVDD4	38	$\overline{\text{LED0}}$	124	SA13	40	TCK	141
DVDD5	56	$\overline{\text{LED1}}$	123	SA14	41	TDI	138
DVDD6	74	$\overline{\text{LED2}}$	121	SA15	42	TDO	139
DVDD7	85	$\overline{\text{LED3}}$	120	SA16	44	TMS	140
DVSS1	122	$\overline{\text{MASTER}}$	3	SA17	45	TXD+	102
DVSS10	7	$\overline{\text{MEMR}}$	51	SA18	46	TXD–	100
DVSS11	52	$\overline{\text{MEMW}}$	50	SA19	47	TXPD+	101
DVSS12	64	NC	1	SA2	23	TXPD–	99
DVSS13	95	NC	35	SA22	17	XTAL1	104
DVSS2	130	NC	36	SA23	18	XTAL2	106

PIN DESIGNATIONS: BUS SLAVE (PIO AND SHARED MEMORY) MODES (TQFP 144)
Listed by Pin Number

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	NC	37	NC	73	NC	109	NC
2	DVSS3	38	DVDD4	74	DVDD6	110	AVSS1
3	$\overline{\text{SMA}}$	39	PRAB12	75	$\overline{\text{SLEEP}}$	111	DO-
4	SA0	40	PRAB13	76	SD0	112	DO+
5	SA1	41	PRAB14	77	SD8	113	AVDD1
6	SA2	42	PRAB15	78	SD1	114	DI-
7	DVSS10	43	DVSS7	79	SD9	115	DI+
8	SA3	44	SA13	80	DVSS8	116	CI-
9	SA4	45	SA14	81	SD2	117	CI+
10	SA5	46	SA15	82	SD10	118	AVDD2
11	SA6	47	$\overline{\text{SRWE}}$	83	SD3	119	DXCVR/$\overline{\text{EAR}}$
12	SA7	48	AEN	84	SD11	120	$\overline{\text{LED3}}$
13	SA8	49	IOCHRDY	85	DVDD7	121	$\overline{\text{LED2}}$
14	SA9	50	$\overline{\text{MEMW}}$	86	SD4	122	DVSS1
15	DVSS4	51	$\overline{\text{MEMR}}$	87	SD12	123	$\overline{\text{LED1}}$
16	SA10	52	DVSS11	88	SD5	124	$\overline{\text{LED0}}$
17	SA11	53	IRQ15	89	SD13	125	DVDD1
18	SA12	54	IRQ12	90	DVSS9	126	PRDB7
19	$\overline{\text{SBHE}}$	55	IRQ11	91	SD6	127	PRDB6
20	DVDD3	56	DVDD5	92	SD14	128	PRDB5
21	PRAB0	57	IRQ10	93	SD7	129	PRDB4
22	PRAB1	58	$\overline{\text{IOCS16}}$	94	SD15	130	DVSS2
23	PRAB2	59	$\overline{\text{BPAM}}$	95	DVSS13	131	PRDB3
24	DVSS5	60	IRQ3	96	RXD-	132	PRDB2/ EEDO
25	PRAB3	61	IRQ4	97	RXD+	133	PRDB1/EEDI
26	PRAB4	62	IRQ5	98	AVDD4	134	PRDB0/EESK
27	PRAB5	63	$\overline{\text{REF}}$	99	TXPD-	135	SHFBUSY
28	PRAB6	64	DVSS12	100	TXD-	136	$\overline{\text{BPCS}}$
29	PRAB7	65	$\overline{\text{SROE}}$	101	TXPD+	137	EECS
30	PRAB8	66	$\overline{\text{SMAM}}$	102	TXD+	138	TDI
31	PRAB9	67	$\overline{\text{IOR}}$	103	AVDD3	139	TDO
32	DVSS6	68	$\overline{\text{IOW}}$	104	XTAL1	140	TMS
33	PRAB10	69	IRQ9	105	AVSS2	141	TCK
34	PRAB11	70	RESET	106	XTAL2	142	DVDD2
35	NC	71	PCMCIA_MODE	107	NC	143	NC
36	NC	72	NC	108	NC	144	NC

PIN DESIGNATIONS: BUS SLAVE (PIO AND SHARED MEMORY) MODES (TQFP 144)

Listed by Pin Name

Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
AEN	48	EECS	137	PRAB13	40	SA7	12
AVDD1	113	IOCHRDY	49	PRAB14	41	SA8	13
AVDD2	118	$\overline{\text{IOCS16}}$	58	PRAB15	42	SA9	14
AVDD3	103	$\overline{\text{IOR}}$	67	PRAB2	23	$\overline{\text{SBHE}}$	19
AVDD4	98	$\overline{\text{IOW}}$	68	PRAB3	25	SD0	76
AVSS1	110	IRQ10	57	PRAB4	26	SD1	78
AVSS2	105	IRQ11	55	PRAB5	27	SD10	82
$\overline{\text{BPAM}}$	59	IRQ12	54	PRAB6	28	SD11	84
$\overline{\text{BPCS}}$	136	IRQ15	53	PRAB7	29	SD12	87
CI+	117	IRQ3	60	PRAB8	30	SD13	89
CI–	116	IRQ4	61	PRAB9	31	SD14	92
DI+	115	IRQ5	62	PRDB0/EESK	134	SD15	94
DI–	114	IRQ9	69	PRDB1/EEDI	133	SD2	81
DO+	112	$\overline{\text{LED0}}$	124	PRDB2/EEDO	132	SD3	83
DO–	111	$\overline{\text{LED1}}$	123	PRDB3	131	SD4	86
DVDD1	125	$\overline{\text{LED2}}$	121	PRDB4	129	SD5	88
DVDD2	142	$\overline{\text{LED3}}$	120	PRDB5	128	SD6	91
DVDD3	20	$\overline{\text{MEMR}}$	51	PRDB6	127	SD7	93
DVDD4	38	$\overline{\text{MEMW}}$	50	PRDB7	126	SD8	77
DVDD5	56	NC	1	$\overline{\text{REF}}$	63	SD9	79
DVDD6	74	NC	35	RESET	70	SHFBUSY	135
DVDD7	85	NC	36	RXD+	97	$\overline{\text{SLEEP}}$	75
DVSS1	122	NC	37	RXD–	96	$\overline{\text{SMAM}}$	66
DVSS10	7	NC	72	SA0	4	$\overline{\text{SMA}}$	3
DVSS11	52	NC	73	SA1	5	$\overline{\text{SROE}}$	65
DVSS12	64	NC	107	SA10	16	$\overline{\text{SRWE}}$	47
DVSS13	95	NC	108	SA11	17	TCK	141
DVSS2	130	NC	109	SA12	18	TDI	138
DVSS3	2	NC	143	SA13	44	TDO	139
DVSS4	15	NC	144	SA14	45	TMS	140
DVSS5	24	PCMCIA_MODE	71	SA15	46	TXD+	102
DVSS6	32	PRAB0	21	SA2	6	TXD–	100
DVSS7	43	PRAB1	22	SA3	8	TXPD+	101
DVSS8	80	PRAB10	33	SA4	9	TXPD–	99
DVSS9	90	PRAB11	34	SA5	10	XTAL1	104
DXCVR/$\overline{\text{EAR}}$	119	PRAB12	39	SA6	11	XTAL2	106

FUNCTIONAL DESCRIPTION

The PCnet-ISA II controller is a highly integrated system solution for the PC-AT ISA architecture. It provides a Full Duplex Ethernet controller, AUI port, and 10BASE-T transceiver. The PCnet-ISA II controller can be directly interfaced to an ISA system bus. The PCnet-ISA II controller contains an ISA bus interface unit, DMA Buffer Management Unit, 802.3 Media Access Control function, separate 136-byte transmit and 128-byte receive FIFOs, IEEE defined Attachment Unit Interface (AUI), and Twisted-Pair Transceiver Media Attachment Unit. In addition, a Sleep function has been incorporated which provides low standby current for power sensitive applications.

The PCnet-ISA II controller is register compatible with the LANCE (Am7990) Ethernet controller and PCnet-ISA (Am79C960). The DMA Buffer Management Unit supports the LANCE descriptor software model and the PCnet-ISA II controller is software compatible with the Novell NE2100 and NE1500T add-in cards.

External remote boot PROMs and Ethernet physical address PROMs are supported. The location of the I/O registers, Ethernet address PROM, and the boot PROM are determined by the programming of the registers internal to PCnet-ISA II. These registers are loaded at RESET from the EEPROM, if an EEPROM is utilized.

Normally, the Ethernet physical address will be stored in the EEPROM with the other configuration data. This reduces the parts count, board space requirements, and power consumption. The option to use a standard parallel 8 bit PROM is provided to manufactures who are concerned about the non-volatile nature of EEPROMs.

The PCnet-ISA II controller's bus master architecture brings to system manufacturers (adapter card and motherboard makers alike) something they have not been able to enjoy with other architectures—a low-cost system solution that provides the lowest parts count and highest performance. As a bus-mastering device, costly and power-hungry external SRAMs are not needed for packet buffering. This results in lower system cost due to fewer components, less real-estate and less power. The PCnet-ISA II controller's advanced bus mastering architecture also provides high data throughput and low CPU utilization for even better performance.

To offer greater flexibility, the PCnet-ISA II controller has a Bus Slave mode to meet varying application needs. The bus slave mode utilizes a local SRAM memory to store the descriptors and buffers that are located in system memory when in Bus Master mode. The SRAM can be slave accessed on the ISA bus through memory cycles in Shared Memory mode or I/O cycles in Programmed I/O mode. The Shared Memory and Programmed I/O architectures offer maximum compatibility with low-end machines, such as PC/XTs that do not support bus mastering, and very high end machines which require local packet buffering for increased system latency.

The network interface provides an Attachment Unit Interface and Twisted-Pair Transceiver functions. Only one interface is active at any particular time. The AUI allows for connection via isolation transformer to 10BASE5 and 10BASE2, thick and thin based coaxial cables. The Twisted-Pair Transceiver interface allows for connection of unshielded twisted-pair cables as specified by the Section 14 supplement to IEEE 802.3 Standard (Type 10BASE-T).

Important Note About The EEPROM Byte Map

The user is cautioned that while the Am79C961A (PCnet-ISA II) and its associated EEPROM are pin compatible to their predecessors the Am79C961 (PCnet-ISA$^+$) and its associated EEPROM, the byte map structure in each of the EEPROMs are different from each other.

The EEPROM byte map structure used for the Am79C961A PCnet-ISA II has the addition of "MISC Config 2, ISACSR9" at word location 10Hex. The EEPROM byte map structure used for the Am79C961 PCnet-ISA+ does not have this.

Therefore, should the user intend to replace the PCnet-ISA$^+$ with the PCnet-ISA II, care **MUST** be taken to reprogram the EEPROM to reflect the new byte map structure needed and used by the PCnet-ISA II. For additional information, refer to the section in this data sheet under *EEPROM* and the Am79C961 PCnet-ISA$^+$ data sheet (PID #18183) under the sections entitled *EEPROM* and *Serial EEPROM Byte Map*.

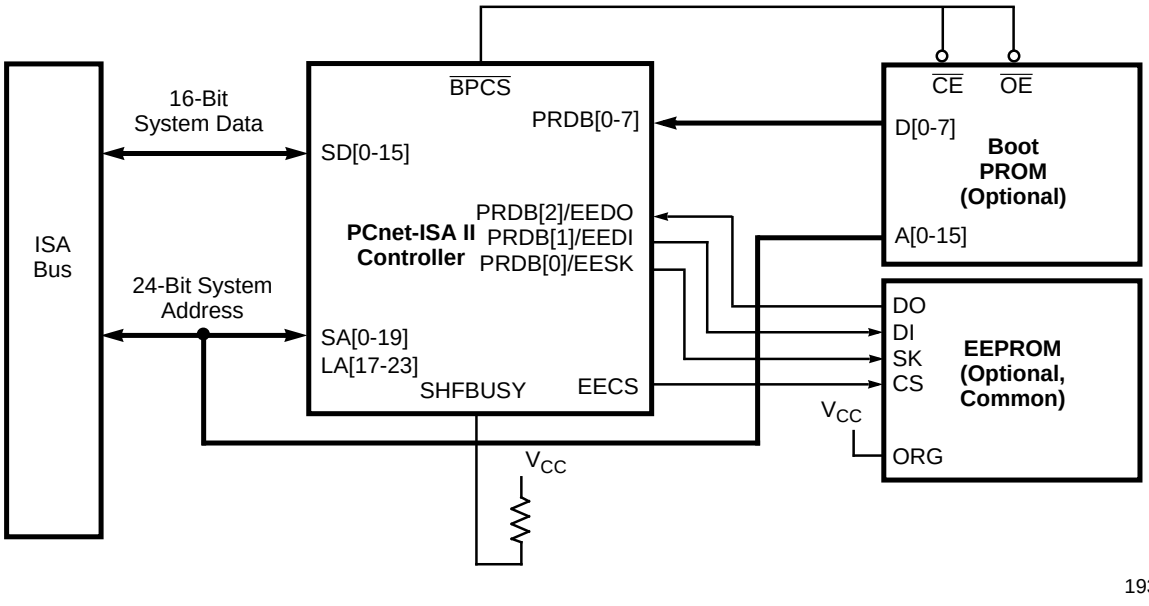
Bus Master Mode

System Interface

The PCnet-ISA II controller has two fundamental operating modes, Bus Master and Bus Slave. Within the Bus Slave mode, the PCnet-ISA II can be programmed for a Shared Memory or Programmed I/O architecture. The selection of either the Bus Master mode or the Bus Slave mode must be done through hard wiring; it is not software configurable. When in the Bus Slave mode, the selection of the Shared Memory or Programmed I/O architecture is done through software with the PIOSEL bit (ISACSR2, bit 13).

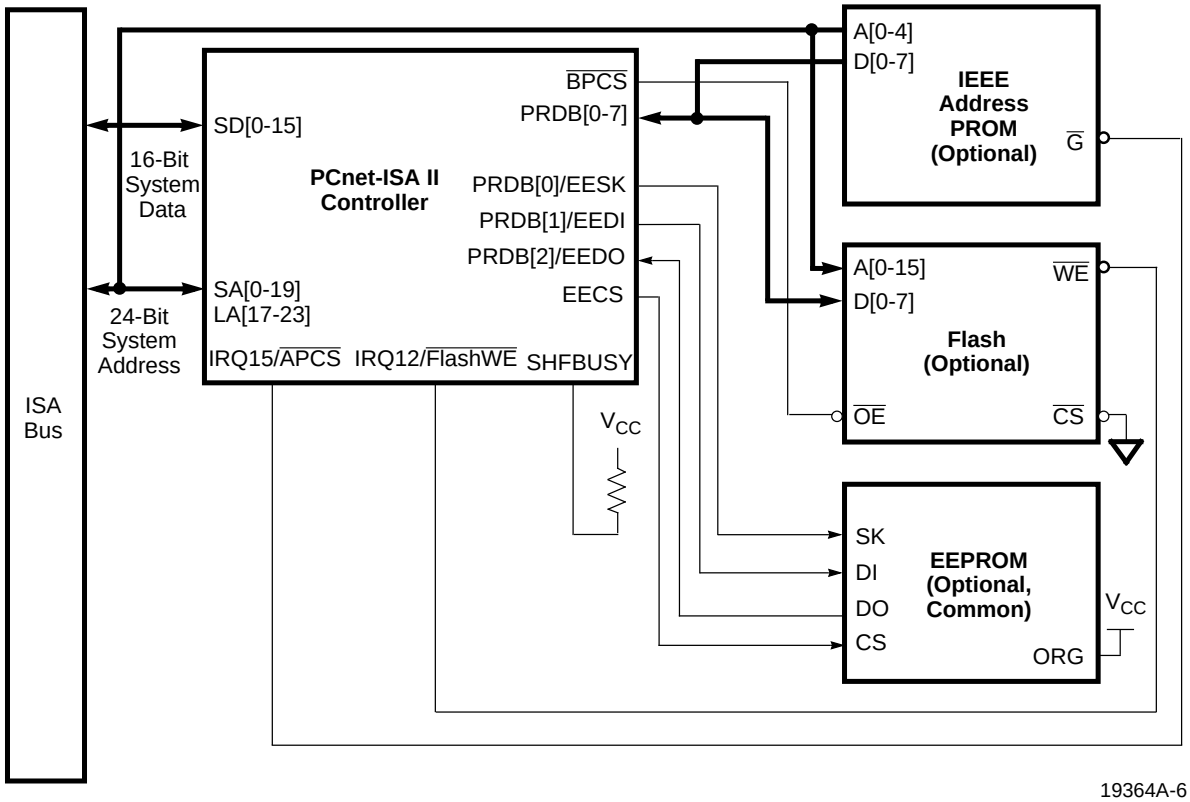
The optional Boot PROM is in memory address space and is expected to be 8–64K. On-chip address comparators control device selection is based on the value in the EEPROM.

The address PROM, board configuration registers, and the Ethernet controller occupy 24 bytes of I/O space and can be located on 16 different starting addresses.



19364A-5

Bus Master Block Diagram Plug and Play Compatible



19364A-6

Bus Master Block Diagram Plug and Play Compatible with Flash and parallel Address PROM Support

Bus Slave Mode

System Interface

The Bus Slave mode is the other fundamental operating mode available on the PCnet-ISA II controller. Within the Bus Slave mode, the PCnet-ISA II can be programmed for a Shared Memory or Programmed I/O architecture. In the Bus Slave mode the PCnet-ISA II controller uses the same descriptor and buffer architecture as in the Bus Master mode, but these data structures are stored in a static RAM controlled by the PCnet-ISA II controller. When operating with the Shared Memory architecture, the local SRAM is visible as a memory resource on the PC which can be accessed through memory cycles on the ISA bus interface. When operating with the Programmed I/O architecture, the local SRAM is accessible through I/O cycles on the ISA bus. Specifically, the SRAM is accessible using the RAP and IDP I/O ports to access the ISACSR0 and ISACSR1 registers, which serve as the SRAM Data port and SRAM Address Pointer port, respectively.

In the Bus Slave mode, the PCnet-ISA II registers and optional Ethernet physical address PROM look the same and are accessed in the same way as in the Bus Master mode.

The Boot PROM is selected by an external device which drives the Boot PROM Address Match ($\overline{\text{BPAM}}$) input to the PCnet-ISA II controller. The PCnet-ISA II controller can perform two 8-bit accesses from the 8-bit Boot PROM and present 16-bits of data to accommodate 16 bit read accesses on the ISA bus.

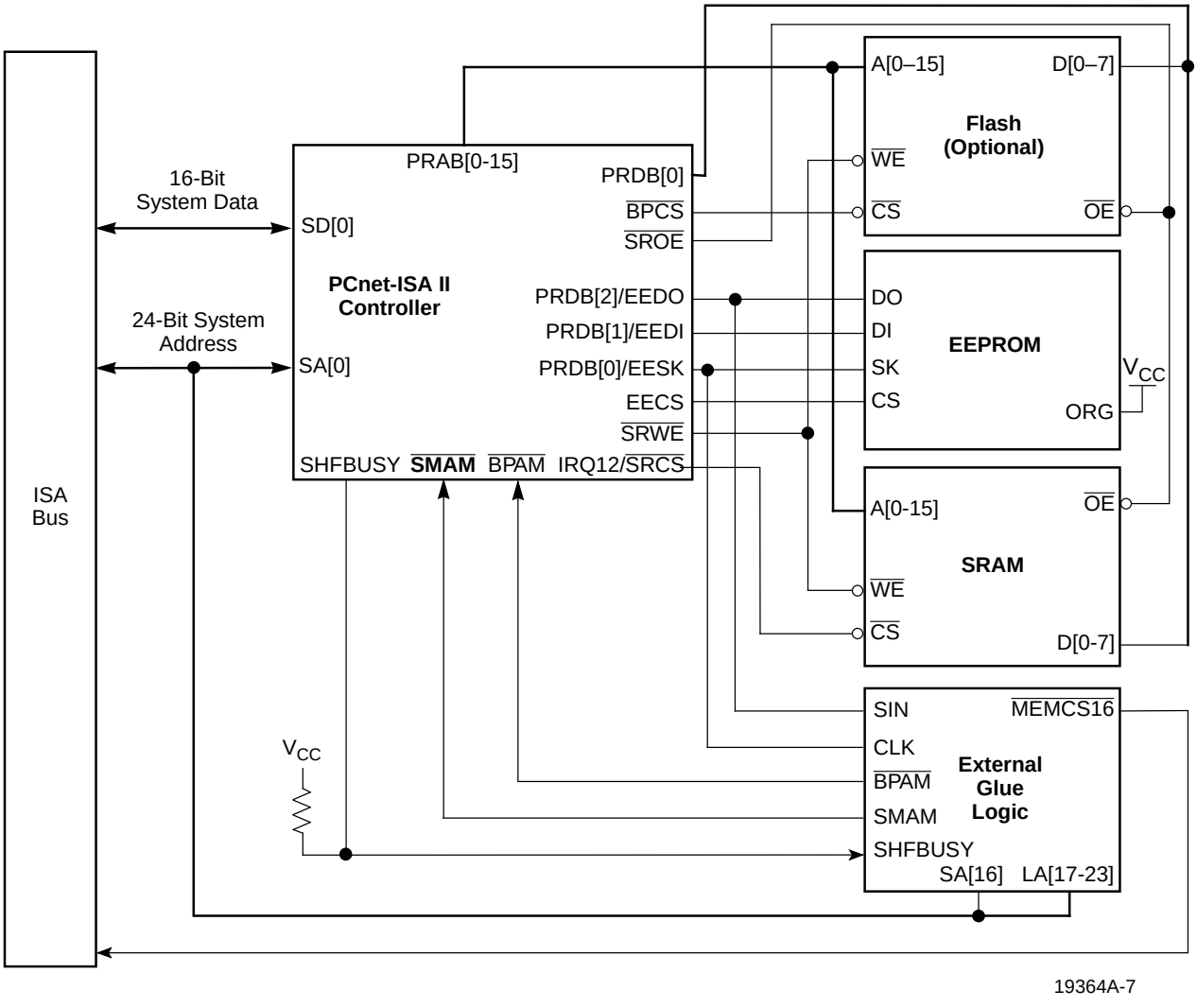
When using the Shared Memory architecture mode, access to the local SRAM works the same way as access to the Boot PROM, with an external device generating the Shared Memory Address Match ($\overline{\text{SMAM}}$) signal and the PCnet-ISA II controller performing the SRAM read or write and the 8/16 bit data conversion.

External logic must also drive $\overline{\text{MEMCS16}}$ appropriately for the 128Kbyte segment decoded from the LA[23:17] signals.

The Programmed I/O architecture mode uses the RAP and IDP ports to allow access to the local SRAM hence, external address decoding is not necessary and the $\overline{\text{SMAM}}$ pin is not used in Programmed I/O architecture mode ($\overline{\text{SMAM}}$ should be tied HIGH in the Programmed I/O architecture mode). Similar to the Shared Memory architecture mode, in the Programmed I/O architecture mode, 8/16 bit conversion occurs when 16 bit reads and writes are performed on the SRAM Data Port (ISACSR1).

Converting the local SRAM accesses from 8-bit cycles to 16-bit cycles allows use of the much faster 16-bit cycle timing while cutting the number of bus cycles in half. This raises performance to more than 400% of what could be achieved with 8-bit cycles. When the Shared Memory architecture mode is used, converting boot PROM accesses to 16-bit cycles allows the two memory resources to be in the same 128 Kbyte block of memory without a clash between two devices with different data widths.

The PCnet-ISA II prefetches data from the SRAM to allow fast, minimum wait-state read accesses of consecutive SRAM addresses. In both the Shared Memory architecture and the Programmed I/O architecture, prefetch data is read from a speculated address that assumes that successive reads in time will be from adjacent ascending addresses in the SRAM. At the beginning of each SRAM read cycle, the PCnet-ISA II determines whether the prefetched data can be assumed to be valid. If the prefetched data can be assumed to be valid, it is driven onto the ISA bus without inserting any wait states. If the prefetched data cannot be assumed to be valid, the PCnet-ISA II will insert wait states into the ISA bus read cycle until the correct word is read from the SRAM.



Note:
SMAM shown only for Shared Memory architecture designs. SMAM should be tied HIGH on the PCnet-ISA II for Programmed I/O architecture designs.

Bus Slave Block Diagram
Plug and Play Compatible with Flash Memory Support

PLUG AND PLAY

Plug and Play is a standardized method of configuring jumperless adapter cards in a system. Plug and Play is a Microsoft standard and is based on a central software configuration program, either in the operating system or elsewhere, which is responsible for configuring all Plug and Play cards in a system. Plug and Play is fully supported by the PCnet-ISA II ethernet controller.

For a copy of the Microsoft Plug and Play specification contact Microsoft Inc. This specification should be referenced in addition to PCnet-ISA II Technical Reference Manual and this data sheet.

Operation

If the PCnet-ISA II ethernet controller is used to boot off the network, the device will come up active at RESET, otherwise it will come up inactive. Information stored in the serial EEPROM is used to identify the card and to describe the system resources required by the card, such as I/O space, Memory space, IRQs and DMA channels. This information is stored in a standardized Read Only format. Operation of the Plug and Play system is shown as follows:

- Isolate the Plug and Play card
- Read the cards resource data
- Identify the card
- Configure its resources

The Plug and Play mode of operation allows the following benefits to the end user.

- Eliminates all jumpers or dip switches from the adapter card
- Ease of use is greatly enhanced
- Allows the ability to uniquely address identical cards in a system, without conflict
- Allows the software configuration program or OS to read out the system resource requirements required by the card
- Defines a mechanism to set or modify the current configuration of each card
- Maintain backward compatibility with other ISA bus adapters

Auto-Configuration Ports

Three 8 bit I/O ports are used by the Plug and Play configuration software on each Plug and Play device to communicate with the Plug and Play registers. The ports are listed in the table below. The software configuration space is defined as a set of 8 bit registers. These registers are used by the Plug and Play software configuration to issue commands, access the resource information, check status, and configure the PCnet-ISA II controller hardware.

Port Name	Location	Type
ADDRESS	0X279 (Printer Status Port)	Write-only
WRITE-DATA	0xA79 (Printer status port + 0x0800)	Write-only
READ-DATA	Relocatable in range 0x0203-0x03FF	Read-only

The address and Write_DATA ports are located at fixed, predefined I/O addresses. The Write_Data port is located at an alias of the Address port. All three auto-configuration ports use a 12-bit ISA address decode.

The READ_DATA port is relocatable within the range 0x203–0x3FF by a command written to the WRITE_DATA port.

ADDRESS PORT

The internal Plug and Play registers are accessed by writing the address to the ADDRESS PORT and then either reading the READ_DATA PORT or writing to the WRITE_DATA PORT. Once the ADDRESS PORT has been written, any number of reads or writes can occur without having to rewrite the ADDRESS PORT.

The ADDRESS PORT is also the address to which the initiation key is written to, which is described later.

WRITE_DATA PORT

The WRITE_DATA PORT is the address to which all writes to the internal Plug and Play registers occur. The destination of the data written to the WRITE_DATA PORT is determined by the last value written to the ADDRESS PORT.

READ_DATA PORT

The READ_DATA PORT is used to read information from the internal Plug and Play registers. The register to be read is determined by the last value of the ADDRESS PORT.

The I/O address of the READ_DATA PORT is set by writing the chosen I/O location to Plug and Play Register 0. The isolation protocol can determine that the address chosen is free from conflict with other devices I/O ports.

Initiation Key

The PCnet-ISA II controller is disabled at reset when operating in Plug and Play mode. It will not respond to any memory or I/O accesses, nor will the PCnet-ISA II controller drive any interrupts or DMA channels.

The initiation key places the PCnet-ISA II device into the configuration mode. This is done by writing a predefined pattern to the ADDRESS PORT. If the proper sequence of I/O writes are detected by the PCnet-ISA II device, the Plug and Play auto-configuration ports are enabled. This pattern must be sequential, i.e., any

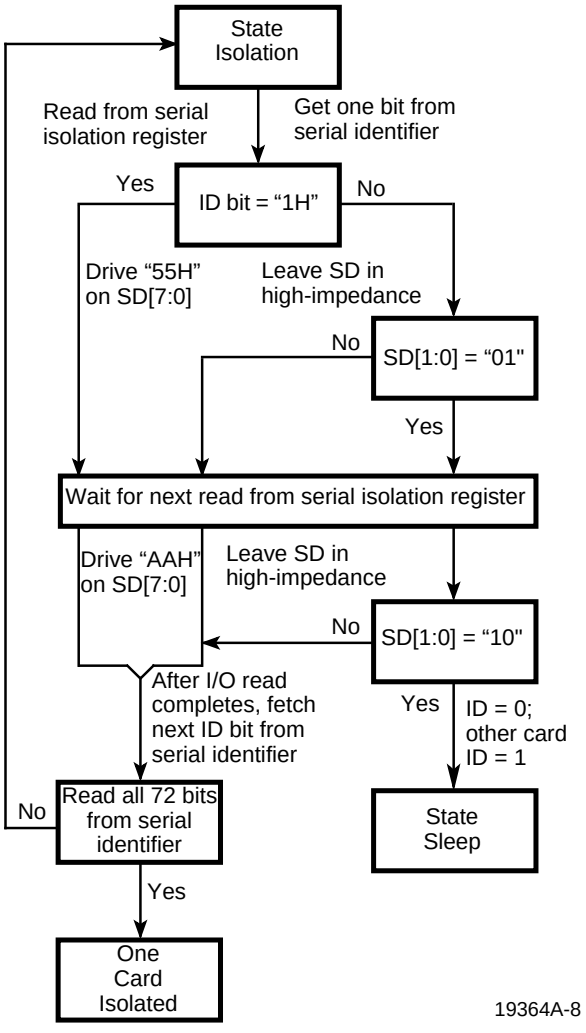
other I/O access to this I/O port will reset the state machine which is checking the pattern. Interrupts should be disabled during this time to eliminate any extraneous I/O cycles.

The exact sequence for the initiation key is listed below in hexadecimal.

```
6A, B5, DA, ED, F6, FB, 7D, BE
DF, 6F, 37, 1B, 0D, 86, C3, 61
B0, 58, 2C, 16, 8B, 45, A2, D1
E8, 74, 3A, 9D, CE, E7, 73, 39
```

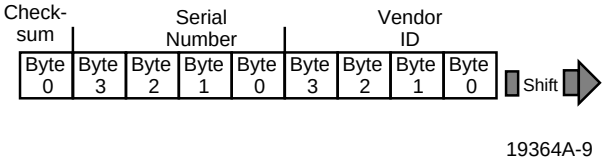
Isolation Protocol

A simple algorithm is used to isolate each Plug and Play card. This algorithm uses the signals on the ISA bus and requires lock-step operation between the Plug and Play hardware and the isolation software.



Plug and Play ISA Card Isolation Algorithm

The key element of this mechanism is that each card contains a unique number, referred to as the serial identifier for the rest of the discussion. The serial identifier is a 72-bit unique, non-zero, number composed of two, 32-bit fields and an 8-bit checksum. The first 32-bit field is a vendor identifier. The other 32 bits can be any value, for example, a serial number, part of a LAN address, or a static number, as long as there will never be two cards in a single system with the same 64 bit number. The serial identifier is accessed bit-serially by the isolation logic and is used to differentiate the cards.



Shifting of Serial Identifier

The shift order for all Plug and Play serial isolation and resource data is defined as bit[0], bit[1], and so on through bit[7].

Hardware Protocol

The isolation protocol can be invoked by the Plug and Play software at any time. The initiation key, described earlier, puts all cards into configuration mode. The hardware on each card expects 72 pairs of I/O read accesses to the READ_DATA port. The card's response to these reads depends on the value of each bit of the serial identifier which is being examined one bit at a time in the sequence shown above.

If the current bit of the serial identifier is a "1", then the card will drive the data bus to 0x55 to complete the first I/O read cycle. If the bit is "0", then the card puts its data bus driver into high impedance. All cards in high impedance will check the data bus during the I/O read cycle to sense if another card is driving D[1:0] to "01". During the second I/O read, the card(s) that drove the 0x55, will now drive a 0xAA. All high impedance cards will check the data bus to sense if another card is driving D[1:0] to "10". Between pairs of Reads, the software should wait at least 30 µs.

If a high impedance card sensed another card driving the data bus with the appropriate data during both cycles, then that card ceases to participate in the current iteration of card isolation. Such cards, which lose out, will participate in future iterations of the isolation protocol.

Note: *During each read cycle, the Plug and Play hardware drives the entire 8-bit databus, but only checks the lower 2 bits.*

If a card was driving the bus or if the card was in high impedance and did not sense another card driving the bus, then it should prepare for the next pair of I/O reads. The card shifts the serial identifier by one bit and uses the shifted bit to decide its response. The above sequence is repeated for the entire 72-bit serial identifier.

At the end of this process, one card remains. This card is assigned a handle referred to as the *Card Select Number* (CSN) that will be used later to select the card. Cards which have been assigned a CSN will not participate in subsequent iterations of the isolation protocol. Cards must be assigned a CSN before they will respond to the other commands defined in the specification.

It should be noted that the protocol permits the 8-bit checksum to be stored in non-volatile memory on the card or generated by the on-card logic in real-time. The same LFSR algorithm described in the initiation key section of the Plug and Play specification is used in the checksum generation.

Software Protocol

The Plug and Play software sends the initiation key to all Plug and Play cards to place them into configuration mode. The software is then ready to perform the isolation protocol.

The Plug and Play software generates 72 pairs of I/O read cycles from the READ_DATA port. The software checks the data returned from each pair of I/O reads for the 0x55 and 0xAA driven by the hardware. If both 0x55 and 0xAA are read back, then the software assumes that the hardware had a “1" bit in that position. All other results are assumed to be a “0.”

During the first 64 bits, software generates a checksum using the received data. The checksum is compared with the checksum read back in the last 8 bits of the sequence.

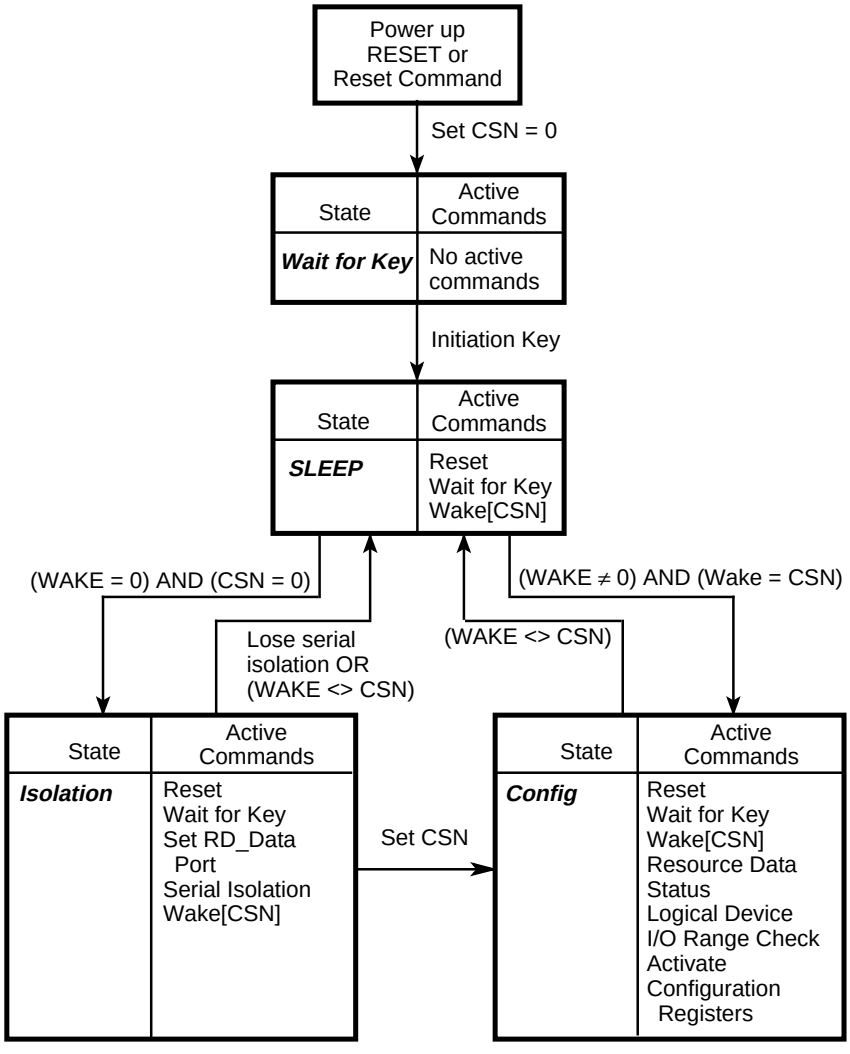
There are two other special considerations for the software protocol. During an iteration, it is possible that the 0x55 and 0xAA combination is never detected. It is also possible that the checksum does not match If either of these cases occur on the first iteration, it must be assumed that the READ_DATA port is in conflict. If a conflict is detected, then the READ_DATA port is relocated. The above process is repeated until a non-conflicting location for the READ_DATA port is found. The entire range between 0x203 and 0x3FF is available, however in practice it is expected that only a few locations will be tried before software determines that no Plug and Play cards are present.

During subsequent iterations, the occurrence of either of these two special cases should be interpreted as the absence of any further Plug and Play cards (i.e. the last card was found in the previous iteration). This terminates the isolation protocol.

Note: *The software must delay 1 ms prior to starting the first pair of isolation reads, and must wait 250 µsec between each subsequent pair of isolation reads. This delay gives the ISA card time to access information from possibly very slow storage devices.*

Plug and Play Card Control Registers

The state transitions and card control commands for the PCnet-ISA II controller are shown in the following figure.



Notes

1. *CSN = Card Select Number*
2. *RESET or the Reset command causes a state transition from the current state to Wait for Key and sets all CSNs to zero.*
3. *The Wait for Key command causes a state transition from the current state to Wait for Key.*

19364A-10

Plug and Play ISA Card State Transitions

Plug and Play Registers

The PCnet-ISA II controller supports all of the defined Plug and Play card control registers. Refer to the tables on the following pages for detailed information.

Plug and Play Standard Registers

Name	Address Port Value	Definition
Set RD_DATA Port	0x00	Writing to this location modifies the address of the port used for reading from the Plug and Play ISA cards. Bits**[7:0]** become I/O read port address bits **[9:2]**. Reads from this register are ignored. I/O Address bits 11:10 should = 00, and 1:0 = 11.
Serial Isolation	0x01	A read to this register causes a Plug and Play card in the ***Isolation*** state to compare one bit of the board's ID. This process is fully described above. This register is read only.
Config Control	0x02	Bit[0] - Reset all logical devices and restore configuration registers to their power-up values. Bit[1] - Return to the ***Wait for Key*** state Bit[2] - Reset CSN to 0 A write to bit[0] of this register performs a reset function on all logical devices. This resets the contents of configuration registers to their default state. All card's logical devices enter their default state and the CSN is preserved. A write to bit[1] of this register causes all cards to enter the ***Wait for Key*** state but all CSNs are preserved and logical devices are not affected. A write to bit[2] of this register causes all cards to reset their CSN to zero. This register is write-only. The values are not sticky, that is, hardware will automatically clear them and there is no need for software to clear the bits.
Wake[CSN]	0x03	A write to this port will cause all cards that have a CSN that matches the write data[7:0] to go from the ***Sleep*** state to either the ***Isolation*** state if the write data for this command is zero or the ***Config*** state if the write data is not zero. This register is write-only. Writing to this register resets the EEPROM pointer to the beginning of the Plug and Play Data Structure.
Resource Data	0x04	A read from this address reads the next byte of resource information. The Status register must be polled until bit[0] is set before this register may be read. This register is read-only.
Status	0x05	Bit[0] when set indicates it is okay to read the next data byte from the Resource Data register. This register is read-only.
Card Select Number	0x06	A write to this port sets a card's CSN. The CSN is a value uniquely assigned to each ISA card after the serial identification process so that each card may be individually selected during a Wake [CSN] command. This register is read/write.
Logical Device Number	0x07	Selects the current logical device. This register is read only. The PCnet-ISA II controller has only 1 logical device, and this register contains a value of 0x00

Plug and Play Logical Device Configuration Registers

The PCnet-ISA II controller supports a subset of the defined Plug and Play logical device control registers. The reason for only supporting a subset of the registers is that the PCnet-ISA II controller does not require as many system resources as Plug and Play allows. For instance, Memory Descriptor 2 is not used, as the PCnet-ISA II controller only requires two memory descriptors, one for the Boot PROM/Flash, and one for the SRAM in Shared Memory Mode.

Plug and Play Logical Device Control Registers

Name	Address Port Value	Definition
Activate	0x30	For each logical device there is one activate register that controls whether or not the logical device is active on the ISA bus. Bit[0], if set, activates the logical device. Bits[7:1] are reserved and must be zero. This is a read/write register. Before a logical device is activated, I/O range check must be disabled.
I/O Range Check	0x31	This register is used to perform a conflict check on the I/O port range programmed for use by a logical device. Bit[7:2] Reserved Bit 1[1] Enable I/O Range check, if set then I/O Range Check is enabled. I/O range check is only valid when the logical device is inactive. Bit[0], if set, forces the logical device to respond to I/O reads of the logical device's assigned I/O range with a 0x55 when I/O range check is in operation. If clear, the logical device drives 0xAA. This register is read/write.

Memory Space Configuration

Name	Register Index	Definition
Memory base address bits[23:16] descriptor 0	0x40	Read/write value indicating the selected memory base address bits[23:16] for memory descriptor 0. This is the Boot Prom Space.
Memory base address bits [23:16] descriptor 0	0x41	Read/write value indicating the selected memory base address bits[15:08] for memory descriptor 0.
Memory control	0x42	**Bit[1]** specifies 8/16-bit control. The encoding **relates** to memory control (bits[4:3]) of the information field in the memory descriptor. Bit[0], =0, indicates the next field is used as a range length for decode (implies range length and base alignment of memory descriptor are equal). Bit[0] is read-only.
Memory upper limit address; bits [23:16] or range length; bits [15:08] for descriptor 0	0x43	Read/write value indicating the selected memory high address bits[23:16] for memory descriptor 0. If bit[0] of memory control is 0, this is the range length. If bit[0] of memory control is 1, this is considered invalid.
Memory upper limit bits [15:08] or range length; bits [15:08] for descriptor 0	0x44	Read/write value indicating the selected memory high address bits[15:08] for memory descriptor 0, either a memory address or a range length as described above.
Memory descriptor 1	0x48-0x4C	Memory descriptor 1. This is the SRAM Space for Shared Memory.

I/O Space ConfigurationI/O Interrupt Configuration

Name	Register Index	Definition
I/O port base address bits[15:08] descriptor 0	0x60	Read/write value indicating the selected I/O lower limit address bits[15:08] for I/O descriptor 0. If a logical device indicates it only uses 10 bit encoding, then bits[15:10] do not need to be supported.

Name	Register Index	Definition
I/O port base address bits[07:00] descriptor 0	0x61	Read/write value indicating the selected I/O lower limit address bits[07:00] for I/O descriptor 0.

Name	Register Index	Definition
Interrupt request level select 0	0x70	Read/write value indicating selected interrupt level. Bits[3:0] select which interrupt level used for Interrupt 0. One selects IRQL 1, fifteen selects IRQL fifteen. IRQL 0 is not a valid interrupt selection and represents no interrupt selection.
Interrupt request type select 0	0x71	Read/write value indicating which type of interrupt is used for the Request Level selected above. Bit[1] : Level, 1 = high, 0 = low Bit[0] : Type, 1 = level, 0 = edge The PCnet-ISA II controller only supports Edge High and Level Low Interrupts.

DMA Channel Configuration

Name	Register Index	Definition
DMA channel select 0	0x74	Read/write value indicating selected DMA channels. Bits[2:0] select which DMA channel is in use for DMA 0. Zero selects DMA channel 0, seven selects DMA channel 7. DMA channel 4, the cascade channel is used to indicate no DMA channel is active.
DMA channel select 1	0x75	Read only with a value of 0x04.

DETAILED FUNCTIONS

EEPROM

Interface

The EEPROM supported by the PCnet-ISA II controller is an industry standard 93C56 2-Kbit EEPROM device which uses a 4-wire interface. This device directly interfaces to the PCnet-ISA II controller through a 4-wire interface which uses 3 of the private data bus pins for Data In, Data Out, and Serial Clock. The Chip Select pin is a dedicated pin from the PCnet-ISA II controller.

***Note**: All data stored in the EEPROM is stored in bit-reversal format. Each word (16 bits) must be written into the EEPROM with bit 15 swapped with bit 0, bit 14 swapped with bit 1, etc.*

This is a 2-Kbit device organized as 128 x 16 bit words. A map of the device as used in the PCnet-ISA II controller is below. The information stored in the EEPROM is as follows:

IEEE address 6 bytes
Reserved10 bytes
EISA ID4 bytes
ISACSRs14 bytes
Plug and Play Defaults19 bytes
8-Bit Checksum1 byte
External Shift Chain2 bytes
Plug and Play Config Info192 bytes

Important Note About The EEPROM Byte Map

The user is cautioned that while the Am79C961A (PCnet-ISA II) and its associated EEPROM are pin compatible to their predecessors the Am79C961 (PCnet-ISA$^+$) and its associated EEPROM, the byte map structure in each of the EEPROMs are different from each other.

The EEPROM byte map structure used for the Am79C961A PCnet-ISA II has the addition of “MISC Config 2, ISACSR9" at word location 10Hex. The EEPROM byte map structure used for the Am79C961 PCnet-ISA+ does not have this.

Therefore, should the user intend to replace the PCnet-ISA$^+$ with the PCnet-ISA II, care **MUST** be taken to reprogram the EEPROM to reflect the new byte map structure needed and used by the PCnet-ISA II. For additional information, refer to the Am79C961 PCnet-ISA$^+$ data sheet (PID #18183) under the sections entitled *EEPROM* and *Serial EEPROM Byte Map*.

Basic EEPROM Byte Map

The following is a byte map of the XXC56 series of EEPROMs used by the PCnet-ISA II Ethernet Controller. This byte map is for the case where a non-PCnet Family compatible software driver is implemented.

			Word Location	
IEEE Address (0h) (Bytes 0 – 5)	Byte 1	Byte 0	0	
	Byte 3	Byte 2	1	
	Byte 5	Byte 4	2	
	Byte 7	Byte 6	3	
	Byte 9	Byte 8	4	
	Byte 11	Byte 10	5	
	Byte 13	Byte 12	6	
	Byte 15	Byte 14	7	
(8h)	EISA Byte 1	EISA Byte 0	8	
EISA Config Reg.	EISA Byte 3	EISA Byte 2	9	
(Ah)	MSRDA, ISACSR0		A	
	MSWRA, ISACSR1		B	
	MISC Config 1, ISACSR2		C	
Internal Registers	LED1 Config, ISACSR5		D	
	LED2 Config, ISACSR6		E	
	LED3 Config, ISACSR7		F	
	MISC Config 2, ISACSR9		10	
(11h)	PnP 0x61	PnP 0x60	11	I/O Ports
	Pnp 0x71	PnP 0x70	12	Interrupts
	Unused	PnP 0x74	13	DMA Channels
Plug and Play Reg.	PnP 0x41	PnP 0x40	14	ROM Memory
	PnP 0x43	PnP 0x42	15	
	Unused	PnP 0x44	16	
	PnP 0x49	PnP 0x48	17	RAM Memory
	PnP 0x4B	PnP 0x4A	18	
	Unused	PnP 0x4C	19	
(1Ah)	8–Bit Checksum	PnP 0xF0		Vendor Byte
(1Bh)	External Shift Chain		1B	
(1Ch)	Unused Locations		1C ⋮ 1F	
(20h)	Plug and Play Starting Location		20	

Note:

Checksum is calculated on words 0 through 0x1Bh (first 56 bytes).

AMD Device Driver Compatible EEPROM Byte Map

The following is a byte map of the XXC56 series of EEPROMs used by the PCnet-ISA II Ethernet Controller. This byte map is for the case where a PCnet Family compatible software driver is implemented.

(This byte map is an application reference for use in developing AMD software devices.)

	Word Location			
	0	Byte 1	Byte 0	IEEE Address (Bytes 0–5)
	1	Byte 3	Byte 2	
	2	Byte 5	Byte 4	
	3	Reserved	Reserved	
	4	HWID (01H)	Reserved	
	5	User Space 1		
	6	16-Bit Checksum 1		
	7	ASCII W (0 x 57H)	ASCII W (0 x 57H)	
	8	EISA Byte 1	EISA Byte 0	
EISA Config Reg.	9	EISA Byte 3	EISA Byte 2	
	A	MSRDA, ISACSR0		
	B	MSWRA, ISACSR1		
	C	MISC Config, ISACR2		
Internal Registers	D	LED1 Config, ISACSR5		
	E	LED2 Config, ISACSR6		
	F	LED3 Config, ISACSR7		
	10	MISC Config 2, ISACSR9		
	11	PnP 0x61	PnP 0x60	I/O Ports
	12	Pnp 0x71	PnP 0x70	Interrupts
	13	Unused	PnP 0x74	DMA Channels
Plug and Play Reg.	14	PnP 0x41	PnP 0x40	ROM Memory
	15	PnP 0x43	PnP 0x42	
	16	Unused	PnP 0x44	
	17	PnP 0x49	PnP 0x48	RAM Memory
	18	PnP 0x4B	PnP 0x4A	
	19	Unused	PnP 0x4C	
	1A	8-Bit Checksum	PnP 0xF0	Vendor Byte
	1B	External Shift Chain		
	1C ⋮ 1F	Unused Locations		
See Appendix C	20	Plug and Play Starting Location		See Appendix C

Note:

Checksum 1 is calculated on words 0 through 5 plus word 7.

Checksum 2 is calculated on words 0 through 0x1Bh (first 56 bytes).

Plug and Play Register Map

The following chart and its bit descriptions show the internal configuration registers associated with the Plug and Play operation. These registers control the configuration of the PCnet-ISA II controller.

Plug and Play Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	READ_DATA							
0x01	SERIAL ISOLATION							
0x02	0	0	0	0	0	RST CSN	WAIT KEY	RST ALL
0x03	WAKE [CSN]							
0x04	RESOURCE_DATA							
0x05	0	0	0	0	0	0	0	READ STATUS
0x06	CSN							
0x07	LOGICAL DEVICE NUMBER							
0x30	0	0	0	0	0	0	0	ACTIVATE
0x31	0	0	0	0	0	0	IORNG	IORNG

READ_DATA	Address of Plug and Play READ_DATA Port.
SERIAL_ISOLATION	Used in the Serial Isolation process.
RST_CSN	Resets CSN register to zero.
WAIT_KEY	Resets Wait for Key State.
RST_ALL	Resets all logical devices.
WAKE [CSN]	Will wake up if write data matches CSN Register.
READ_STATUS	Read Status of RESOURCE DATA.
RESOURCE_DATA	Next pending byte read from EEPROM.
CSN	Plug and Play CSN Value.
ACTIVATE	Indicates that the PCnet-ISA II device should be activated.
IORNG	Bits used to enable the I/O Range Check Command.

The following chart and its bit descriptions show the internal command registers associated with the Plug and Play operation. These registers control the PCnet-ISA II controller Plug and Play operation.

Plug and Play Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x60	0	0	0	0	0	0	1	IOAM3
0x61	IOAM2	IOAM1	IOAM0	0	0	0	0	0
0x70	0	0	0	0	IRQ3	IRQ2	IRQ1	IRQ0
0x71	0	0	0	0	0	0	IRQ_LVL	IRQ_TYPE
0x74	0	0	0	0	0	DMA2	DMA1	DMA0
0x40	0	0	0	0	1	1	0	BPAM3
0x41	BPAM2	BPAM1	BPAM0	0	0	0	0	0
0x42	0	0	0	0	0	0	BP_16B	0
0x43	1	1	1	1	1	1	1	BPSZ3
0x44	BPSZ2	BPSZ1	BPSZ0	0	0	0	0	0
0x48	0	0	0	0	1	1	SRAM4	SRAM3
0x49	SRAM2	SRAM1	SRAM0	0	0	0	0	0
0x4A	0	0	0	0	0	0	SR16B	0
0x4B	1	1	1	1	1	1	1	SRSZ3
0x4c	SRSZ2	SRSZ1	SRSZ0	0	0	0	0	0
0xF0	0	LGCY_EN	DXCVRP	FL_SEL	BP_CS	APROM_EN	AEN_CS	IO_MODE

PCnet–ISA II's Legacy Bit Feature Description

The current PCnet-ISA II chip is designed such that it always responds to Plug and Play configuration software. There are situations where this response to the Plug and Play software is undesirable. An example of this is when a fixed configuration is required, or when the only possible resource available for the PCnet-ISA II conflicts with a present but not used resource such as IRQ, or when the chip is used in a system with a buggy PnP BIOS.

To function in the situations above, a new feature has been added to the PCnet-ISA II chip. This new feature makes the chip ignore the PnP software's special initiation key sequence (6A). This will effectively turn the chip into the "Legacy" mode operation, where it will be visible in the I/O space, and only special setup programs will be able to reconfigure it. In case the EEPROM is missing, empty, or corrupted, the chip will still recognize AMD's special initiation key sequence (6B).

To enable this feature, a one has to be written into the LGCY_EN bit, which is bit 6 of the Plug and Play register 0xF0. A preferred method would be set this bit in the Vendor Byte (PnP 0xF0) field of the EEPROM located in word offset 0x1A.

Plug & Play Register Locations Detailed Description (Refer to the Plug & Play Register Map above).

IOAM[3:0] I/O Address Match to bits [8:5] of SA bus (PnP 0x60–0x61). Controls the base address of PCnet-ISA II. The IOAM will be written with a value from the EEPROM.

IOAM[3:0]				Base Address (Hex)
0	0	0	0	200
0	0	0	1	220
0	0	1	0	240
0	0	1	1	260
0	1	0	0	280
0	1	0	1	2A0
0	1	1	0	2C0
0	1	1	1	2E0
1	0	0	0	300
1	0	0	1	320
1	0	1	0	340
1	0	1	1	360
1	1	0	0	380
1	1	0	1	3A0
1	1	1	0	3C0
1	1	1	1	3E0

IRQ[3:0] IRQ selection on the ISA bus (PnP 0x70). Controls which interrupt will be asserted. ISA Edge sensitive or EISA level mode is controlled by IRQ_TYPE bit in PnP 0x71. Default is ISA Edge Sensitive. The IRQ signals will not be driven unless PnP activate register bit is set.

IRQ[3:0]				ISA IRQ Pin
0	0	1	1	IRQ3 (Default)
0	1	0	0	IRQ4
0	1	0	1	IRQ5
1	0	0	1	IRQ9
1	0	1	0	IRQ10
1	0	1	1	IRQ11
1	1	0	1	IRQ12
1	1	1	0	IRQ15

IRQ Type IRQ Type(PnP 0x71). Indicates the type of interrupt setting; Level is 1, Edge is 0.

IRQ_LVL IRQ Level (PnP 0x71). A read-only register bit that indicates the type of setting, active high or low. Always complement of IRQ_TYPE.

DMA[2:0] DMA Channel Select (PnP 0x74). Controls the DRQ and DMA selection of PCnet-ISA II. The DMA[2:0] register will be written with a value from the EEPROM. {For Bus Master Mode Only} The DRQ signals will not be driven unless Plug and Play activate register bit is set.

DMA[2:0]			DMA Channel (DRQ/DACK Pair)
0	1	1	Channel 3
1	0	1	Channel 5
1	1	0	Channel 6
1	1	1	Channel 7
1	0	0	No DMA Channel

BPAM[3:0] Boot PROM Address Match to bits [16:13] of SA bus (PnP 0x40–0x41). Selects the location where the Boot PROM Address match decode is started. The BPAM will be written with a value from the EEPROM.

BPAM[3:0]				Address Location (Hex)	Size Supported (K bytes)
0	0	0	0	C0000	8, 16, 32, 64
0	0	0	1	C2000	8
0	0	1	0	C4000	8, 16
0	0	1	1	C6000	8
0	1	0	0	C8000	8, 16, 32
0	1	0	1	CA000	8
0	1	1	0	CC000	8, 16
0	1	1	1	CE000	8
1	0	0	0	D0000	8, 16, 32, 64
1	0	0	1	D2000	8
1	0	1	0	D4000	8, 16
1	0	1	1	D6000	8
1	1	0	0	D8000	8, 16, 32
1	1	0	1	DA000	8
1	1	1	0	DC000	8, 16
1	1	1	1	DE000	8

BP_16B Boot PROM 16-bit access (PnP 0x42). Is asserted if Boot PROM cycles should respond as an 16-bit device. In Bus Master mode, all boot PROM cycles will only be 8 bits in width.

BPSZ[3:0] Boot PROM Size (PnP 0x43–0x44). Selects the size of the boot PROM selected.

BPSZ[3:0]				Boot PROM Size
0	x	x	x	No Boot PROM Selected
1	1	1	1	8 K
1	1	1	0	16 K
1	1	0	0	32 K
1	0	0	0	64 K

SRAM[4:0] Static RAM Address Match to bits [17:13] of SA bus (PnP 0x48-0x49). Selects the starting location of the Shared Memory when using the

Shared Memory architecture mode. The SRAM[2:0] bits are used for performing address decoding on the SA[15:13] address bits as shown in the table below. SRAM[4] and SRAM[3] must reflect the external address match logic for SA[17] and SA[16], respectively. The SRAM[4:0] bits are ignored when in the Bus Master mode or in the Programmed I/O Architecture mode.

SRAM[2:0]			SA[15:13]			SRAM Size (K bytes)
0	0	0	0	0	0	8, 16, 32, 64
0	0	1	0	0	1	8
0	1	0	0	1	0	8, 16
0	1	1	0	1	1	8
1	0	0	1	0	0	8, 16, 32
1	0	1	1	0	1	8
1	1	0	1	1	0	8, 16
1	1	1	1	1	1	8

SR_16B — Static RAM 16-bit access (PnP 0x4A). If asserted, the PCnet-ISA II will respond to SRAM cycles as a 16-bit device. This bit should be set if external logic is designed to assert the $\overline{\text{MEMCS16}}$ signal when accesses to the shared memory are decoded. This bit is ignored when in the Bus Master mode or in the Programmed I/O Architecture mode.

SRSZ[3:0] — Static RAM size (PnP 0x4B-0x4C). Selects the size of the static RAM. The SRSZ[3:0] bits are ignored when in the Bus Master mode or in the Programmed I/O Architecture mode.

SRSZ[3:0]				Shared Memory Size
0	x	x	x	No Static RAM Selected
1	1	1	1	8 K
1	1	1	0	16 K
1	1	0	0	32 K
1	0	0	0	64 K

Vendor Defined Byte (PnP 0xF0)

LGCY_EN — Legacy mode enable. When written with a one, the PCnet-ISA II will not respond to the Plug and Play initiation key sequence (6A) but will respond to the AMD key sequence (6B). Therefore, it cannot be reconfigured by the Plug and Play software. When set to zero (default), the PCnet-ISA II will respond to the 6A key sequence if the EEPROM read was successful, otherwise it will respond to the 6B key sequence.

DXCVRP — DXCVR Polarity. The DXCVRP bit sets the polarity of the DXCVR pin. When DXCVRP is cleared (default), the DXCVR pin is driven HIGH when the Twisted Pair port is active or SLEEP mode has been entered and driven LOW when the AUI port is active. When DXCVRP is set, the DXCVR pin is driven LOW when the Twisted Pair port is active or SLEEP mode has been entered and driven HIGH when the AUI port is active.

The DXCVRP should generally be left cleared when the PCnet-ISA II is being used with an external DC-DC converter that has an active low enable pin. The DXCVRP should generally be set when the PCnet-ISA II is being used with an external DC-DC converter that has an active high enable pin.

IO_MODE — I/O Mode. When set to one, the internal selection will respond as a 16-bit port, (i.e. drive $\overline{\text{IOCS16}}$ pin). When IO_MODE is set to zero, (Default), the internal I/O selection will respond as an 8-bit port.

AEN_CS — External Decode Logic for I/O Registers. When written with a one, the PCnet-ISA II will use the AEN pin as I/O chip select bar, to allow for external decode logic for the upper address bit of SA [9:5]. The purpose of this pin is to allow I/O locations, not supported with the IOAM[3:0], selection, to be defined outside the range 0x200–0x3F7. When set to a zero, (Default), I/O Selection will use IOAM[3:0].

APROM_EN — External Parallel IEEE Address PROM. When set, the IRQ15 pin is reconfigured to be an Address Chip Select low, similar to $\overline{\text{APCS}}$ pin in the existing PCnet-ISA (Am79C960) device. The purpose of this bit is to allow for both a serial EEPROM and parallel PROM to coexist. When APROM_EN is set, the IEEE address located in the serial EEPROM will be ignored and parallel access will occur over the PRDB bus. When APROM_EN is cleared,

default state, the IEEE address will be read in from the serial device and written to an internal RAM. When the I/O space of the IEEE PROM is selected, PCnet-ISA II, will access the contents of this RAM for I/O read cycles. I/O write cycles will be ignored.

BP_CS — Boot PROM Chip Select. When BP_CS is set to one, BALE will act as an external chip select (active low) above bit 15 of the address bus. BALE = 0, will select the boot PROM when $\overline{MEMR}$ is asserted low if the BP_CS bit is set and BPAM[2:0] match SA[15:13] and BPSZ[3:0] matches the selected size. When BP_CS is set to zero. BALE will act as the normal address latch strobe to capture the upper address bits for memory access to the boot PROM. BP_CS is by default low. The primary purpose of this bit is to allow non-ISA bus applications to support larger Boot PROMS or non-standard Boot PROM/Flash locations.

FL_SEL — Flash Memory Device Selected. When set, the Boot PROM is replaced with an external Flash memory device. In Bus Master Mode, $\overline{BPCS}$ is replaced with $\overline{Flash\_OE}$. IRQ12 becomes $\overline{Flash\_WE}$. The Flash's CS pin is grounded. In shared memory mode, $\overline{BPCS}$ is replaced with $\overline{Flash\_CS}$. IRQ12 becomes $\overline{Static\_RAM\_CS}$ pin. The $\overline{SROE}$ and $\overline{SRWE}$ signals are connected to both the SRAM and Flash memory devices. FL_SEL is cleared by a reset, which is the default.

Checksum Failure

After RESET, the PCnet-ISA II controller begins reading the EEPROM and storing the information in registers inside PCnet-ISA II controller. PCnet-ISA II controller does a checksum on word locations 0-1Bh inclusive and if the byte checksum = FFh, then the data read from the EEPROM is considered good. If the checksum is not equal to FFh, then the PCnet-ISA II controller enters what is called software relocatable mode.

In software relocatable mode, the device functions the same as in Plug and Play mode, except that it does not respond to the same initiation key as Plug and Play supports. Instead, a different key is used to bring PCnet-ISA II controller out of the Wait For Key state. This key is as follows:

```
6B, 35, 9A, CD, E6, F3, 79, BC
5E, AF, 57, 2B, 15, 8A, C5, E2
F1, F8, 7C, 3E, 9F, 4F, 27, 13
09, 84, 42, A1, D0, 68, 34, 1A
```

Use Without EEPROM

In some designs, especially PC motherboard applications, it may be desirable to eliminate the EEPROM altogether. This would save money, space, and power consumption.

The operation of this mode is similar to when the PCnet-ISA II controller encounters a checksum error, except that to enter this mode the SHFBUSY pin is left unconnected. The device will enter software relocatable mode, and the BIOS on the motherboard can wake up the device, configure it, load the IEEE address (possibly stored in Flash ROM) into the PCnet-ISA II controller, and activate the device.

External Scan Chain

The External Scan Chain is a set of bits stored in the EEPROM which are not used in the PCnet-ISA II controller but which can be used with external hardware to allow jumperless configuration of external devices.

After RESET, the PCnet-ISA II controller begins reading the EEPROM and storing the information in registers inside the PCnet-ISA II controller. SHFBUSY is held high during the read of the EEPROM. If external circuitry is added, such as a shift register, which is clocked from SCLK and is attached to DO from the EEPROM, data read out of the EEPROM will be shifted into the shift register. After reading the EEPROM to the end of the External Shift Chain, and if there is a correct checksum, SHFBUSY will go low. This will be used to latch the information from the EEPROM into the shift register. If the checksum is invalid, SHFBUSY will not go low, indicating that the EEPROM may be bad.

Flash PROM

Use

Instead of using a PROM or EPROM for the Boot PROM, it may be desirable to use a Flash or EEPROM type of device for storing the Boot code. This would allow for in-system updates and changes to the information in the Boot ROM without opening up the PC. It may also be desirable to store statistics or drivers in the Flash device.

Interface

To use a Flash-type device with the PCnet-ISA II controller, Flash Select is set in register 0F0h of the

Plug and Play registers. Flash Select is cleared by RESET (default).

In bus master mode, $\overline{\text{BPCS}}$ becomes $\overline{\text{Flash\_OE}}$ and IRQ12 becomes $\overline{\text{Flash\_WE}}$. The Flash ROM devices $\overline{\text{CS}}$ pin is connected to ground.

In shared memory mode, $\overline{\text{BPCS}}$ becomes $\overline{\text{Flash\_CS}}$ and IRQ12 becomes the static RAM Chip Select, and the $\overline{\text{SROE}}$ and $\overline{\text{SRWE}}$ signals are connected to both the SRAM and Flash devices.

Optional IEEE Address PROM

Normally, the Ethernet physical address will be stored in the EEPROM with the other configuration data. This reduces the parts count, board space requirements, and power consumption. The option to use a standard parallel 8 bit PROM is provided to manufacturers who are concerned about the non-volatile nature of EEPROMs.

To use a 8 bit parallel PROM to store the IEEE address data instead of storing it in the EEPROM, the APROM_EN bit is set in the Plug and Play registers by the EEPROM upon RESET. IRQ15 is redefined by the setting of this bit to be $\overline{\text{APCS}}$, or ADDRESS PROM CHIP SELECT. This pin is connected to an external 8 bit PROM, such as a 27LS19. The address pins of the PROM are connected to the lower address pins of the ISA bus, and the data lines are connected to the private data bus.

In this mode, any accesses to the IEEE address will be passed to the external PROM and the data will be passed through the PCnet-ISA II controller to the system data bus.

EISA Configuration Registers

The PCnet-ISA II controller has support for the 4-byte EISA Configuration Registers. These are used in EISA systems to identify the card and load the appropriate configuration file for that card. This feature is enabled using bit 10 of ISACSR2. When set to 1, the EISA Configuration registers will be enabled and will be read at I/O location 0xC80–0xC83. The contents of these 4 registers are stored in the EEPROM and are automatically read in at RESET.

Bus Interface Unit (BIU)

The bus interface unit is a mixture of a 20 MHz state machine and asynchronous logic. It handles two types of accesses; accesses where the PCnet-ISA II controller is a slave and accesses where the PCnet-ISA II controller is the Current Master.

In slave mode, signals like $\overline{\text{IOCS16}}$ are asserted and deasserted as soon as the appropriate inputs are received. IOCHRDY is asynchronously driven LOW if the PCnet-ISA II controller needs a wait state. It is released synchronously when the PCnet-ISA II controller is ready.

When the PCnet-ISA II controller is the Current Master, all the signals it generates are synchronous to the on-chip 20 MHz clock.

DMA Transfers

The BIU will initiate DMA transfers according to the type of operation being performed. There are three primary types of DMA transfers:

1. Initialization Block DMA Transfers

During initialization, the PCnet-ISA II transfers 12 words from the initialization block in memory to internal registers. These 12 words are transferred through different bus mastership period sequences, depending on whether the TIMER bit (CSR4, bit 13) is set and, if TIMER is set, on the value in the Bus Activity Timer register (CSR82).

If the TIMER bit is reset (default), the 12 words are always transferred during three separate bus mastership periods. During each bus mastership period, four words (8 bytes) will be read from contiguous memory addresses.

If the TIMER bit is set, the 12 words may be transferred using anywhere from 1 to 3 bus mastership periods, depending on the value of the Bus Activity Timer register (CSR82). During each bus mastership period, a minimum of four words (8 bytes) will be read from contiguous memory addresses. If the TIMER bit is set and the value in the Bus Activity Timer register allows it, 8 or all 12 words of the initialization block are read during a single bus mastership period.

2. Descriptor DMA Transfers

Descriptor DMA transfers are performed to read or write to transmit or receive descriptors. All transmit and receive descriptor READ accesses require 3 word reads (TMD1, TMD0, then TMD2 for transmit descriptors and RMD1, RMD0, then RMD2 for receive descriptors). Transmit and receive descriptor WRITE accesses to unchained descriptors or the last descriptor in a chain (ENP set) require 2 word writes (TMD1 then TMD3 for transmit and RMD1 then RMD3 for receive). Transmit and receive descriptor WRITE accesses to chained descriptors that do not have ENP set require 1 word write (TMD1 for transmit and RMD1 for receive). During descriptor write accesses, only the bytes which need to be written are written, as controlled by the SA0 and $\overline{\text{SBHE}}$ pins.

If the TIMER bit is reset (default), all accesses during a single bus mastership period will be either all read or all write and will be to only one descriptor. Hence, when the TIMER bit is reset, the bus mastership periods for descriptor accesses are always either 3, 2, or 1 cycles

long, depending on which descriptor operation is being performed.

If the TIMER bit is set, the 3, 2, or 1 cycles required in a descriptor access may be performed as a part of a bus mastership period in which any combination of descriptor reads and writes and buffer reads and writes are performed. When the TIMER bit is set, the Bus Activity Timer (CSR82) and the bus access requirements of the PCnet-ISA II govern the operations performed during a single bus mastership period.

3. FIFO DMA Transfers

FIFO DMA transfers occur when the PCnet-ISA II microcode determines that transfers to and/or from the FIFOs are required. Once the PCnet-ISA II BIU has been granted bus mastership, it will perform a series of consecutive transfer cycles before relinquishing the bus.

When the Bus Activity Timer is disabled by clearing the TIMER (CSR4, bit 13) bit, all FIFO DMA transfers within a bus mastership period will be either read or write cycles, and all transfers will be to adjacent, ascending addresses. When the Bus Activity Timer is enabled by setting the TIMER bit, DMA transfers within a bus mastership period may consist of any mixture of read and write cycles, without restriction on the address ordering. This mode of operation allows the PCnet-ISA II to accomplish more during each bus ownership period.

The number of data transfer cycles contained within a single bus mastership period is in general dependent on the programming of the DMAPLUS (CSR4, bit 14) and the TIMER (CSR4, bit 13) options. Several other factors will also affect the length of the bus mastership period. The possibilities are as follows:

If DMAPLUS = 0 and TIMER = 0, a maximum of 16 transfers to or from the FIFO will be performed by default. This default value may be changed by writing to the DMA Burst Register (CSR80, bits 7:0). Since TIMER = 0, all FIFO DMA transfers within a bus mastership period will be either read or write cycles, and all transfers will be to adjacent, ascending addresses. Note that DMAPLUS = 0 merely sets a maximum value for the number of FIFO transfers that may occur during one bus mastership period. The minimum number of transfers in the bus mastership period will be determined by the settings of the FIFO watermarks and the conditions of the FIFOs, and the value of the Bus Activity Timer (CSR82) if the TIMER bit is set.

If DMAPLUS = 1 and TIMER = 0, the bus mastership period will continue until the transmit FIFO is filled to its high threshold (read transfers) or the receive FIFO is emptied to its low threshold (write transfers). Other variables may also affect the end point of the bus mastership period in this mode, including the particular conditions existing within the FIFOs, and receive and transmit status conditions. Since TIMER = 0, all FIFO DMA transfers within a bus mastership period will be either read or write cycles, and all transfers will be to adjacent, ascending addresses.

If TIMER = 1, the bus mastership period will continue until all “pending bus operations” are completed or until the Bus Activity Timer value (CSR82) has expired. These bus operations may consist of any mixture of descriptor and buffer read and write accesses. If DMAPLUS = 1, “pending bus operations” includes any descriptor accesses and buffer accesses that need to be performed. If DMAPLUS = 0, “pending bus operations” include any descriptor accesses that need to be performed and any buffer accesses that need to be performed up to the limit specified by the DMA Burst Register (CSR80, bits 7:0).

Note that when TIMER=1, following a last bus transaction during a bus mastership period, the PCnet-ISA II may keep ownership of the bus for up to approximately 1µs. The PCnet-ISA II determines whether there are further pending bus operations by waiting approximately 1µs after the completion of every bus operation (e.g. a descriptor or FIFO access). If, during the 1 µs period, no further bus operations are requested by the internal Buffer Management Unit, the PCnet-ISA II determines that there are no further pending operations and gives up bus ownership. This 1 µs of unused bus ownership time is more than made up for by the efficiency gained by being able to perform any mixture of descriptor and buffer read and write accesses during a single bus ownership period.

The FIFO thresholds are programmable (see description of CSR80), as are the DMA Burst Register and Bus Activity Timer values. The exact number of transfer cycles in the case of DMAPLUS = 1 will be dependent on the latency of the system bus to the PCnet-ISA II controller’s DMA request and the speed of bus operation, but will be limited by the value in the Bus Activity Timer register (if the TIMER bit is set), the FIFO condition, and receive and transmit status. Barring a time-out by either of these registers, or exceptional receive and transmit events, or an end of packet signal from the FIFO, the FIFO watermark settings and the extent of Bus Grant latency will be the major factors determining the number of accesses performed during any given arbitration cycle when DMAPLUS = 1.

The IOCHRDY response of the memory device will also affect the number of transfers when DMAPLUS = 1, since the speed of the accesses will affect the state of the FIFO. During accesses, the FIFO may be filling or emptying on the network end. A slower memory response will allow additional data to accumulate inside of the FIFO (during write transfers from the receive FIFO). If the accesses are slow enough, a complete word may become available before the end of the arbitration cycle and thereby increase the number of

transfers in that cycle. The general rule is that the longer the Bus Grant latency or the slower the bus transfer operations (or clock speed) or the higher the transmit watermark or the lower the receive watermark or any combination thereof, the longer will be the average bus mastership period.

Buffer Management Unit (BMU)

The buffer management unit is a microcoded 20 MHz state machine which implements the initialization block and the descriptor architecture.

Initialization

PCnet-ISA II controller initialization includes the reading of the initialization block in memory to obtain the operating parameters. The initialization block is read when the INIT bit in CSR0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure correct operation. See previous section "1. Initialization Block DMA Transfer." Once the initialization block has been read in and processed, the BMU knows where the receive and transmit descriptor rings are. On completion of the read operation and after internal registers have been updated, IDON will be set in CSR0, and an interrupt generated if IENA is set.

The Initialization Block is vectored by the contents of CSR1 (least significant 16 bits of address) and CSR2 (most significant 8 bits of address). The block contains the user defined conditions for PCnet-ISA II controller operation, together with the address and length information to allow linkage of the transmit and receive descriptor rings.

There is an alternative method to initialize the PCnet-ISA II controller. Instead of initialization via the initialization block in memory, data can be written directly into the appropriate registers. Either method may be used at the discretion of the programmer. If the registers are written to directly, the INIT bit must not be set, or the initialization block will be read in, thus overwriting the previously written information. Please refer to Appendix D for details on this alternative method.

Reinitialization

The transmitter and receiver section of the PCnet-ISA II controller can be turned on via the initialization block (MODE Register DTX, DRX bits; CSR15[1:0]). The state of the transmitter and receiver are monitored through CSR0 (RXON, TXON bits). The PCnet-ISA II controller should be reinitialized if the transmitter and/or the receiver were not turned on during the original initialization and it was subsequently required to activate them, or if either section shut off due to the detection of an error condition (MERR, UFLO, TX BUFF error).

Reinitialization may be done via the initialization block or by setting the STOP bit in CSR0, followed by writing to CSR15, and then setting the START bit in CSR0. Note that this form of restart will not perform the same in the PCnet-ISA II controller as in the LANCE. In particular, the PCnet-ISA II controller reloads the transmit and receive descriptor pointers (working registers) with their respective base addresses. This means that the software must clear the descriptor's own bits and reset its descriptor ring pointers before the restart of the PCnet-ISA controller. The reload of descriptor base addresses is performed in the LANCE only after initialization, so a restart of the LANCE without initialization leaves the LANCE pointing at the same descriptor locations as before the restart.

Suspend

The PCnet-ISA II controller offers a suspend mode that allows easy updating of the CSR registers without going through a full reinitialization of the device. The suspend mode also allows stopping the device with orderly termination of all network activity.

The host requests the PCnet-ISA II controller to enter the suspend mode by setting SPND (CSR5, bit 0) to ONE. The host must poll SPND until it reads back ONE to determine that the PCnet-ISA II controller has entered the suspend mode. When the host sets SPND to ONE, the PCnet-ISA II controller first finishes all on-going transmit activity and updates the corresponding transmit descriptor entries. It then finishes all on-going receive activity and updates the corresponding receive descriptor entries. It then sets the read-version of SPND to ONE and enters the suspend mode. In suspend mode, all of the CSR registers are accessible. As long as the PCnet-ISA II controller is not reset while in suspend mode (by asserting the RESET pin, reading the RESET register, or by setting the STOP bit), no reinitialization of the device is required after the device comes out of suspend mode. When SPND is set to ZERO, the PCnet-ISA II controller will leave the suspend mode and will continue at the transmit and receive descriptor ring locations where it had left when it entered the suspend mode.

Buffer Management

Buffer management is accomplished through message descriptor entries organized as ring structures in memory. There are two rings, a receive ring and a transmit ring. The size of a message descriptor entry is 4 words (8 bytes).

Descriptor Rings

Each descriptor ring must be organized in a contiguous area of memory. At initialization time (setting the INIT bit in CSR0), the PCnet-ISA II controller reads the user-defined base address for the transmit and receive descriptor rings, which must be on an 8-byte boundary, as well as the number of entries contained in the descriptor rings. By default, a maximum of 128 ring entries is permitted when utilizing the initialization block, which uses values of TLEN and RLEN to specify

the transmit and receive descriptor ring lengths. However, the ring lengths can be manually defined (up to 65535) by writing the transmit and receive ring length registers (CSR76,78) directly.

Each ring entry contains the following information:

- The address of the actual message data buffer in user or host memory
- The length of the message buffer
- Status information indicating the condition of the buffer

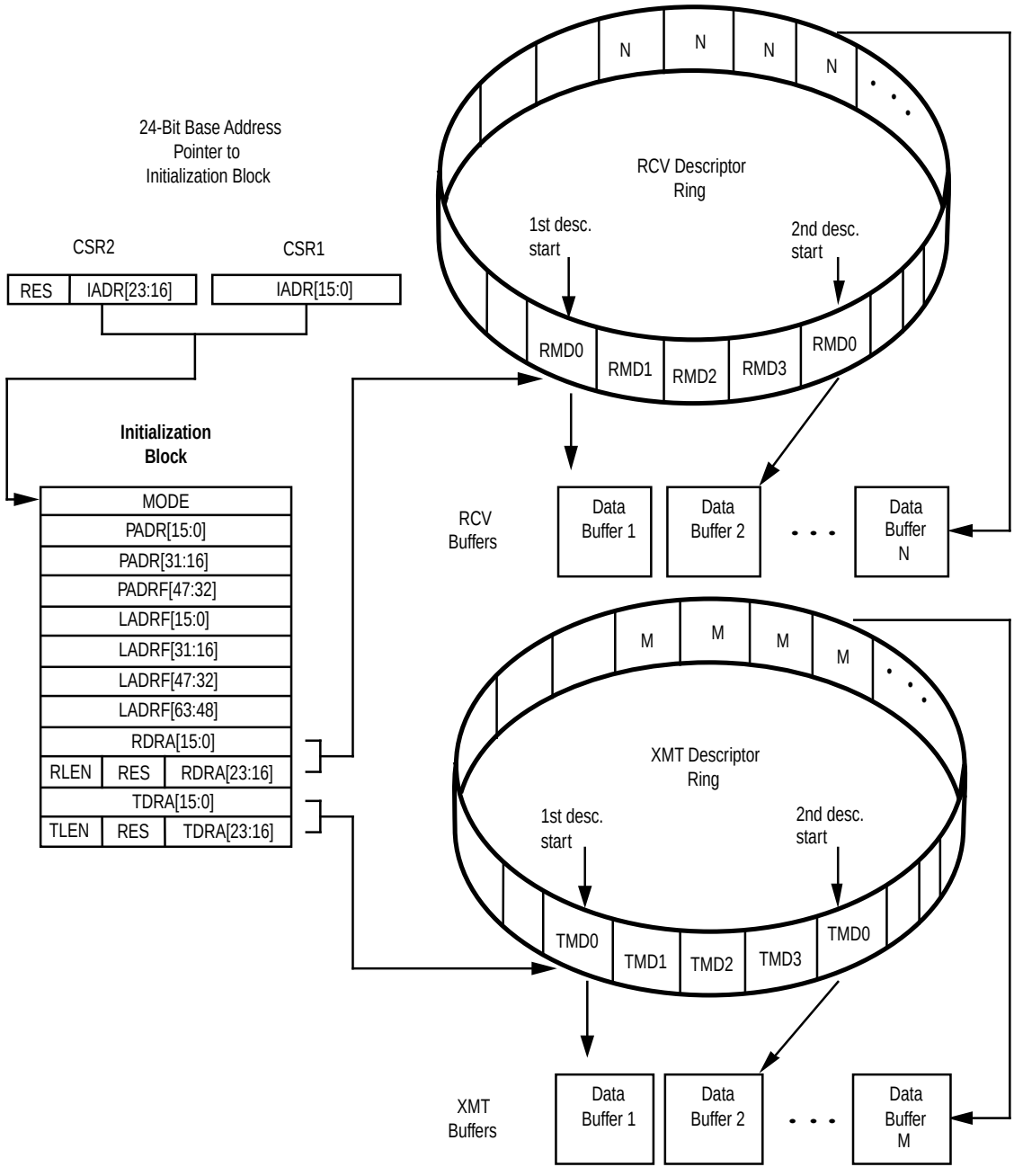
Receive descriptor entries are similar (but not identical) to transmit descriptor entries. Both are composed of four registers, each 16 bits wide for a total of 8 bytes.

To permit the queuing and de-queuing of message buffers, ownership of each buffer is allocated to either the PCnet-ISA II controller or the host. The OWN bit within the descriptor status information, either TMD or RMD (see section on TMD or RMD), is used for this purpose. “Deadly Embrace” conditions are avoided by the ownership mechanism. Only the owner is permitted to relinquish ownership or to write to any field in the descriptor entry. A device that is not the current owner of a descriptor entry cannot assume ownership or change any field in the entry.

Descriptor Ring Access Mechanism

At initialization, the PCnet-ISA II controller reads the base address of both the transmit and receive descriptor rings into CSRs for use by the PCnet-ISA II controller during subsequent operation.

When transmit and receive functions begin, the base address of each ring is loaded into the current descriptor address registers and the address of the next descriptor entry in the transmit and receive rings is computed and loaded into the next descriptor address registers.



Initialization Block and Descriptor Rings

19364A-11

Polling

When there is no channel activity and there is no pre- or post-receive or transmit activity being performed by the PCnet-ISA II controller then the PCnet-ISA II controller will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership. If the DPOLL bit in CSR4 is set, then the transmit polling function is disabled.

A typical polling operation consists of the following: The PCnet-ISA II controller will use the current receive descriptor address stored internally to vector to the appropriate Receive Descriptor Table Entry (RDTE). It will then use the current transmit descriptor address (stored internally) to vector to the appropriate Transmit Descriptor Table Entry (TDTE). These accesses will be made to RMD1 and RMD0 of the current RDTE and TMD1 and TMD0 of the current TDTE at periodic poll-

ing intervals. All information collected during polling activity will be stored internally in the appropriate CSRs. (i.e. CSR18–19, CSR40, CSR20–21, CSR42, CSR50, CSR52). Unowned descriptor status will be internally ignored.

A typical receive poll occurs under the following conditions:

1. PCnet-ISA II controller does not possess ownership of the current RDTE and
   the poll time has elapsed and
   RXON = 1,

or

2. PCnet-ISA II controller does not possess ownership of the next RDTE and
   the poll time has elapsed and
   RXON = 1,

If RXON = 0, the PCnet-ISA II controller will never poll RDTE locations.

If RXON = 1, the system should always have at least one RDTE available for the possibility of a receive event. When there is only one RDTE, there is no polling for next RDTE.

A typical transmit poll occurs under the following conditions:

1. PCnet-ISA II controller does not possess ownership of the current TDTE and
   DPOLL = 0 and
   TXON = 1 and
   the poll time has elapsed,

or

2. PCnet-ISA II controller does not possess ownership of the current TDTE and
   DPOLL = 0 and
   TXON = 1 and
   a packet has just been received,

or

3. PCnet-ISA II controller does not possess ownership of the current TDTE and
   DPOLL = 0 and
   TXON = 1 and
   a packet has just been transmitted.

The poll time interval is nominally defined as 32,768 crystal clock periods, or 1.6 ms. However, the poll time register is controlled internally by microcode, so any other microcode controlled operation will interrupt the incrementing of the poll count register. For example, when a receive packet is accepted by the PCnet-ISA II controller, the device suspends execution of the poll-time-incrementing microcode so that a receive microcode routine may instead be executed. Poll-time-incrementing code is resumed when the receive operation has completely finished. Note, however, that following the completion of any receive or transmit operation, a poll operation will always be performed. The poll time count register is never reset. Note that if a non-default is desired, then a strict sequence of setting the INIT bit in CSR0, waiting for the IDON bit in CSR0, then writing to CSR47, and then setting STRT in CSR0 must be observed, otherwise the default value will not be overwritten. See the CSR47 section for details.

Setting the TDMD bit of CSR0 will cause the microcode controller to exit the poll counting code and immediately perform a polling operation. If RDTE ownership has not been previously established, then an RDTE poll will be performed ahead of the TDTE poll.

Transmit Descriptor Table Entry (TDTE)

If, after a TDTE access, the PCnet-ISA II controller finds that the OWN bit of that TDTE is not set, then the PCnet-ISA II controller resumes the poll time count and re-examines the same TDTE at the next expiration of the poll time count.

If the OWN bit of the TDTE is set, but STP = 0, the PCnet-ISA II controller will immediately request the bus in order to reset the OWN bit of this descriptor; this condition would normally be found following a LCOL or RETRY error that occurred in the middle of a transmit packet chain of buffers. After resetting the OWN bit of this descriptor, the PCnet-ISA II controller will again immediately request the bus in order to access the next TDTE location in the ring.

If the OWN bit is set and the buffer length is 0, the OWN bit will be reset. In the LANCE the buffer length of 0 is interpreted as a 4096-byte buffer. It is acceptable to have a 0 length buffer on transmit with STP=1 or STP=1 *and* ENP = 1. It is not acceptable to have 0 length buffer with STP = 0 *and* ENP = 1.

If the OWN bit is set and the start of packet (STP) bit is set, then microcode control proceeds to a routine that will enable transmit data transfers to the FIFO.

If the transmit buffers are data chained (ENP = 0 in the first buffer), then the PCnet-ISA II controller will look ahead to the next transmit descriptor after it has performed at least one transmit data transfer from the first buffer. More than one transmit data transfer may possibly take place, depending upon the state of the transmitter. The transmit descriptor look ahead reads TMD0 first and TMD1 second. The contents of TMD0 and TMD1 will be stored in Next TX Descriptor Address (CSR32), Next TX Byte Count (CSR66) and Next TX Status (CSR67) regardless of the state of the OWN bit. This transmit descriptor lookahead operation is performed only once.

If the PCnet-ISA II controller does not own the next TDTE (i.e. the second TDTE for this packet), then it will complete transmission of the current buffer and then

update the status of the current (first) TDTE with the BUFF and UFLO bits being set. If DXSUFLO is 0 (bit 6 CSR3), then this will cause the transmitter to be disabled (CSR0, TXON = 0). The PCnet-ISA II controller will have to be restarted to restore the transmit function. The situation that matches this description implies that the system has not been able to stay ahead of the PCnet-ISA II controller in the transmit descriptor ring and therefore, the condition is treated as a fatal error. To avoid this situation, the system should always set the transmit chain descriptor own bits in reverse order.

If the PCnet-ISA II controller does own the second TDTE in a chain, it will gradually empty the contents of the first buffer (as the bytes are needed by the transmit operation), perform a single-cycle DMA transfer to update the status (reset the OWN bit in TMD1) of the first descriptor, and then it may perform one data DMA access on the second buffer in the chain before executing another lookahead operation. (i.e. a lookahead to the third descriptor).

The PCnet-ISA II controller can queue up to two packets in the transmit FIFO. Call them packet "X" and packet "Y", where "Y" is after "X". Assume that packet "X" is currently being transmitted. Because the PCnet-ISA II controller can perform lookahead data transfer over an ENP, it is possible for the PCnet-ISA II controller to update a TDTE in a buffer belonging to packet "Y" while packet "X" is being transmitted if packet "Y" uses data chaining. This operation will result in non-sequential TDTE accesses as packet "X" completes transmission and the PCnet-ISA II controller writes out its status, since packet "X"'s TDTE is before the TDTE accessed as part of the lookahead data transfer from packet "Y".

This should not cause any problem for properly written software which processes buffers in sequence, waiting for ownership before proceeding.

If an error occurs in the transmission before all of the bytes of the current buffer have been transferred, then TMD2 and TMD1 of the current buffer will be written; in that case, data transfers from the next buffer will not commence. Instead, following the TMD2/TMD1 update, the PCnet-ISA II controller will go to the next transmit packet, if any, skipping over the rest of the packet which experienced an error, including chained buffers.

This is done by returning to the polling microcode where it will immediately access the next descriptor and find the condition OWN = 1 and STP = 0 as described earlier. In that case, the PCnet-ISA II controller will reset the own bit for this descriptor and continue in like manner until a descriptor with OWN = 0 (no more transmit packets in the ring) or OWN = 1 and STP = 1 (the first buffer of a new packet) is reached.

At the end of any transmit operation, whether successful or with errors, and the completion of the descriptor updates, the PCnet-ISA II controller will always perform another poll operation. As described earlier, this poll operation will begin with a check of the current RDTE, unless the PCnet-ISA II controller already owns that descriptor. Then the PCnet-ISA II controller will proceed to polling the next TDTE. If the transmit descriptor OWN bit has a zero value, then the PCnet-ISA II controller will resume poll time count incrementation. If the transmit descriptor OWN bit has a value of ONE, then the PCnet-ISA II controller will begin filling the FIFO with transmit data and initiate a transmission. This end-of-operation poll avoids inserting poll time counts between successive transmit packets.

Whenever the PCnet-ISA II controller completes a transmit packet (either with or without error) and writes the status information to the current descriptor, then the TINT bit of CSR0 is set to indicate the completion of a transmission. This causes an interrupt signal if the IENA bit of CSR0 has been set and the TINTM bit of CSR3 is reset.

Receive Descriptor Table Entry (RDTE)

If the PCnet-ISA II controller does not own both the current and the next Receive Descriptor Table Entry, then the PCnet-ISA II controller will continue to poll according to the polling sequence described above. If the receive descriptor ring length is 1, there is no next descriptor, and no look ahead poll will take place.

If a poll operation has revealed that the current and the next RDTE belongs to the PCnet-ISA II controller, then additional poll accesses are not necessary. Future poll operations will not include RDTE accesses as long as the PCnet-ISA II controller retains ownership to the current and the next RDTE.

When receive activity is present on the channel, the PCnet-ISA II controller waits for the complete address of the message to arrive. It then decides whether to accept or reject the packet based on all active addressing schemes. If the packet is accepted the PCnet-ISA II controller checks the current receive buffer status register CRST (CSR40) to determine the ownership of the current buffer.

If ownership is lacking, then the PCnet-ISA II controller will immediately perform a (last ditch) poll of the current RDTE. If ownership is still denied, then the PCnet-ISA II controller has no buffer in which to store the incoming message. The MISS bit will be set in CSR0 and an interrupt will be generated if IENA = 1 (CSR0) and MISSM = 0 (CSR3). Another poll of the current RDTE will not occur until the packet has finished.

If the PCnet-ISA II controller sees that the last poll (either a normal poll or the last-ditch effort described in the above paragraph) of the current RDTE shows valid ownership, then it proceeds to a poll of the next RDTE.

Following this poll, and regardless of the outcome of this poll, transfers of receive data from the FIFO may begin.

Regardless of ownership of the second receive descriptor, the PCnet-ISA II controller will continue to perform receive data DMA transfers to the first buffer, using burst-cycle DMA transfers. If the packet length exceeds the length of the first buffer, and the PCnet-ISA II controller does not own the second buffer, ownership of the current descriptor will be passed back to the system by writing a zero to the OWN bit of RMD1 and status will be written indicating buffer (BUFF = 1) and possibly overflow (OFLO = 1) errors.

If the packet length exceeds the length of the first (current) buffer, and the PCnet-ISA II controller does own the second (next) buffer, ownership will be passed back to the system by writing a zero to the OWN bit of RMD1 when the first buffer is full. Receive data transfers to the second buffer may occur before the PCnet-ISA II controller proceeds to look ahead to the ownership of the third buffer. Such action will depend upon the state of the FIFO when the status has been updated on the first descriptor. In any case, lookahead will be performed to the third buffer and the information gathered will be stored in the chip, regardless of the state of the ownership bit. As in the transmit flow, lookahead operations are performed only once.

This activity continues until the PCnet-ISA II controller recognizes the completion of the packet (the last byte of this receive message has been removed from the FIFO). The PCnet-ISA II controller will subsequently update the current RDTE status with the end of packet (ENP) indication set, write the message byte count (MCNT) of the complete packet into RMD2 and overwrite the “current” entries in the CSRs with the “next” entries.

Media Access Control

The Media Access Control engine incorporates the essential protocol requirements for operation of a compliant Ethernet/802.3 node, and provides the interface between the FIFO sub-system and the Manchester Encoder/Decoder (MENDEC).

This section describes operation of the MAC engine when operating in Half Duplex mode. When in Half Duplex mode, the MAC engine is fully compliant to Section 4 of ISO/IEC 8802-3 (ANSI/IEEE Standard 1990 Second Edition) and ANSI/IEEE 802.3 (1985). When operating in Full Duplex mode, the MAC engine behavior changes as described in the Full Duplex Operation section.

The MAC engine provides programmable enhanced features designed to minimize host supervision and pre or post-message processing. These features include the ability to disable retries after a collision, dynamic FCS generation on a packet-by-packet basis, and automatic pad field insertion and deletion to enforce minimum frame size attributes.

The two primary attributes of the MAC engine are:

- Transmit and receive message data encapsulation
  - — Framing (frame boundary delimitation, frame synchronization)
  - — Addressing (source and destination address handling)
  - — Error detection (physical medium transmission errors)
- Media access management
  - — Medium allocation (collision avoidance)
  - — Contention resolution (collision handling)

Transmit and Receive Message Data Encapsulation

The MAC engine provides minimum frame size enforcement for transmit and receive packets. When APAD_XMT = 1 (bit 11 in CSR4), transmit messages will be padded with sufficient bytes (containing 00h) to ensure that the receiving station will observe an information field (destination address, source address, length/type, data and FCS) of 64-bytes. When ASTRP_RCV = 1 (bit 10 in CSR4), the receiver will automatically strip pad bytes from the received message by observing the value in the length field, and stripping excess bytes if this value is below the minimum data size (46 bytes). Both features can be independently over-ridden to allow illegally short (less than 64 bytes of packet data) messages to be transmitted and/or received. The use of these features reduce bus bandwidth usage because the pad bytes are not transferred to or from host memory.

Framing (frame boundary delimitation, frame synchronization)

The MAC engine will autonomously handle the construction of the transmit frame. Once the Transmit FIFO has been filled to the predetermined threshold (set by XMTSP in CSR80), and providing access to the channel is currently permitted, the MAC engine will commence the 7-byte preamble sequence (10101010b, where first bit transmitted is a 1). The MAC engine will subsequently append the Start Frame Delimiter (SFD) byte (10101011b) followed by the serialized data from the Transmit FIFO. Once the data has been completed, the MAC engine will append the FCS (most significant bit first) which was computed on the entire data portion of the message.

Note that the user is responsible for the correct ordering and content in each of the fields in the frame, including the destination address, source address, length/type and packet data.

The receive section of the MAC engine will detect an incoming preamble sequence and lock to the encoded

clock. The internal MENDEC will decode the serial bit stream and present this to the MAC engine. The MAC will discard the first 8 bits of information before searching for the SFD sequence. Once the SFD is detected, all subsequent bits are treated as part of the frame. The MAC engine will inspect the length field to ensure minimum frame size, strip unnecessary pad characters (if enabled), and pass the remaining bytes through the Receive FIFO to the host. If pad stripping is performed, the MAC engine will also strip the received FCS bytes, although the normal FCS computation and checking will occur. Note that apart from pad stripping, the frame will be passed unmodified to the host. If the length field has a value of 46 or greater, the MAC engine will not attempt to validate the length against the number of bytes contained in the message.

If the frame terminates or suffers a collision before 64 bytes of information (after SFD) have been received, the MAC engine will automatically delete the frame from the Receive FIFO, without host intervention.

Addressing (source and destination address handling)

The first 6 bytes of information after SFD will be interpreted as the destination address field. The MAC engine provides facilities for physical, logical, and broadcast address reception. In addition, multiple physical addresses can be constructed (perfect address filtering) using external logic in conjunction with the EADI interface.

Error detection (physical medium transmission errors)

The MAC engine provides several facilities which report and recover from errors on the medium. In addition, the network is protected from gross errors due to inability of the host to keep pace with the MAC engine activity.

On completion of transmission, the following transmit status is available in the appropriate TMD and CSR areas:

- The exact number of transmission retry attempts (ONE, MORE, or RTRY).
- Whether the MAC engine had to Defer (DEF) due to channel activity.
- Loss of Carrier, indicating that there was an interruption in the ability of the MAC engine to monitor its own transmission. Repeated LCAR errors indicate a potentially faulty transceiver or network connection.
- Late Collision (LCOL) indicates that the transmission suffered a collision after the slot time. This is indicative of a badly configured network. Late collisions should not occur in a normal operating network.
- Collision Error (CERR) indicates that the transceiver did not respond with an SQE Test message within the predetermined time after a transmission completed. This may be due to a failed transceiver, disconnected or faulty transceiver drop cable, or the fact the transceiver does not support this feature (or the feature is disabled).

In addition to the reporting of network errors, the MAC engine will also attempt to prevent the creation of any network error due to the inability of the host to service the MAC engine. During transmission, if the host fails to keep the Transmit FIFO filled sufficiently, causing an underflow, the MAC engine will guarantee the message is either sent as a runt packet (which will be deleted by the receiving station) or has an invalid FCS (which will also cause the receiver to reject the message).

The status of each receive message is available in the appropriate RMD and CSR areas. FCS and Framing errors (FRAM) are reported, although the received frame is still passed to the host. The FRAM error will only be reported if an FCS error is detected and there are a non-integral number of bits in the message. The MAC engine will ignore up to seven additional bits at the end of a message (dribbling bits), which can occur under normal network operating conditions. The reception of eight additional bits will cause the MAC engine to de-serialize the entire byte, and will result in the received message and FCS being modified.

The PCnet-ISA II controller can handle up to 7 dribbling bits when a received packet terminates. During the reception, the CRC is generated on every serial bit (including the dribbling bits) coming from the cable, although the internally saved CRC value is only updated on the eighth bit (on each byte boundary). The framing error is reported to the user as follows:

1. If the number of the dribbling bits are 1 to 7 and there is no CRC error, then there is no Framing error (FRAM = 0).
2. If the number of the dribbling bits are less than 8 and there is a CRC error, then there is also a Framing error (FRAM = 1).
3. If the number of dribbling bits = 0, then there is no Framing error. There may or may not be a CRC (FCS) error.

Counters are provided to report the Receive Collision Count and Runt Packet Count used for network statistics and utilization calculations.

Note that if the MAC engine detects a received packet which has a 00b pattern in the preamble (after the first 8 bits, which are ignored), the entire packet will be ignored. The MAC engine will wait for the network to go inactive before attempting to receive the next packet.

Media Access Management

The basic requirement for all stations on the network is to provide fairness of channel allocation. The 802.3/ Ethernet protocol defines a media access mechanism which permits all stations to access the channel with equality. Any node can attempt to contend for the channel by waiting for a predetermined time (Inter Packet Gap interval) after the last activity, before transmitting on the medium. The channel is a multidrop communications medium (with various topological configurations permitted) which allows a single station to transmit and all other stations to receive. If two nodes simultaneously contend for the channel, their signals will interact, causing loss of data (defined as a collision). It is the responsibility of the MAC to attempt to avoid and recover from a collision, to guarantee data integrity for the end-to-end transmission to the receiving station.

Medium Allocation (collision avoidance)

The IEEE 802.3 Standard (ISO/IEC 8802-3 1990) requires that the CSMA/CD MAC monitor the medium traffic by looking for carrier activity. When carrier is detected the medium is considered busy, and the MAC should defer to the existing message.

The IEEE 802.3 Standard also allows optional two part deferral after a receive message.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.1:

> ***"Note**: It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the interpacket gap based on this indication it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in 4.2.8, are recommended when InterFrameSpacingPart1 is other than zero:*
>
> *(1) Upon completing a transmission, start timing the interpacket gap, as soon as transmitting and carrier Sense are both false.*
>
> *(2) When timing an interpacket gap following reception, reset the interpacket gap timing if carrier Sense becomes true during the first 2/3 of the interpacket gap timing interval. During the final 1/3 of the interval the timer shall not be reset to ensure fair access to the medium. An initial period shorter than 2/3 of the interval is permissible including zero."*

The MAC engine implements the optional receive two part deferral algorithm, with a first part inter-frame-spacing time of 6.0 µs. The second part of the inter-frame-spacing interval is therefore 3.6 µs.

The PCnet-ISA II controller will perform the two-part deferral algorithm as specified in Section 4.2.8 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 9.6 µs InterFrameSpacing after the receive carrier is de-asserted. During the first part deferral (InterFrameSpacingPart1 – IFS1) the PCnet-ISA II controller will defer any pending transmit frame and respond to the receive message. The IPG counter will be reset to zero continuously until the carrier de-asserts, at which point the IPG counter will resume the 9.6 µs count once again. Once the IFS1 period of 6.0 µs has elapsed, the PCnet-ISA II controller will begin timing the second part deferral (InterFrameSpacingPart2 – IFS2) of 3.6 µs. Once IFS1 has completed, and IFS2 has commenced, the PCnet-ISA II controller will not defer to a receive packet if a transmit packet is pending. This means that the PCnet-ISA II controller will not attempt to receive the receive packet, since it will start to transmit, and generate a collision at 9.6 µs. The PCnet-ISA II controller will guarantee to complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

In addition, transmit two part deferral is implemented as an option which can be disabled using the DXMT2PD bit (CSR3). Two-part deferral after transmission is useful for ensuring that severe IPG shrinkage cannot occur in specific circumstances, causing a transmit message to follow a receive message so closely as to make them indistinguishable.

During the time period immediately after a transmission has been completed, the external transceiver (in the case of a standard AUI connected device), should generate the SQE Test message (a nominal 10 MHz burst of 5-15 bit times duration) on the CI± pair (within 0.6 µs – 1.6 µs after the transmission ceases). During the time period in which the SQE Test message is expected the PCnet-ISA II controller will not respond to receive carrier sense.

See ANSI/IEEE Std 802.3-1990 Edition, 7.2.4.6 (1):

> *"At the conclusion of the output function, the DTE opens a time window during which it expects to see the* signal_quality_error *signal asserted on the Control In circuit. The time window begins when the CARRIER_STATUS becomes CARRIER_OFF. If execution of the output function does not cause CARRIER_ON to occur, no SQE test occurs in the DTE. The duration of the window shall be at least 4.0 µs but no more than 8.0 µs. During the time window the Carrier Sense Function is inhibited."*

The PCnet-ISA II controller implements a carrier sense "blinding" period within 0 – 4.0 µs from de-assertion of carrier sense after transmission. This effectively means that when transmit two part deferral is enabled (DXMT2PD is cleared) the IFS1 time is from 4 µs to 6 µs after a transmission. However, since IPG shrinkage below 4 µs will rarely be encountered on a correctly configured network, and since the fragment size will be

larger than the 4 µs blinding window, then the IPG counter will be reset by a worst case IPG shrinkage/ fragment scenario and the PCnet-ISA II controller will defer its transmission. In addition, the PCnet-ISA II controller will not restart the "blinding" period if carrier is detected within the 4.0 µs – 6.0 µs IFS1 period, but will commence timing of the entire IFS1 period.

Contention resolution (collision handling)

Collision detection is performed and reported to the MAC engine by the integrated Manchester Encoder/ Decoder (MENDEC).

If a collision is detected before the complete preamble/ SFD sequence has been transmitted, the MAC Engine will complete the preamble/SFD before appending the jam sequence. If a collision is detected after the preamble/SFD has been completed, but prior to 512 bits being transmitted, the MAC Engine will abort the transmission, and append the jam sequence immediately. The jam sequence is a 32-bit all zeroes pattern.

The MAC Engine will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of collision will cause the transmission to be re-scheduled, dependent on the backoff time that the MAC Engine computes. If a single retry was required, the ONE bit will be set in the Transmit Frame Status (TMD1 in the Transmit Descriptor Ring). If more than one retry was required, the MORE bit will be set. If all 16 attempts experienced collisions, the RTRY bit (in TMD2) will be set (ONE and MORE will be clear), and the transmit message will be flushed from the FIFO. If retries have been disabled by setting the DRTY bit in the MODE register (CSR15), the MAC Engine will abandon transmission of the frame on detection of the first collision. In this case, only the RTRY bit will be set and the transmit message will be flushed from the FIFO.

If a collision is detected after 512 bit times have been transmitted, the collision is termed a late collision. The MAC Engine will abort the transmission, append the jam sequence, and set the LCOL bit. No retry attempt will be scheduled on detection of a late collision, and the FIFO will be flushed.

The IEEE 802.3 Standard requires use of a "truncated binary exponential backoff" algorithm which provides a controlled pseudo-random mechanism to enforce the collision backoff interval, before re-transmission is attempted.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.5:

"At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to re-transmit the frame. The delay is an integer multiple of slot Time. The number of slot times to delay before the nth re-transmission attempt is chosen as a uniformly distributed random integer r in the range:

$0 \le r < 2^k$, where k = min (n,10)."

The PCnet-ISA II controller provides an alternative algorithm, which suspends the counting of the slot time/IPG during the time that receive carrier sense is detected. This algorithm aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. The algorithm effectively accelerates the increase in the backoff time in busy networks, and allows nodes not involved in the collision to access the channel while the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time out their slot time counters as normal.

Manchester Encoder/Decoder (MENDEC)

The integrated Manchester Encoder/Decoder provides the PLS (Physical Layer Signaling) functions required for a fully compliant IEEE 802.3 station. The MENDEC provides the encoding function for data to be transmitted on the network using the high accuracy on-board oscillator, driven by either the crystal oscillator or an external CMOS-level compatible clock. The MENDEC also provides the decoding function from data received from the network. The MENDEC contains a Power On Reset (POR) circuit, which ensures that all analog portions of the PCnet-ISA II controller are forced into their correct state during power-up, and prevents erroneous data transmission and/or reception during this time.

External Crystal Characteristics

When using a crystal to drive the oscillator, the crystal specification shown in the specification table may be used to ensure less than ±0.5 ns jitter at DO±.

External Crystal Characteristics

Parameter	Min	Nom	Max	Unit
1. Parallel Resonant Frequency		20		MHz
2. Resonant Frequency Error (CL = 20 pF)	–50		+50	PPM
3.Change in Resonant Frequency With Respect To Temperature (0° – 70° C; CL = 20 pF)*	–40		+40	PPM
4. Crystal Capacitance			20	pF
5. Motional Crystal Capacitance (C1)		0.022		pF
6. Series Resistance			25	Ω
7. Shunt Capacitance			7	pF
8. Drive Level			TBD	mW

Requires trimming crystal spec; no trim is 50 ppm total

External Clock Drive Characteristics

When driving the oscillator from an external clock source, XTAL2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than ±0.5 ns jitter at DO±.

Clock Frequency:	20 MHz ±0.01%
Rise/Fall Time (tR/tF):	< 6 ns from 0.5 V to V_{DD}–0.5
XTAL1 HIGH/LOW Time (tHIGH/tLOW):	40 – 60% duty cycle
XTAL1 Falling Edge to Falling Edge Jitter:	< ±0.2 ns at 2.5 V input (VDD/2)

MENDEC Transmit Path

The transmit section encodes separate clock and NRZ data input signals into a standard Manchester encoded serial bit stream. The transmit outputs (DO±) are designed to operate into terminated transmission lines. When operating into a 78 Ω terminated transmission line, the transmit signaling meets the required output levels and skew for Cheapernet, Ethernet, and IEEE-802.3.

Transmitter Timing and Operation

A 20 MHz fundamental-mode crystal oscillator provides the basic timing reference for the MENDEC portion of the PCnet-ISA II controller. The crystal input is divided by two to create the internal transmit clock reference. Both clocks are fed into the Manchester Encoder to generate the transitions in the encoded data stream. The internal transmit clock is used by the MENDEC to internally synchronize the Internal Transmit Data (ITXDAT) from the controller and Internal Transmit Enable (ITXEN). The internal transmit clock is also used as a stable bit-rate clock by the receive section of the MENDEC and controller.

The oscillator requires an external 0.005% crystal, or an external 0.01% CMOS-level input as a reference. The accuracy requirements, if an external crystal is used, are tighter because allowance for the on-chip oscillator must be made to deliver a final accuracy of 0.01%.

Transmission is enabled by the controller. As long as the ITXEN request remains active, the serial output of the controller will be Manchester encoded and appear at DO±. When the internal request is dropped by the controller, the differential transmit outputs go to one of two idle states, dependent on TSEL in the Mode Register (CSR15, bit 9):

TSEL LOW:	The idle state of DO± yields “zero” differential to operate transformer-coupled loads
TSEL HIGH:	In this idle state, DO+ is positive with respect to DO– (logical HIGH).

Receive Path

The principal functions of the receiver are to signal the PCnet-ISA II controller that there is information on the receive pair, and to separate the incoming Manchester encoded data stream into clock and NRZ data.

The receiver section (see Receiver Block Diagram) consists of two parallel paths. The receive data path is a zero threshold, wide bandwidth line receiver. The carrier path is an offset threshold bandpass detecting line receiver. Both receivers share common bias networks to allow operation over a wide input common mode range.

Input Signal Conditioning

Transient noise pulses at the input data stream are rejected by the Noise Rejection Filter. Pulse width rejection is proportional to transmit data rate which is fixed at 10 MHz for Ethernet systems but which could be different for proprietary networks. DC inputs more negative than minus 100 mV are also suppressed.

The Carrier Detection circuitry detects the presence of an incoming data packet by discerning and rejecting noise from expected Manchester data, and controls the stop and start of the phase-lock loop during clock acquisition. Clock acquisition requires a valid Manchester bit pattern of 1010b to lock onto the incoming message.

When input amplitude and pulse width conditions are met at DI±, a clock acquisition cycle is initiated.

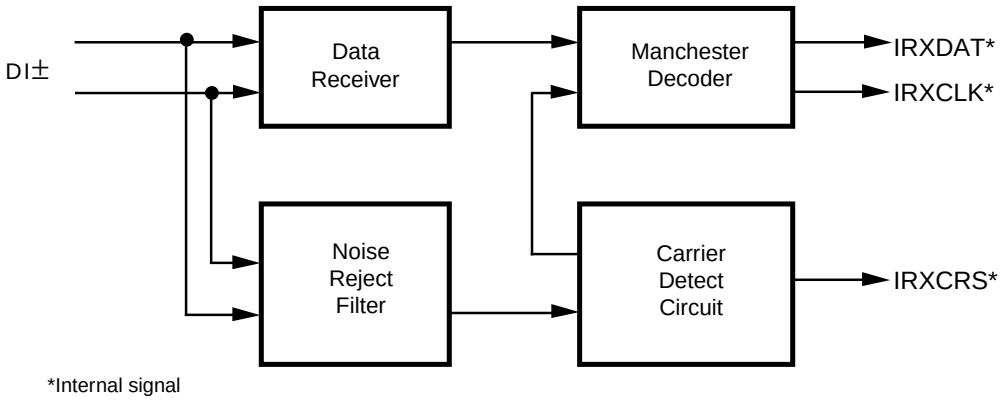
Clock Acquisition

When there is no activity at DI± (receiver is idle), the receive oscillator is phase-locked to STDCLK. The first negative clock transition (bit cell center of first valid Manchester “0") after clock acquisition begins interrupts the receive oscillator. The oscillator is then restarted at the second Manchester “0" (bit time 4) and is phase-locked to it. As a result, the MENDEC acquires the clock from the incoming Manchester bit pattern in 4 bit times with a “1010" Manchester bit pattern.

The internal receiver clock, IRXCLK, and the internal received data, IRXDAT, are enabled 1/4 bit time after clock acquisition in bit cell 5. IRXDAT is at a HIGH state when the receiver is idle (no IRXCLK). IRXDAT however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever IRXCLK is enabled. At 1/4 bit time through bit cell 5, the controller portion of the PCnet-ISA II controller sees the first IRXCLK transition. This also strobes in the incoming fifth bit to the MENDEC as Manchester “1". IRXDAT may make a transition after the IRXCLK rising edge in bit cell 5, but its state is still undefined. The Manchester “1" at bit 5 is clocked to IRXDAT output at 1/4 bit time in bit cell 6.

PLL Tracking

After clock acquisition, the phase-locked clock is compared to the incoming transition at the bit cell center (BCC) and the resulting phase error is applied to a correction circuit. This circuit ensures that the phase-locked clock remains locked on the received signal. Individual bit cell phase corrections of the Voltage Controlled Oscillator (VCO) are limited to 10% of the phase difference between BCC and phase-locked clock.



Receiver Block Diagram

Carrier Tracking and End of Message

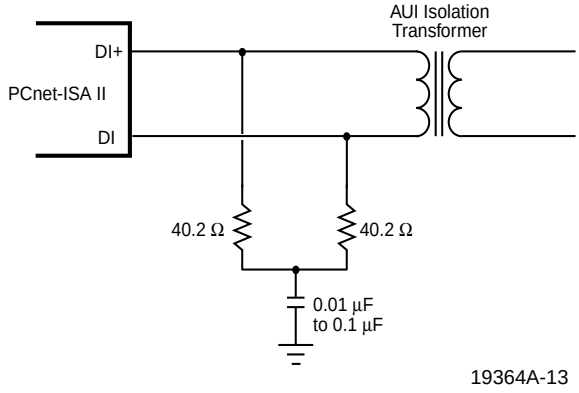
The carrier detection circuit monitors the DI± inputs after IRXCRS is asserted for an end of message. IRXCRS de-asserts 1 to 2 bit times after the last positive transition on the incoming message. This initiates the end of reception cycle. The time delay from the last rising edge of the message to IRXCRS deassert allows the last bit to be strobed by IRXCLK and transferred to the controller section, but prevents any extra bit(s) at the end of message. When IRXCRS de-asserts an IRXCRS hold off timer inhibits IRXCRS assertion for at least 2 bit times.

Data Decoding

The data receiver is a comparator with clocked output to minimize noise sensitivity to the DI± inputs. Input error is less than ±35 mV to minimize sensitivity to input rise and fall time. IRXCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit, and clocks the data out on IRXDAT on the following IRXCLK. The data receiver also generates the signal used for phase detector comparison to the internal MENDEC voltage controlled oscillator (VCO).

Differential Input Terminations

The differential input for the Manchester data (DI±) should be externally terminated by two 40.2 Ω ±1% resistors and one optional common-mode bypass capacitor, as shown in the Differential Input Termination diagram below. The differential input impedance, Z_{IDF}, and the common-mode input impedance, Z_{ICM}, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. If SIP devices are used, 39 Ω is the nearest usable equivalent value. The CI± differential inputs are terminated in exactly the same way as the DI± pair.



Differential Input Termination

Collision Detection

A MAU detects the collision condition on the network and generates a differential signal at the CI± inputs. This collision signal passes through an input stage which detects signal levels and pulse duration. When the signal is detected by the MENDEC it sets the internal collision signal, ICLSN, HIGH. The condition continues for approximately 1.5 bit times after the last LOW-to-HIGH transition on CI±.

Jitter Tolerance Definition

The MENDEC utilizes a clock capture circuit to align its internal data strobe with an incoming bit stream. The clock acquisition circuitry requires four valid bits with the values 1010b. Clock is phase-locked to the negative transition at the bit cell center of the second “0" in the pattern.

Since data is strobed at 1/4 bit time, Manchester transitions which shift from their nominal placement through 1/4 bit time will result in improperly decoded data. With this as the criteria for an error, a definition of “Jitter Handling” is:

> The peak deviation approaching or crossing 1/4 bit cell position from nominal input transition, for which the MENDEC section will properly decode data.

Attachment Unit Interface (AUI)

The AUI is the PLS (Physical Layer Signaling) to PMA (Physical Medium Attachment) interface which connects the DTE to a MAU. The differential interface provided by the PCnet-ISA II controller is fully compliant with Section 7 of ISO 8802-3 (ANSI/IEEE 802.3).

After the PCnet-ISA II controller initiates a transmission, it will expect to see data “looped-back” on the DI± pair (when the AUI port is selected). This will internally generate a “carrier sense”, indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This “carrier sense” signal must be asserted within sometime before end of transmission. If “carrier sense” does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Descriptor Ring (TMD3, bit 11) after the packet has been transmitted.

Twisted Pair Transceiver (T-MAU)

This section describes operation of the T-MAU when operating in the Half Duplex mode. When in Half Duplex mode, the T-MAU implements the Medium Attachment Unit (MAU) functions for the Twisted Pair Medium as specified by the supplement to IEEE 802.3 standard (Type 10BASE-T). When operating in Full Duplex mode, the MAC engine behavior changes as described in the Full Duplex Operation section. The T-MAU provides twisted pair driver and receiver circuits, including on-board transmit digital predistortion and receiver squelch, and a number of additional features including Link Status indication, Automatic Twisted Pair Receive Polarity Detection/Correction and Indication, Receive Carrier Sense, Transmit Active and Collision Present indication.

Twisted Pair Transmit Function

The differential driver circuitry in the TXD± and TXP± pins provides the necessary electrical driving capability and the pre-distortion control for transmitting signals over maximum length Twisted Pair cable, as specified by the 10BASE-T supplement to the IEEE 802.3 Standard. The transmit function for data output meets the propagation delays and jitter specified by the standard.

Twisted Pair Receive Function

The receiver complies with the receiver specifications of the IEEE 802.3 10BASE-T Standard, including noise immunity and received signal rejection criteria (‘Smart Squelch’). Signals meeting these criteria appearing at the RXD± differential input pair are routed to the MENDEC. The receiver function meets the propagation delays and jitter requirements specified by the standard. The receiver squelch level drops to half its threshold value after unsquelch to allow reception of minimum amplitude signals and to offset carrier fade in the event of worst case signal attenuation conditions.

Note that the 10BASE-T Standard defines the receive input amplitude at the external Media Dependent Interface (MDI). Filter and transformer loss are not specified. The T-MAU receiver squelch levels are designed to account for a 1 dB insertion loss at 10 MHz for the type of receive filters and transformers usually used.

Normal 10BASE-T compatible receive thresholds are invoked when the LRT bit (CSR15, bit 9) is LOW. When the LRT bit is set, the Low Receive Threshold option is invoked, and the sensitivity of the T-MAU receiver is increased. Increasing T-MAU sensitivity allows the use of lines longer than the 100 m target distance of standard 10BASE-T (assuming typical 24 AWG cable). Increased receiver sensitivity compensates for the increased signal attenuation caused by the additional cable distance.

However, making the receiver more sensitive means that it is also more susceptible to extraneous noise, primarily caused by coupling from co-resident services (crosstalk). For this reason, end users may wish to invoke the Low Receive Threshold option on 4-pair cable only. Multi-pair cables within the same outer sheath have lower crosstalk attenuation, and may allow noise emitted from adjacent pairs to couple into the receive pair, and be of sufficient amplitude to falsely unsquelch the T-MAU.

Link Test Function

The link test function is implemented as specified by 10BASE-T standard. During periods of transmit pair inactivity,’Link beat pulses’ will be periodically sent over the twisted pair medium to constantly monitor medium integrity.

When the link test function is enabled (DLNKTST bit in CSR15 is cleared), the absence of link beat pulses and receive data on the RXD± pair will cause the TMAU to go into the Link Fail state. In the Link Fail state, data transmission, data reception, data loopback and the collision detection functions are disabled and remain disabled until valid data or greater than 5 consecutive link pulses appear on the RXD± pair. During Link Fail, the Link Status (LNKST indicated by $\overline{\text{LED0}}$) signal is inactive. When the link is identified as functional, the LNKST signal is asserted, and $\overline{\text{LED0}}$ output will be activated. Upon power up or assertion of the RESET pin, the T-MAU will be forced into the Link Fail state. Reading the RESET register of the PCnet-ISA$^+$ (software RESET) has no effect on the T-MAU

In order to inter-operate with systems which do not implement Link Test, this function can be disabled by setting the DLNKTST bit. With Link Test disabled, the Data Driver, Receiver and Loopback functions as well as Collision Detection remain enabled irrespective of the presence or absence of data or link pulses on the RXD± pair. Link Test pulses continue to be sent regardless of the state of the DLNKTST bit.

Polarity Detection and Reversal

The T-MAU receive function includes the ability to invert the polarity of the signals appearing at the RXD± pair if the polarity of the received signal is reversed (such as in the case of a wiring error). This feature allows data packets received from a reverse wired RXD± input pair to be corrected in the T-MAU prior to transfer to the MENDEC. The polarity detection function is activated following reset or Link Fail, and will reverse the receive polarity based on both the polarity of any previous link beat pulses and the polarity of subsequent packets with a valid End Transmit Delimiter (ETD).

When in the Link Fail state, the T-MAU will recognize link beat pulses of either positive or negative polarity. Exit from the Link Fail state occurs at the reception of 5 – 6 consecutive link beat pulses of identical polarity. On entry to the Link Pass state, the polarity of the last 5 link beat pulses is used to determine the initial receive polarity configuration and the receiver is reconfigured to subsequently recognize only link beat pulses of the previously recognized polarity.

Positive link beat pulses are defined as transmitted signal with a positive amplitude greater than 585 mV with a pulse width of 60 ns – 200 ns. This positive excursion may be followed by a negative excursion. This definition is consistent with the expected received signal at a correctly wired receiver, when a link beat pulse, which fits the template of Figure 14-12 of the 10BASE-T Standard, is generated at a transmitter and passed through 100 m of twisted pair cable.

Negative link beat pulses are defined as transmitted signals with a negative amplitude greater than 585 mV with a pulse width of 60 ns – 200 ns. This negative excursion may be followed by a positive excursion. This definition is consistent with the expected received signal at a reverse wired receiver, when a link beat pulse which fits the template of Figure 14-12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

The polarity detection/correction algorithm will remain “armed” until two consecutive packets with valid ETD of identical polarity are detected. When “armed,” the receiver is capable of changing the initial or previous polarity configuration according to the detected ETD polarity.

On receipt of the first packet with valid ETD following reset or link fail, the T-MAU will use the inferred polarity information to configure its RXD± input, regardless of its previous state. On receipt of a second packet with a valid ETD with correct polarity, the detection/correction algorithm will “lock-in” the received polarity. If the second (or subsequent) packet is not detected as confirming the previous polarity decision, the most recently detected ETD polarity will be used as the default. Note that packets with invalid ETD have no effect on updating the previous polarity decision. Once two consecutive packets with valid ETD have been received, the T-MAU will lock the correction algorithm until either a Link Fail condition occurs or RESET is asserted.

During polarity reversal, an internal POL signal will be active. During normal polarity conditions, this internal POL signal is inactive. The state of this signal can be read by software and/or displayed by LED when enabled by the LED control bits in the ISA Bus Configuration Registers (ISACSR5, 6, 7).

Twisted Pair Interface Status

Three internal signals (XMT, RCV and COL) indicate whether the T-MAU is transmitting, receiving, or in a collision state. These signals are internal signals and the behavior of the LED outputs depends on how the LED output circuitry is programmed.

The T-MAU will power up in the Link Fail state and the normal algorithm will apply to allow it to enter the Link Pass state. In the Link Pass state, transmit or receive activity will be indicated by assertion of RCV signal going active. If T-MAU is selected using the PORTSEL bits in CSR15, when moving from AUI to T-MAU selection, the T-MAU will be forced into the Link Fail state.

In the Link Fail state, XMT, RCV and COL are inactive.

Collision Detect Function

Activity on both twisted pair signals RXD± and TXD± constitutes a collision, thereby causing the COL signal to be asserted. (COL is used by the LED control circuits) COL will remain asserted until one of the two col-

liding signals changes from active to idle. COL stays active for 2 bit times at the end of a collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

The SQE function is disabled when the 10BASE-T port is selected and in Link Fail state.

Jabber Function

The Jabber function inhibits the twisted pair transmit function of the T-MAU if the TXD± circuit is active for an excessive period (20 ms–150 ms). This prevents any one node from disrupting the network due to a 'stuck-on' or faulty transmitter. If this maximum transmit time is exceeded, the T-MAU transmitter circuitry is disabled, the JAB bit is set (CSR4, bit 1), and the COL signal asserted. Once the transmit data stream to the T-MAU is removed, an "unjab" time of 250 ms – 750 ms will elapse before the T-MAU deasserts COL and re-enables the transmit circuitry.

Power Down

The T-MAU circuitry can be made to go into low power mode. This feature is useful in battery powered or low duty cycle systems. The T-MAU will go into power down mode when RESET is active, **coma mode** is active, or the T-MAU is not selected. Refer to the Power Down Mode section for a description of the various power down modes.

Any of the three conditions listed above resets the internal logic of the T-MAU and places the device into power down mode. In this mode, the Twisted Pair driver pins (TXD±,TXP±) are asserted LOW, and the internal T-MAU status signals (LNKST, RCVPOL, XMT, RCV and COLLISION) are inactive.

Once the $\overline{\text{SLEEP}}$ pin is deasserted, the T-MAU will be forced into the Link Fail state. The T-MAU will move to the Link Pass state only after 5–6 link beat pulses and/or a single received message is detected on the RXD± pair.

In **Snooze** mode, the T-MAU receive circuitry will remain enabled even while the $\overline{\text{SLEEP}}$ pin is driven LOW.

The T-MAU circuitry will always go into power down mode if RESET is asserted, **coma** is enabled, or the T-MAU is not selected.

Full Duplex Operation

The PCnet-ISA II supports Full Duplex operation on the 10BASE-T, AUI, and GPSI ports. Full Duplex operation allows simultaneous transmit and receive activity on the TXD± and RXD± pairs of the 10BASE-T port, the DO± and DI± pairs of the AUI port, and the TXDAT and RXDAT pins of the GPSI port. It is enabled by the FDEN and AUIFD bits located in ISACSR9. When operating in the Full Duplex mode, the following changes to device operation are made:

Bus Interface/Buffer Management Unit changes:

1. The first 64 bytes of every transmit frame are not preserved in the transmit FIFO during transmission of the first 512 bits transmitted on the network, as described in the Transmit Exception Conditions section. Instead, when Full Duplex mode is active and a frame is being transmitted, the XMTFW bits (CSR80, bits 9, 8) **always** govern when transmit DMA is requested.
2. Successful reception of the first 64 bytes of every receive frame is not a requirement for Receive DMA to begin as described in the Receive Exception Conditions section. Instead, receive DMA will be requested as soon as either the RCVFW threshold (CSR80 bits 12, 13) is reached or a complete valid receive frame is in the Receive FIFO, regardless of length. This receive FIFO operation is identical to when the RPA bit (CSR124, bit 3) is set during Half Duplex mode operation.

MAC Engine changes:

1. Changes to the Transmit Deferral mechanism:
   A. Transmission is not deferred while receive is active.
   B. The Inter Packet Gap (IPG) counter which governs transmit deferral during the IPG between back-to-back transmits is started when transmit activity for the first packet ends instead of when transmit **and** carrier activity ends.
2. When the AUI or GPSI port is active, Loss of Carrier (LCAR) reporting is disabled (LCAR is still reported when the 10BASE-T port is active if a packet is transmitted while in the Link Fail state).
3. The 4.0 µs carrier sense blinding period after a transmission during which the SQE test normally occurs is disabled.
4. When the AUI or GPSI port is active, the SQE Test error (Collision Error, CERR) reporting is disabled (CERR is still reported when the 10BASE-T port is active if a packet is transmitted while in the Link Fail state).
5. The collision indication input to the MAC Engine is ignored.

T-MAU changes:

1. The transmit to receive loopback path in the T-MAU is disabled.
2. The collision detect circuit is disabled.
3. The "heartbeat" generation (SQE Test function) is disabled.

EADI (External Address Detection Interface)

This interface is provided to allow external address filtering. It is selected by setting the EADISEL bit in ISACSR2. This feature is typically utilized for terminal servers, bridges and/or router type products. The use of external logic is required to capture the serial bit stream from the PCnet-ISA II controller, compare it with a table of stored addresses or identifiers, and perform the desired function.

The EADI interface operates directly from the NRZ decoded data and clock recovered by the Manchester decoder or input to the GPSI, allowing the external address detection to be performed in parallel with frame reception and address comparison in the MAC Station Address Detection (SAD) block.

SRDCLK is provided to allow clocking of the receive bit stream into the external address detection logic. SRDCLK runs only during frame reception activity. Once a received frame commences and data and clock are available, the EADI logic will monitor 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 SF/BD output will be driven HIGH.

After SF/BD is asserted the serial data from SRD should be de-serialized and sent to a content addressable memory (CAM) or other address detection device.

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

If an internal address match is detected by comparison with either the Physical or Logical Address field, the frame will be accepted regardless of the condition of $\overline{EAR}$. Incoming frames which do not pass the internal address comparison will continue to be received. This allows approximately 58 byte times after the last destination address bit is available to generate the $\overline{EAR}$ signal, assuming the device is not configured to accept runt packets. $\overline{EAR}$ will be ignored after 64 byte times after the SFD, and the frame will be accepted if $\overline{EAR}$ has not been asserted before this time. If Runt Packet Accept is configured, 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, 4 bytes for FCS) after the last bit of the destination address is available. $\overline{EAR}$ must have a pulse width of at least 200 ns.

Note that setting the PROM bit (CSR15, bit 15) will cause all receive frames to be received, regardless of the state of the $\overline{EAR}$ input.

If the DRCUPA bit (CSR15.B) is set and the logical address (LADRF) is set to zero, only frames which are not rejected by $\overline{EAR}$ will be received.

The EADI interface will operate as long as the STRT bit in CSR0 is set, even if the receiver and/or transmitter are disabled by software (DTX and DRX bits in CSR15 set). This situation is useful as a power down mode in that the PCnet-ISA II controller will not perform any DMA operations; this saves power by not utilizing the ISA bus driver circuits. However, external circuitry could still respond to specific frames on the network to facilitate remote node control.

The table below summarizes the operation of the EADI features.

Internal/External Address Recognition Capabilities

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

General Purpose Serial Interface (GPSI)

The PCnet-ISA II controller contains a General Purpose Serial Interface (GPSI) designed for testing the digital portions of the chip. The MENDEC, AUI, and twisted pair interface are by-passed once the device is set up in the special "test mode" for accessing the GPSI functions. Although this access is intended only for testing the device, some users may find the non-encoded data functions useful in some special applications. Note, however, that the GPSI functions can be accessed only when the PCnet-ISA II devices operate as a bus master.

The PCnet-ISA II GPSI signals are consistent with the LANCE digital serial interface. Since the GPSI functions can be accessed only through a special test mode, expect some loss of functionality to the device when the GPSI is invoked. The AUI and 10BASE-T analog interfaces are disabled along with the internal

MENDEC logic. The LA (unlatched address) pins are removed and become the GPSI signals, therefore, only 20 bits of address space is available. The table below shows the GPSI pin configuration:

To invoke the GPSI signals, follow the procedure below:

1. After reset or I/O read of Reset Address, write 10b to PORTSEL bits in CSR15.
2. Set the ENTST bit in CSR4
3. Set the GPSIEN bit in CSR124 (see note below)

(The pins LA17–LA23 will change function after the completion of the above three steps.)

4. Clear the ENTST bit in CSR4
5. Clear Media Select bits in ISACSR2
6. Define the PORTSEL bits in the MODE register (CSR15) to be 10b to define GPSI port. The MODE register image is in the initialization block.

Note: *LA pins will be tristated before writing to GPSIEN bit. After writing to GPSIEN, LA[17–21] will be inputs, LA[22–23] will be outputs.*

GPSI Pin Configurations

GPSI Function	GPSI I/O Type	LANCE GPSI Pin	PCnet-ISA II GPSI Pin	PCnet-ISA II Pin Number	PCnet-ISA II Normal Pin Function
Receive Data	I	RX	RXDAT	5	LA17
Receive Clock	I	RCLK	SRDCLK	6	LA18
Receive Carrier Sense	I	RENA	RXCRS	7	LA19
Collision	I	CLSN	CLSN	9	LA20
Transmit Clock	I	TCLK	STDCLK	10	LA21
Transmit Enable	O	TENA	TXEN	11	LA22
Transmit Data	O	TX	TXDAT	12	LA23

Note:
The GPSI Function is available only in the Bus Master Mode of operation.

IEEE 1149.1 Test Access Port Interface

An IEEE 1149.1 compatible boundary scan Test Access Port is provided for board-level continuity test and diagnostics. All digital input, output, and input/output pins are tested. Analog pins, including the AUI differential driver (DO±) and receivers (DI±, CI±), and the crystal input (XTAL1/XTAL2) pins, are tested. The T-MAU drivers TXD±, TXP±, and receiver RXD± are also tested.

The following is a brief summary of the IEEE 1149.1 compatible test functions implemented in the PCnet-ISA II controller.

Boundary Scan Circuit

The boundary scan test circuit requires four extra pins (TCK, TMS, TDI and TDO), defined as the Test Access Port (TAP). It includes a finite state machine (FSM), an instruction register, a data register array, and a power-on reset circuit. Internal pull-up resistors are provided for the TDI, TCK, and TMS pins. The TCK pin must not be left unconnected. The boundary scan circuit remains active during sleep.

TAP FSM

The TAP engine is a 16-state FSM, driven by the Test Clock (TCK) and the Test Mode Select (TMS) pins. This FSM is in its reset state at power-up or RESET. An independent power-on reset circuit is provided to ensure the FSM is in the TEST_LOGIC_RESET state at power-up.

Supported Instructions

In addition to the minimum IEEE 1149.1 requirements (BYPASS, EXTEST and SAMPLE instructions), three additional instructions (IDCODE, TRIBYP and SETBYP) are provided to further ease board-level testing. All unused instruction codes are reserved. See the table below for a summary of supported instructions.

Instruction Register and Decoding Logic

After hardware or software RESET, the IDCODE instruction is always invoked. The decoding logic gives signals to control the data flow in the DATA registers according to the current instruction.

Boundary Scan Register (BSR)

Each BSR cell has two stages. A flip-flop and a latch are used in the SERIAL SHIFT STAGE and the PARALLEL OUTPUT STAGE, respectively.

There are four possible operational modes in the BSR cell:

1	Capture
2	Shift
3	Update
4	System Function

Other Data Registers

(1) BYPASS REG (1 BIT)
(2) DEV ID REG (32 bits

Bits 31–28:	Version
Bits 27–12:	Part number **(2261h)**
Bits 11–1:	Manufacturer ID. The 11 bit manufacturer ID code for AMD is 00000000001 according to JEDEC Publication 106-A.
Bit 0:	Always a logic 1

IEEE 1149.1 Supported Instruction Summary

Instruction Name	Description	Selected Data Reg	Mode	Instruction Code
EXTEST	External Test	BSR	Test	0000
IDCODE	ID Code Inspection	ID REG	Normal	0001
SAMPLE	Sample Boundary	BSR	Normal	0010
TRIBYP	Force Tristate	Bypass	Normal	0011
SETBYP	Control Boundary to 1/0	Bypass	Test	0100
BYPASS	Bypass Scan	Bypass	Normal	1111

Power Saving Modes

The PCnet-ISA II controller supports two hardware power-savings modes. Both are entered by asserting the $\overline{\text{SLEEP}}$ pin LOW.

In **coma** mode, the PCnet-ISA II controller will go into deep sleep with no support to automatically wake itself up. Sleep mode is enabled when the AWAKE bit in ISACSR2 is reset. This mode is the default powerdown mode.

In **Snooze** mode, enabled by setting the AWAKE bit in ISACSR2 and driving the $\overline{\text{SLEEP}}$ pin LOW, the T-MAU receive circuitry will remain enabled even while the $\overline{\text{SLEEP}}$ pin is driven LOW. The $\overline{\text{LED0}}$ output will also continue to function, indicating a good 10BASE-T link if

there are link beat pulses or valid frames present. This LED0 pin can be used to drive a LED and/or external hardware that directly controls the SLEEP pin of the PCnet-ISA II controller. This configuration effectively wakes the system when there is any activity on the 10BASE-T link.

Access Operations (Software)

We begin by describing how byte and word data are addressed on the ISA bus, including conversion cycles where 16-bit accesses are turned into 8-bit accesses because the resource accessed did not support 16-bit operations. Then we describe how registers and other resources are accessed. This section is for the device programmer, while the next section (bus cycles) is for the hardware designer.

I/O Resources

The PCnet-ISA II controller has both I/O and memory resources. In the I/O space the resources are organized as indicated in the following table:

The PCnet-ISA II controller does not respond to any addresses outside of the offset range 0-17h. I/O offsets 18h and up are not used by the PCnet-ISA II controller.

Offset	#Bytes	Register
0h	16	IEEE Address
10h	2	RDP
12h	2	RAP (shared by RDP and IDP)
14h	2	Reset
16h	2	IDP

I/O Register Access

The register address port (RAP) is shared by the register data port (RDP) and the ISACSR data port (IDP) to save registers. To access the Ethernet controller's RDP or IDP, the RAP should be written first, followed by the read or write access to the RDP or IDP. I/O register accesses should be coded as 16-bit accesses, even if the PCnet-ISA II controller is hardware configured for 8-bit I/O bus cycles. It is acceptable (and transparent) for the motherboard to turn a 16-bit software access into two separate 8-bit hardware bus cycles. The motherboard accesses the low byte before the high byte and the PCnet-ISA II controller has circuitry to specifically support this type of access.

The reset register causes a reset when read. Any value will be accepted and the cycle may be 8 or 16 bits wide. Writes are ignored.

All PCnet-ISA II controller register accesses should be coded as 16-bit operations.

"Note that the RAP is cleared on Reset."

IEEE Address Access

The address PROM may be an external memory device that contains the node's unique physical Ethernet address and any other data stored by the board manufacturer. The software accesses must be 16-bit. This information may be stored in the EEPROM.

Boot PROM Access

The boot PROM is an external memory resource located by the address selected by the EEPROM or the BPAM input in **slave** mode. It may be software accessed as an 8-bit or 16-bit resource but the latter is recommended for best performance.

Static RAM Access

The static RAM is only present in the Bus Slave mode. In the Bus Slave mode, two SRAM access schemes are available. When the Shared Memory architecture mode is selected, the SRAM is accessed using ISA memory cycles to the address range selected by the SMAM input. It may be accessed as an 8 or 16-bit resource but the latter is recommended for best performance. When the Programmed I/O architecture mode is selected, the SRAM is accessed through ISACSR0 and ISACSR1 using the RAP and IDP.

Bus Cycles (Hardware)

The PCnet-ISA II controller supports both 8-bit and 16-bit hardware bus cycles. The following sections outline where any limitations apply based upon the architecture mode and/or the resource that is being accessed (PCnet-ISA II controller registers, address PROM, boot PROM, or shared memory SRAM). For completeness, the following sections are arranged by architecture (Bus Master Mode or Bus Slave Mode). SRAM resources apply only to Bus Slave Mode.

All resources (registers, PROMs, SRAM) are presented to the ISA bus by the PCnet-ISA II controller. With few exceptions, these resources can be configured for either 8-bit or 16-bit bus cycles. The I/O resources (registers, address PROM) are width configured using the EEPROM. The memory resources (boot PROM, SRAM) are width configured by external hardware.

For 16-bit memory accesses, hardware external to the PCnet-ISA II controller asserts MEMCS16 when either of the two memory resources is selected. The ISA bus requires that all memory resources within a block of 128 Kbytes be the same width, either 8- or 16-bits. The reason for this is that the MEMCS16 signal is generally a decode of the LA17-23 address lines. 16-bit memory capability is desirable since two 8-bit accesses take the same amount of time as four 16-bit accesses.

All accesses to 8-bit resources (which do not return MEMCS16 or IOCS16) use SD0-7. If an odd byte is accessed, the Current Master swap buffer turns on.

During an odd byte read the swap buffer copies the data from SD0-7 to the high byte. During an odd byte write the Current Master swap buffer copies the data from the high byte to SD0-7. The PCnet-ISA II controller can be configured to be an 8-bit I/O resource even in a 16-bit system; this is set by the EEPROM. It is recommended that the PCnet-ISA II controller be configured for 8-bit only I/O bus cycles for maximum compatibility with PC/AT clone motherboards.

When the PCnet-ISA II controller is in an 8-bit system such as a PC/XT, $\overline{SBHE}$ and $\overline{IOCS16}$ must be left unconnected (these signals do not exist in the PC/XT). This will force ALL resources (I/O and memory) to support only 8-bit bus cycles. The PCnet-ISA II controller will function in an 8-bit system only if configured for Bus Slave Mode.

Accesses to 16-bit resources (which do return $\overline{MEMCS16}$ or $\overline{IOCS16}$) use either or both SD0–7 and SD8–15. A word access is indicated by A0=0 and $\overline{SBHE}$=0 and data is transferred on all 16 data lines. An even byte access is indicated by A0=0 and $\overline{SBHE}$=1 and data is transferred on SD0–7. An odd-byte access is indicated by A0=1 and $\overline{SBHE}$=0 and data is transferred on SD8-15. It is illegal to have A0=1 and $\overline{SBHE}$=1 in any bus cycle. The PCnet-ISA II controller returns only $\overline{IOCS16}$; $\overline{MEMCS16}$ must be generated by external hardware if desired. The use of $\overline{MEMCS16}$ applies only to Shared Memory Mode.

The following table describes all possible types of ISA bus accesses, including Permanent Master as Current Master and PCnet-ISA II controller as Current Master. The PCnet-ISA II controller will not work with 8-bit memory while it is Current Master. Any descriptions of 8-bit memory accesses are for when the Permanent Master is Current Master**.**

The two byte columns (D0–7 and D8–15) indicate whether the bus master or slave is driving the byte. $\overline{CS16}$ is a shorthand for $\overline{MEMCS16}$ and $\overline{IOCS16}$.

Bus Master Mode

The PCnet-ISA II controller can be configured as a Bus Master only in systems that support bus mastering. In addition, the system is assumed to support 16-bit memory (DMA) cycles (the PCnet-ISA II controller does not use the $\overline{MEMCS16}$ signal on the ISA bus). This does not preclude the PCnet-ISA II controller from doing 8-bit I/O transfers. The PCnet-ISA II controller will not function as a bus master in 8-bit platforms such as the PC/XT.

Refresh Cycles

Although the PCnet-ISA II controller is neither an originator or a receiver of refresh cycles, it does need to avoid unintentional activity during a refresh cycle in bus master mode. A refresh cycle is performed as follows: First, the $\overline{REF}$ signal goes active. Then a valid refresh address is placed on the address bus. $\overline{MEMR}$ goes active, the refresh is performed, and $\overline{MEMR}$ goes inactive. The refresh address is held for a short time and them goes invalid. Finally, $\overline{REF}$ goes inactive. During a refresh cycle, as indicated by $\overline{REF}$ being active, the PCnet-ISA II controller ignores $\overline{DACK}$ if it goes active until it goes inactive. It is necessary to ignore $\overline{DACK}$ during a refresh because some motherboards generate a false $\overline{DACK}$ at that time.

ISA Bus Accesses

R/W	A0	$\overline{SBHE}$	$\overline{CS16}$	D0–7	D8–15	Comments
RD	0	1	x	Slave	Float	Low byte RD
RD	1	0	1	Slave	**Float**	High byte RD with swap
RD	0	0	1	Slave	Float	16-Bit RD converted to low byte RD
RD	1	0	0	Float	Slave	High byte RD
RD	0	0	0	Slave	Slave	16-Bit RD
WR	0	1	x	Master	Float	Low byte WR
WR	1	0	1	Master	Float	High byte WR with swap
WR	0	0	1	Master	Master	16-Bit WR converted to low byte WR
WR	1	0	0	Float	Master	High byte WR
WR	0	0	0	Master	Master	16-Bit WR

Address PROM Cycles External PROM

The Address PROM is a small (16 bytes) 8-bit PROM connected to the PCnet-ISA II controller Private Data Bus. The PCnet-ISA II controller will support only 8-bit ISA I/O bus cycles for the address PROM; this limitation is transparent to software and does not preclude 16-bit software I/O accesses. An access cycle begins

with the Permanent Master driving AEN LOW, driving the addressess valid, and driving IOR active. The PCnet-ISA II controller detects this combination of signals and arbitrates for the Private Data Bus (PRDB) if necessary. IOCHRDY is driven LOW during accesses to the address PROM.

When the Private Data Bus becomes available, the PCnet-ISA II controller drives APCS active, releases IOCHRDY, turns on the data path from PRD0-7, and enables the SD0-7 drivers (but not SD8-15). During this bus cycle, IOCS16 is not driven active. This condition is maintained until IOR goes inactive, at which time the bus cycle ends. Data is removed from SD0-7 within 30 ns.

Address PROM Cycles Using EEPROM Data

Default mode. In this mode, the IEEE address information is stored not in an external parallel PROM but in the EEPROM along with other configuration information. PCnet-ISA II will respond to I/O reads from the IEEE address (the first 16 bytes of the I/O map) by supplying data from an internal RAM inside PCnet-ISA II. This internal RAM is loaded with the IEEE address at RESET and is write protected.

Ethernet Controller Register Cycles

Ethernet controller registers (RAP, RDP, IDP) are naturally 16-bit resources but can be configured to operate with 8-bit bus cycles provided the proper protocol is followed. This means on a read, the PCnet-ISA II controller will only drive the low byte of the system data bus; if an odd byte is accessed, it will be swapped down. The high byte of the system data bus is never driven by the PCnet-ISA II controller under these conditions. On a write cycle, the even byte is placed in a holding register. An odd byte write is internally swapped up and augmented with the even byte in the holding register to provide an internal 16-bit write. This allows the use of 8-bit I/O bus cycles which are more likely to be compatible with all ISA-compatible clones, but requires that both bytes be written in immediate succession. This is accomplished simply by treating the PCnet-ISA II controller registers as 16-bit software resources. The motherboard will convert the 16-bit accesses done by software into two sequential 8-bit accesses, an even byte access followed immediately by an odd byte access.

An access cycle begins with the Permanent Master driving AEN LOW, driving the address valid, and driving IOR or IOW active. The PCnet-ISA II controller detects this combination of signals and drives IOCHRDY LOW. IOCS16 will also be driven LOW if 16-bit I/O bus cycles are enabled. When the register data is ready, IOCHRDY will be released HIGH. This condition is maintained until IOR or IOW goes inactive, at which time the bus cycle ends.

RESET Cycles

A read to the reset address causes an PCnet-ISA II controller reset. This has the same effect as asserting the RESET pin on the PCnet-ISA+ controller (which happens on system power up or on a hard boot) except that the T-MAU is NOT reset. The T-MAU will retain its link pass/fail state, disregarding the software RESET command. The subsequent write cycle needed in the NE2100 LANCE based family of Ethernet cards is not required but does not have any harmful effects. IOCS16 is not asserted in this cycle.

ISA Configuration Register Cycles

The ISA configuration registers are accessed by placing the address of the desired register into the RAP and reading the IDP. The ISACSR bus cycles are identical to all other PCnet-ISA II controller register bus cycles.

Boot PROM Cycles

The Boot PROM is an 8-bit PROM connected to the PCnet-ISA II controller Private Data Bus (PRDB) and can occupy up to 64K of address space. Since the PCnet-ISA II controller does not generate MEMCS16, only 8-bit ISA memory bus cycles to the boot PROM are supported in Bus Master Mode; this limitation is transparent to software and does not preclude 16-bit software memory accesses. A boot PROM access cycle begins with the Permanent Master driving the addresses valid, REF inactive, and MEMR active. (AEN is not involved in memory cycles). The PCnet-ISA II controller detects this combination of signals, drives IOCHRDY LOW, and reads a byte out of the Boot PROM. The data byte read is driven onto the lower system data bus lines and IOCHRDY is released. This condition is maintained until MEMR goes inactive, at which time the access cycle ends.

The BPCS signal generated by the PCnet-ISA II controller is three 20 MHz clock cycles wide (300 ns). Including delays, the Boot PROM has 275 ns to respond to the BPCS signal from the PCnet-ISA II controller. This signal is intended to be connected to the CS pin on the boot PROM, with the PROM OE pin tied to ground.

Current Master Operation

Current Master operation only occurs in the Bus Master mode. It does not occur in the Bus Slave mode.

There are three phases to the use of the bus by the PCnet-ISA II controller as Current Master, the Obtain Phase, the Access Phase, and the Release Phase.

Obtain Phase

A Master Mode Transfer Cycle begins by asserting DRQ. When the Permanent Master asserts DACK, the PCnet-ISA II controller asserts MASTER, signifying it has taken control of the ISA bus. The Permanent Master tristates the address, command, and data lines

within 60 ns of $\overline{DACK}$ going active. The Permanent Master drives AEN inactive within 71 ns of $\overline{MASTER}$ going active.

Access Phase

The ISA bus requires a wait of at least 125 ns after $\overline{MASTER}$ is asserted before the new master is allowed to drive the address, command, and data lines. The PCnet-ISA II controller will actually wait 3 clock cycles or 150 ns.

The following signals are not driven by the Permanent Master and are simply pulled HIGH: BALE, IOCHRDY, $\overline{IOCS16}$, $\overline{MEMCS16}$, $\overline{SRDY}$. Therefore, the PCnet-ISA II controller assumes the memory which it is accessing is 16 bits wide and can complete an access in the time programmed for the PCnet-ISA II controller $\overline{MEMR}$ and $\overline{MEMW}$ signals. Refer to the ISA Bus Configuration Register description section.

Release Phase

When the PCnet-ISA II controller is finished with the bus, it drives the command lines inactive. 50 ns later, the controller tri-states the command, address, and data lines and drives DRQ inactive. 50 ns later, the controller drives $\overline{MASTER}$ inactive.

The Permanent Master drives AEN active within 71 ns of $\overline{MASTER}$ going inactive. The Permanent Master is allowed to drive the command lines no sooner than 60 ns after $\overline{DACK}$ goes inactive.

Master Mode Memory Read Cycle

After the PCnet-ISA II controller has acquired the ISA bus, it can perform a memory read cycle. All timing is generated relative to the 20 MHz clock (network clock). Since there is no way to tell if memory is 8-bit or 16-bit or when it is ready, the PCnet-ISA II controller by default assumes 16-bit, 1 wait state memory. The wait state assumption is based on the default value in the MSRDA register in ISACSR0.

The cycle begins with SA0-19, $\overline{SBHE}$, and LA17-23 being presented. The ISA bus requires them to be valid for at least 28 ns before a read command and the PCnet-ISA II controller provides one clock or 50 ns of setup time before asserting $\overline{MEMR}$.

The ISA bus requires $\overline{MEMR}$ to be active for at least 219 ns, and the PCnet-ISA II controller provides a default of 5 clocks, or 250 ns, but this can be tuned for faster systems with the Master Mode Read Active (MSRDA) register (see section 2.5.2). Also, if IOCHRDY is driven LOW, the PCnet-ISA II controller will wait. The wait state counter must expire and IOCHRDY must be HIGH for the PCnet-ISA II controller to continue.

The PCnet-ISA II controller then accepts the memory read data. The ISA bus requires all command lines to remain inactive for at least 97 ns before starting another bus cycle and the PCnet-ISA II controller provides at least two clocks or 100 ns of inactive time.

The ISA bus requires read data to be valid no more than 173 ns after receiving $\overline{MEMR}$ active and the PCnet-ISA II controller requires 10 ns of data setup time. The ISA bus requires read data to provide at least 0 ns of hold time and to be removed from the bus within 30 ns after $\overline{MEMR}$ goes inactive. The PCnet-ISA II controller requires 0 ns of data hold time.

Master Mode Memory Write Cycle

After the PCnet-ISA II controller has acquired the ISA bus, it can perform a memory write cycle. All timing is generated relative to a 20 MHz clock which happens to be the same as the network clock. Since there is no way to tell if memory is 8- or 16-bit or when it is ready, the PCnet-ISA II controller by default assumes 16-bit, 1 wait state memory. The wait state assumption is based on the default value in the MSWRA register in ISACSR1.

The cycle begins with SA0-19, $\overline{SBHE}$, and LA17-23 being presented. The ISA bus requires them to be valid at least 28 ns before $\overline{MEMW}$ goes active and data to be valid at least 22 ns before $\overline{MEMW}$ goes active. The PCnet-ISA II controller provides one clock or 50 ns of setup time for all these signals.

The ISA bus requires $\overline{MEMW}$ to be active for at least 219 ns, and the PCnet-ISA II controller provides a default of 5 clocks, or 250 ns, but this can be tuned for faster systems with the Master Mode Write Active (MSWRA) register (ISACSR1). Also, if IOCHRDY is driven LOW, the PCnet-ISA II controller will wait. IOCHRDY must be HIGH for the PCnet-ISA II controller to continue.

The ISA bus requires data to be valid for at least 25 ns after $\overline{MEMW}$ goes inactive, and the PCnet-ISA II controller provides one clock or 50 ns.

The ISA bus requires all command lines to remain inactive for at least 97 ns before starting another bus cycle. The PCnet-ISA II controller provides at least two clocks or 100 ns of inactive time when bit 4 in ISACSR2 is set. The EISA bus requires all command lines to remain inactive for at least 170 ns before starting another bus cycle. When bit 4 in ISACSR4 is cleared, the PCnet-ISA II controller provides 200 ns of inactive time.

Back-to-Back DMA Requests

The PCnet-ISA II provides for fair bus bandwidth sharing between two bus mastering devices on the ISA bus through an adaptive delay which is inserted between back-to-back DMA requests.

When the PCnet-ISA II requires bus access immediately following a bus ownership period, it first checks the status of the three currently unused DRQ pins. If a

lower priority DRQ pin than the one currently being used by the PCnet-ISA II is asserted, the PCnet-ISA II will wait 2.6 µs after the deassertion of $\overline{DACK}$ before re-asserting its DRQ pin. If no lower priority DRQ pin is asserted, the PCnet-ISA II may re-assert its DRQ pin after as short as 1.1 µs following $\overline{DACK}$ deassertion. The priorities assumed by the PCnet-ISA II are ordered DRQ3, DRQ5, DRQ6, DRQ7, with DRQ3 having highest priority and DRQ7 having the lowest priority. This priority ordering matches that used by typical ISA bus DMA controllers.

This adaptive delay scheme allows for fair bus bandwidth sharing when two bus mastering devices, e.g. two PCnet-ISA II devices, are on an ISA bus. The controller using the higher priority DMA channel cannot lock out the controller using the lower priority DMA channel because of the 2.6 µs delay that is inserted before DRQ reassertion when a lower priority DRQ pin is asserted. When there is no lower priority DMA request asserted, the PCnet-ISA II re-requests the bus immediately, providing optimal performance when there is no competition for bus access.

Bus Slave Mode

The PCnet-ISA II can be configured to be a bus slave for systems that do not support bus mastering or require a local memory to tolerate high bus latencies. In the Bus Slave mode, the I/O map of the PCnet-ISA II is identical to the I/O map when in the Bus Master mode (see I/O Resources section). Hence, the address PROM, controller registers, and Reset port are accessed through I/O cycles on the ISA bus. However, the initialization block, descriptor rings, and buffers, which are located in system memory when in the Bus Master mode, are located in a local SRAM when in the Bus Slave mode. The local SRAM can be accessed by memory cycles on the ISA bus (Shared Memory architecture) or by I/O cycles on the ISA bus (Programmed I/O mode).

Address PROM Cycles External PROM

The Address PROM is a small (16 bytes) 8-bit PROM connected to the PCnet-ISA II controller Private Data Bus (PRDB). The PCnet-ISA II controller will support only 8-bit ISA I/O bus cycles for the address PROM; this limitation is transparent to software and does not preclude 16-bit software I/O accesses. An access cycle begins with the Permanent Master driving AEN LOW, driving the addresses valid, and driving $\overline{IOR}$ active. The PCnet-ISA II controller detects this combination of signals and arbitrates for the Private Data Bus if necessary. IOCHRDY is always driven LOW during address PROM accesses.

When the Private Data Bus becomes available, the PCnet-ISA II controller drives $\overline{APCS}$ active, releases IOCHRDY, turns on the data path from PRD0-7, and enables the SD0-7 drivers (but not SD8-15). During this bus cycle, $\overline{IOCS16}$ is not driven active. This condition is maintained until IOR goes inactive, at which time the access cycle ends. Data is removed from SD0-7 within 30 ns.

The PCnet-ISA II controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if $\overline{SBHE}$ has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

Ethernet Controller Register Cycles

Ethernet controller registers (RAP, RDP, ISACSR) are naturally 16-bit resources but can be configured to operate with 8-bit bus cycles provided the proper protocol is followed. This is programmable by the EEPROM. This means on a read, the PCnet-ISA II controller will only drive the low byte of the system data bus; if an odd byte is accessed, it will be swapped down. The high byte of the system data bus is never driven by the PCnet-ISA II controller under these conditions. On a write, the even byte is placed in a holding register. An odd-byte write is internally swapped up and augmented with the even byte in the holding register to provide an internal 16-bit write. This allows the use of 8-bit I/O bus cycles which are more likely to be compatible with all clones, but requires that both bytes be written in immediate succession. This is accomplished simply by treating the PCnet-ISA II controller controller registers as 16-bit software resources. The motherboard will convert the 16-bit accesses done by software into two sequential 8-bit accesses, an even-byte access followed immediately by an odd-byte access.

An access cycle begins with the Permanent Master driving AEN LOW, driving the address valid, and driving $\overline{IOR}$ or $\overline{IOW}$ active. The PCnet-ISA II controller detects this combination of signals and drives IOCHRDY LOW. $\overline{IOCS16}$ will also be driven LOW if 16-bit I/O bus cycles are enabled. When the register data is ready, IOCHRDY will be released HIGH. This condition is maintained until $\overline{IOR}$ or $\overline{IOW}$ goes inactive, at which time the bus cycle ends.

The PCnet-ISA II controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if $\overline{SBHE}$ has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

RESET Cycles

A read to the reset address causes an PCnet-ISA II controller reset. This has the same effect as asserting the RESET pin on the PCnet-ISA$^+$ controller (which happens on system power up or on a hard boot) except that the T-MAU is NOT reset. The T-MAU will retain its link pass/fail state, disregarding the software RESET command. The subsequent write cycle needed in the NE2100 LANCE- based family of Ethernet cards is not required but does not have any harmful effects. $\overline{IOCS16}$ is not asserted in this cycle.

ISA Configuration Register Cycles

The ISA configuration register is accessed by placing the address of the desired register into the RAP and reading the IDP. The ISACSR bus cycles are identical to all other PCnet-ISA II controller register bus cycles.

Boot PROM Cycles

The Boot PROM is an 8-bit PROM connected to the PCnet-ISA II controller Private Data Bus (PRDB), and can occupy up to 64 Kbytes of address space. In Shared Memory Mode, an external address comparator is responsible for asserting $\overline{BPAM}$ to the PCnet-ISA II controller. $\overline{BPAM}$ is intended to be a perfect decode of the boot PROM address space, i.e. LA17-23, SA16. The LA bus must be latched with BALE in order to provide stable signal for $\overline{BPAM}$. $\overline{REF}$ inactive must be used by the external logic to gate boot PROM address decoding. This same logic must assert $\overline{MEMCS16}$ to the ISA bus if 16-bit Boot PROM bus cycles are desired.

In the Bus Slave mode, boot PROM cycles can be programmed to be 8 or 16-bit ISA memory cycles with the BP_16B bit (PnP 0x42). If the BP_16B bit is set, the PCnet-ISA II assumes 16-bit ISA memory cycles for the boot PROM. In this case, the external hardware responsible for generating $\overline{BPAM}$ must also generate $\overline{MEMCS16}$. A 16-bit boot PROM bus cycle begins with the Permanent Master driving the addresses valid and $\overline{MEMR}$ active. (AEN is not involved in memory cycles). External hardware would assert $\overline{BPAM}$ and $\overline{MEMCS16}$. The PCnet-ISA II controller detects this combination of signals, drives IOCHRDY LOW, and reads two bytes out of the boot PROM. The data bytes read from the PROM are driven by the PCnet-ISA II controller onto SD0-15 and IOCHRDY is released. This condition is maintained until $\overline{MEMR}$ goes inactive, at which time the access cycle ends.

The PCnet-ISA II controller will perform 8-bit ISA bus cycle operation for all resource (registers, PROMs, SRAM) if $\overline{SBHE}$ has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

The $\overline{BPCS}$ signal generated by the PCnet-ISA II controller is three 20 MHz clock cycles wide (350 ns). Including delays, the Boot PROM has 275 ns to respond to the $\overline{BPCS}$ signal from the PCnet-ISA II controller. This signal is intended to be connected to the $\overline{CS}$ pin on the boot PROM, with the PROM $\overline{OE}$ pin tied to ground.

Static RAM Cycles – Shared Memory Architecture

In the Shared Memory Architecture mode, the SRAM is an 8-bit device connected to the PCnet-ISA II controller Private Bus, and can occupy up to 64 Kbytes of address space. The SRAM is memory mapped into the ISA memory space at an address range determined by external decode logic. The external address comparator is responsible for asserting $\overline{SMAM}$ to the PCnet-ISA II controller. $\overline{SMAM}$ is intended to be a perfect decode of the SRAM address space, i.e. LA17-23, SA16 for 64 Kbytes of SRAM. The LA signals must be latched by BALE in order to provide a stable decode for $\overline{SMAM}$. The PCnet-ISA II controller assumes 16-bit ISA memory bus cycles for the SRAM, so this same logic must assert $\overline{MEMCS16}$ to the ISA bus if 16-bit bus cycles are to be supported.

A 16-bit SRAM bus cycle begins with the Permanent Master driving the addresses valid, $\overline{REF}$ inactive, and either $\overline{MEMR}$ or $\overline{MEMW}$ active. (AEN is not involved in memory cycles). External hardware would assert $\overline{SMAM}$ and $\overline{MEMCS16}$. The PCnet-ISA II controller detects this combination of signals and initiates the SRAM access.

In a write cycle, the PCnet-ISA II controller stores the data into an internal holding register, allowing the ISA bus cycle to finish normally. The data in the holding register will then be written to the SRAM without the need for ISA bus control. In the event the holding register is already filled with unwritten SRAM data, the PCnet-ISA II controller will extend the ISA write cycle by driving IOCHRDY LOW until the unwritten data is stored in the SRAM. The current ISA bus cycle will then complete normally.

In a read cycle, the PCnet-ISA II controller arbitrates for the Private Bus. If it is unavailable, the PCnet-ISA II controller drives IOCHRDY LOW. The PCnet-ISA II controller compares the 16 bits of address on the System Address Bus with that of a data word held in an internal pre-fetch register.

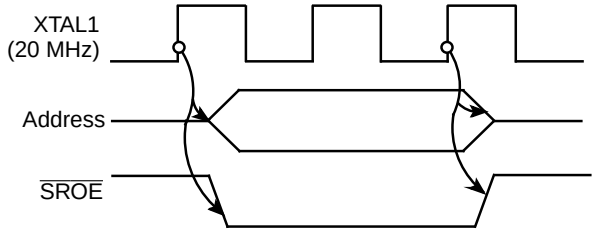
If the address does not match that of the prefetched SRAM data, then the PCnet-ISA II controller drives IOCHRDY LOW and reads two bytes from the SRAM. The PCnet-ISA II controller then proceeds as though the addressed data location had been prefetched.

If the internal prefetch buffer contains the correct data, then the pre-fetch buffer data is driven on the System Data bus. If IOCHRDY was previously driven LOW due to either Private Data Bus arbitration or SRAM access, then it is released HIGH. The PCnet-ISA II controller remains in this state until $\overline{MEMR}$ is de-asserted, at which time the PCnet-ISA II controller performs a new prefetch of the SRAM. In this way memory read wait states can be minimized.

The PCnet-ISA II controller performs prefetches of the SRAM between ISA bus cycles. The SRAM is prefetched in an incrementing word address fashion. Prefetched data are invalidated by any other activity on the Private Bus, including Shared Memory Writes by either the ISA bus or the network interface, and also address and boot PROM reads.

The only way to configure the PCnet-ISA II controller for 8-bit ISA bus cycles for SRAM accesses is to configure the entire PCnet-ISA II controller to support only 8-bit ISA bus cycles. This is accomplished by leaving the $\overline{SBHE}$ pin disconnected. The PCnet-ISA II controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if $\overline{SBHE}$ has never been driven active since the last RESET, such as in the case of an 8-bit system like the PC/XT. In this case, the external address decode logic must not assert $\overline{MEMCS16}$ to the ISA bus, which will be the case if $\overline{MEMCS16}$ is left unconnected. It is possible to manufacture a dual 8/16 bit PCnet-ISA II controller adapter card, as the $\overline{MEMCS16}$ and $\overline{SBHE}$ signals do not exist in the PC/XT environment.

At the memory device level, each SRAM Private Bus read cycle takes two 50 ns clock periods for a maximum read access time of 75 ns. The timing looks like this:

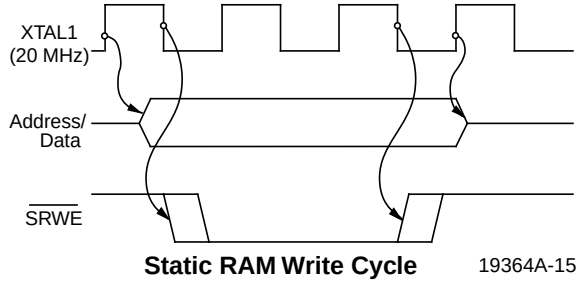


19364A-14

Static RAM Read Cycle

The address and $\overline{SROE}$ go active within 20 ns of the clock going HIGH. Data is required to be valid 5 ns before the end of the second clock cycle. Address and $\overline{SROE}$ have a 0 ns hold time after the end of the second clock cycle. Note that the PCnet-ISA II controller does not normally provide a separate SRAM $\overline{CS}$ signal; SRAM $\overline{CS}$ must always be asserted.

SRAM Private Bus write cycles require three 50 ns clock periods to guarantee non-negative address setup and hold times with regard to $\overline{SRWE}$. The timing is illustrated as follows:



Static RAM Write Cycle 19364A-15

Address and data are valid 20 ns after the rising edge of the first clock period. $\overline{SRWE}$ goes active 20 ns after the falling edge of the first clock period. $\overline{SRWE}$ goes inactive 20 ns after the falling edge of the third clock period. Address and data remain valid until the end of the third clock period. Rise and fall times are nominally 5 ns. Non-negative setup and hold times for address and data with respect to $\overline{SRWE}$ are guaranteed. $\overline{SRWE}$ has a pulse width of typically 100 ns, minimum 75 ns.

Static RAM Cycles – Programmed I/O Architecture

In the Programmed I/O Architecture mode, the SRAM is an 8-bit device connected to the PCnet-ISA II controller Private Bus, and can occupy up to 64 Kbytes of address space. The SRAM is accessed through the ISACSR0 and ISACSR1 registers which serve as the SRAM Data port and SRAM Address pointer, respectively. Since the ISACSRs are used to access the SRAM, simple I/O accesses (to RAP and IDP) which are decoded by the PCnet-ISA II are used to access the SRAM without any external decoding logic.

The RAP and IDP ports are naturally 16-bit resources and can be accessed with 16-bit ISA I/O cycles if the IO_MODE bit (PnP 0xF0) is set. As discussed in the Ethernet Controller Register Cycles section, 8-bit I/O cycles are also allowed, provided the proper protocol is followed. This protocol requires that byte accesses must be performed in pairs, with the even byte access always being followed by associated odd byte access. In the Programmed I/O architecture mode, when accessing the SRAM Data Port in particular (ISACSR0), the restrictions on byte accesses are slightly different. Even byte accesses (accesses where A0 = 0, $\overline{SBHE}$ = 1) may be performed to ISACSR0 without any restriction. A corresponding odd byte access need not be performed following the even byte access as is required when accessing all other controller registers. In fact, odd byte accesses (accesses where A0 = 1, $\overline{SBHE}$ = 1) may not be performed to ISACSR0, except when they are the result of a software 16-bit access that are automatically converted to two byte accesses by motherboard logic.

Since the internal PCnet-ISA II registers are used to access the SRAM in the Programmed I/O architecture mode, the access cycle on the ISA bus is identical to that described in the Ethernet Controller Register Cycles section.

To minimize the number of I/O cycles required to access the SRAM, the PCnet-ISA II auto-increments the SRAM Address Pointer (ISACSR1) by one or two following every read or write to the SRAM Data Port (ISACSR0). If a single byte read or write to the SRAM Data Port occurs, the SRAM Address Pointer is automatically incremented by 1. If a word read or write to the SRAM Data Port occurs, the SRAM Address Pointer is automatically incremented by 2. This allows

reads and writes to adjacent ascending addresses in the SRAM to be performed without intervening writes to the SRAM Address Pointer. Since buffer accesses comprise a high percentage of all accesses to the SRAM, and buffer accesses are typically performed in adjacent ascending order, the auto-increment of the SRAM Address Pointer reduces the required ISA bus cycles significantly.

In addition to the auto-incrementing of the SRAM Address pointer, the PCnet-ISA II performs write posting on writes to the SRAM and read prefetching on reads from the SRAM to maximize performance in the Programmed I/O architecture mode.

Write Posting: When a write cycle to the SRAM Data Port occurs, the PCnet-ISA II controller stores the data into an internal holding register, allowing the ISA bus cycle to finish normally. The data in the holding register will then be written to the SRAM without the need for ISA bus control. In the event that the holding register is already filled with unwritten SRAM data, the PCnet-ISA II controller will extend the ISA write cycle by driving OCHRDY LOW until the unwritten data is stored in the SRAM. Once the data is written into the SRAM, the new write data is stored into the internal holding register and IOCHRDY is released allowing the ISA bus cycle to complete.

Read Prefetching: To gain performance on read accesses to the SRAM, the PCnet-ISA II performs prefetches of the SRAM after every read from the SRAM Data Port. The prefetch is performed using the speculated address that results from the auto-increment that occurs on the SRAM Address Pointer following every access to the SRAM Data Port. Following every read access, the 16-bit word following the just-read SRAM byte or word is prefetched and placed in a holding register. If a **word** read from the SRAM Data Port occurs before a “prefetch invalidation event” occurs, the prefetched word is driven onto the SD[15:0] pins without a wait state (no IOCHRDY LOW assertion). A “prefetch invalidation event” is defined as any activity on the Private Bus other than SRAM reads. This includes SRAM writes by either the ISA bus or the network interface, address or boot PROM reads, or any write to the SRAM Address Pointer.

The PCnet-ISA II interface to the SRAM in the Programmed I/O architecture mode is identical to that in the Shared Memory Architecture mode. Hence, the SRAM Read and Write cycle descriptions and diagrams shown in the “Static RAM Cycles – Shared Memory Architecture” section apply.

Transmit Operation

The transmit operation and features of the PCnet-ISA II controller are controlled by programmable options.

Transmit Function Programming

Automatic transmit features, such as retry on collision, FCS generation/transmission, and pad field insertion, can all be programmed to provide flexibility in the (re-)transmission of messages.

Disable retry on collision (DRTY) is controlled by the DRTY bit of the Mode register (CSR15) in the initialization block.

Automatic pad field insertion is controlled by the APAD_XMT bit in CSR4. If APAD_XMT is set, automatic pad field insertion is enabled, the DXMTFCS feature is over-ridden, and the 4-byte FCS will be added to the transmitted frame unconditionally. If APAD_XMT is cleared, no pad field insertion will take place and runt packet transmission is possible.

The disable FCS generation/transmission feature can be programmed dynamically on a frame by frame basis. See the ADD_FCS description of TMD1.

Transmit FIFO Watermark (XMTFW in CSR80) sets the point at which the BMU (Buffer Management Unit) requests more data from the transmit buffers for the FIFO. This point is based upon how many 16-bit bus transfers (2 bytes) could be performed to the existing empty space in the transmit FIFO.

Transmit Start Point (XMTSP in CSR80) sets the point when the transmitter actually tries to go out on the media. This point is based upon the number of bytes written to the transmit FIFO for the current frame.

When the entire frame is in the FIFO, attempts at transmission of preamble will commence regardless of the value in XMTSP. The default value of XMTSP is 10b, meaning 64 bytes full.

Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (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/controlling process. Setting the APAD_XMT bit in CSR4 enables the automatic padding feature. The pad is placed between the LLC data field and FCS field in the 802.3 frame. FCS is always added if the frame is padded, regardless of the state of DXMTFCS. The transmit frame will be padded by bytes with the value of 00h. The default value of APAD_XMT is 0, and this will disable auto pad generation after RESET.

It is the responsibility of upper layer software to 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 II controller to compute the actual number of pad bytes

to be inserted. The PCnet-ISA II controller will append pad bytes dependent on the actual number of bits transmitted onto the network. Once the last data byte of the frame has completed prior to appending the FCS, the PCnet-ISA II controller will check 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:

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:

Preamble + (Min Frame Size + FCS) bits

At the point that FCS is to be appended, the transmitted frame should contain:

Preamble + (Min Frame Size - FCS) bits
64+ (512- 32) bits

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

Transmit FCS Generation

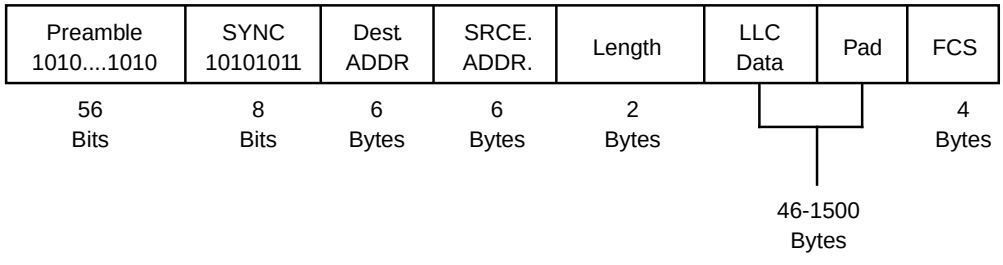
Automatic generation and transmission of FCS for a transmit frame depends on the value of DXMTFCS bit in CSR15. When DXMTFCS = 0 the transmitter will generate and append the FCS to the transmitted frame. If the automatic padding feature is invoked (APAD_XMT is SET in CSR4), the FCS will be appended by the PCnet-ISA II controller regardless of the state of DXMTFCS. Note that the calculated FCS is transmitted most-significant bit first. The default value of DXMTFCS is 0 after RESET.

Transmit Exception Conditions

Exception conditions for frame transmission fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-ISA II controller are basically collisions within the slot time with automatic retry. The PCnet-ISA II controller will ensure that collisions which occur within 512 bit times from the start of transmission (including preamble) will be automatically retried with no host intervention. The transmit FIFO ensures this by guaranteeing that data contained within the FIFO will not be overwritten until at least 64 bytes (512 bits) of data have been successfully transmitted onto the network.

If 16 total attempts (initial attempt plus 15 retries) fail, the PCnet-ISA II controller sets the RTRY bit in the current transmit TDTE in host memory (TMD2), gives up ownership (sets the OWN bit to zero) for this packet, and processes the next packet in the transmit ring for transmission.



19364A-16

ISO 8802-3 (IEEE/ANSI 802.3) Data Frame

Abnormal network conditions include:

- Loss of carrier
- Late collision
- SQE Test Error (Does not apply to 10BASE-T port.)

These should not occur on a correctly configured 802.3 network, and will be reported if they do.

When an error occurs in the middle of a multi-buffer frame transmission, the error status will be written in the current descriptor. The OWN bit(s) in the subsequent descriptor(s) will be reset until the STP (the next frame) is found.

Loss of Carrier

A loss of carrier condition will be reported if the PCnet-ISA II controller cannot observe receive activity while it is transmitting on the AUI port. After the PCnet-ISA II controller initiates a transmission, it will expect to see data "looped back" on the DI± pair. This will internally generate a "carrier sense," indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted before the end of the transmission. If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in TMD2 after the frame has been transmitted. The frame will not be re-tried on the basis of an LCAR error. In 10BASE-T mode LCAR will indicate that Jabber or Link Fail state has occurred.

Late Collision

A late collision will be reported if a collision condition occurs after one slot time (512 bit times) after the transmit process was initiated (first bit of preamble commenced). The PCnet-ISA II controller will abandon the transmit process for the particular frame, set Late Collision (LCOL) in the associated TMD3, and process the next transmit frame in the ring. Frames experiencing a late collision will not be re-tried. Recovery from this condition must be performed by upper-layer software.

SQE Test Error

During the inter packet gap time following the completion of a transmitted message, the AUI CI± pair is asserted by some transceivers as a self-test. The integral Manchester Encoder/Decoder will expect the SQE Test Message (nominal 10 MHz sequence) to be returned via the CI± pair within a 40 network bit time period after DI± pair goes inactive. If the CI± inputs are not asserted within the 40 network bit time period following the completion of transmission, then the PCnet-ISA II controller will set the CERR bit in CSR0. CERR will be asserted in 10BASE-T mode after transmit if T-MAU is in Link Fail state. CERR will never cause INTR to be activated. It will, however, set the ERR bit in CSR0.

Host related transmit exception conditions include BUFF and UFLO as described in the Transmit Descriptor section.

Receive Operation

The receive operation and features of the PCnet-ISA II controller are controlled by programmable options.

Receive Function Programming

Automatic pad field stripping is enabled by setting the ASTRP_RCV bit in CSR4; this can provide flexibility in the reception of messages using the 802.3 frame format.

All receive frames can be accepted by setting the PROM bit in CSR15. When PROM is set, the PCnet-ISA II controller will attempt to receive all messages, subject to minimum frame enforcement. Promiscuous mode overrides the effect of the Disable Receive Broadcast bit on receiving broadcast frames.

The point at which the BMU will start to transfer data from the receive FIFO to buffer memory is controlled by the RCVFW bits in CSR80. The default established during reset is 10b, which sets the threshold flag at 64 bytes empty.

Automatic Pad Stripping

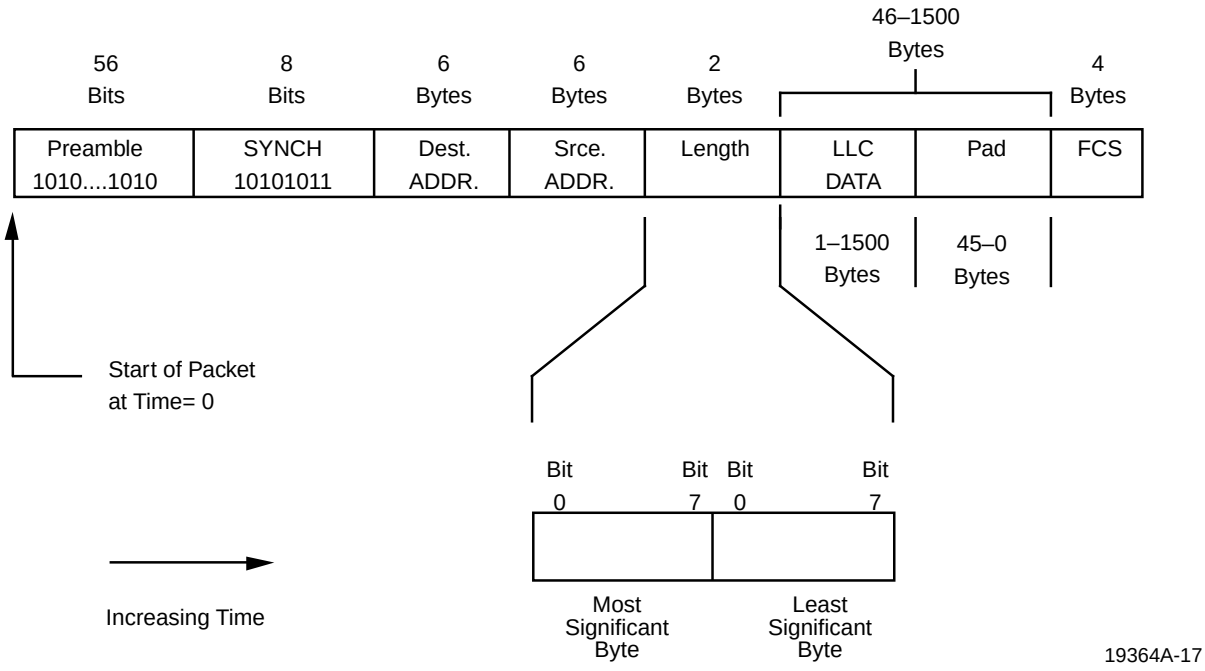
During reception of an 802.3 frame the pad field can be stripped automatically. ASTRP_RCV (bit 10 in CSR4) = 1 enables the automatic pad stripping feature. The pad field will be stripped before the frame is passed to the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since 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 had 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 frame. The length indicates the actual number of LLC data bytes contained in the message. Any received frame which contains a length field less than 46 bytes will have the pad field stripped (if ASTRP_RCV is set). Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified.

Since any valid Ethernet Type field value will always be greater than a normal 802.3 Length field (≥46), the PCnet-ISA II controller will not attempt to strip valid Ethernet frames.

Note that for some network protocols the value passed in the Ethernet Type and/or 802.3 Length field is not compliant with either standard and may cause problems.

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



IEEE/ANSI 802.3 Frame and Length Field Transmission Order

Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the PCnet-ISA II controller. Note that if the Automatic Pad Stripping feature is enabled, the received FCS will be verified against the value computed for the incoming bit stream including pad characters, but it will not be passed to the host. If a FCS error is detected, this will be reported by the CRC bit in RMD1.

Receive Exception Conditions

Exception conditions for frame reception fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-ISA II controller are basically collisions within the slot time and automatic runt packet rejection. The PCnet-ISA II controller will ensure that collisions which occur within 512 bit times from the start of reception (excluding preamble) will be automatically deleted from the receive FIFO with no host intervention. The receive FIFO will delete any frame which is composed of fewer than 64 bytes provided that the Runt Packet Accept (RPA bit in CSR124) feature has not been enabled. This criteria will be met regardless of whether the receive frame was the first (or only) frame in the FIFO or if the receive frame was queued behind a previously received message.

Abnormal network conditions include:

- FCS errors
- Late collision

These should not occur on a correctly configured 802.3 network and will be reported if they do.

Host related receive exception conditions include MISS, BUFF, and OFLO. These are described in the Receive Descriptor section.

MAGIC PACKET™ OPERATION

In the Magic Packet mode, PCnet-ISA II completes any transmit and receive operations in progress, suspends normal activity, and enters into a state where only a Magic Packet could be detected. A Magic Packet frame is a frame that contains a data sequence which repeats the Physical Address (PADR[47:00]) at least sixteen times frame sequentially, with bit[00] received first. In Magic Packet suspend mode, the PCnet-ISA II remains powered up. Slave accesses to the PCnet-ISA II are still possible, the same as any other mode. All of the received packets are flushed from the receive FIFO. An LED and/or interrupt pin could be activated, indicating the receive of a Magic Packet frame. This indication could be used for a variety of management tasks.

Magic Packet Mode Activation

This mode can be enabled by either software or external hardware means, but in either case, the MP_MODE bit (CSR5, bit 1) must be set first.

Hardware Activation. This is done by driving the $\overline{\text{SLEEP}}$ pin low. Deasserting the $\overline{\text{SLEEP}}$ pin will return the PCnet-ISA II to normal operation.

Software Activation. This is done by setting the MP_ENBL bit (CSR5, bit 2). Resetting this bit will return the PCnet-ISA II to normal operation.

Magic Packet Receive Indicators

The reception of a Magic Packet can be indicated either through one of the LEDs 1, 2 or 3, and/or the activation of the interrupt pin. MP_INT bit (CSR5, bit 4) will also be set upon the receive of the Magic Packet.

LED Indication. Either one of the LEDs 1, 2, or 3 could be activated by the receive of the Magic Packet. The "Magic Packet enable" bit (bit 9) in the ISACSR 5, 6 or 7 should be set to enable this feature. Note that the polarity of the LED2 could be controlled by the LEDXOR bit (ISACSR6, bit 14). The LED could be deactivated by setting the STOP bit or resetting the MP_ENBL bit (CSR5, bit 2).

Interrupt Indication. Interrupt pin could be activated by the receive of the Magic Packet. The MP_I_ENBL bit (CSR5, bit 3) and IENA bit (CSR0, bit 6) should be set to enable this feature.

Loopback Operation

Loopback is a mode of operation intended for system diagnostics. In this mode, the transmitter and receiver are both operating at the same time so that the controller receives its own transmissions. The controller provides two types of internal loopback and three types of external loopback. In internal loopback mode, the transmitted data can be looped back to the receiver at one of two places inside the controller without actually transmitting any data to the external network. The receiver will move the received data to the next receive buffer, where it can be examined by software. Alternatively, external loopback causes transmissions to go off-chip. For the AUI port, frame transmission occurs normally and assumes that an external MAU will loop the frame back to the chip. For the 10BASE-T port, two external loopback options are available, both of which require a valid link pass state and both of which transmit data frames at the RJ45 interface. Selection of these modes is defined by the TMAU_LOOPE bit in ISACSR2. One option loops the data frame back inside the chip, and is compatible with a 'live' network. The other option requires an external device (such as a 'loopback plug') to loop the data back to the chip, a function normally not available on a 10BASE-T network.

The PCnet-ISA II chip has two dedicated FCS generators, eliminating the traditional LANCE limitations on loopback FCS operation. The receive FCS generation logic is always enabled. The transmit FCS generation logic can be disabled (to emulate LANCE type loopback operation) by setting the DXMTFCS bit in the Mode register (CSR15). In this configuration, software must generate the FCS and append the four FCS bytes to the transmit frame data.

The loopback facilities of the MAC Engine allow full operation to be verified without disturbance to the network. Loopback operation is also affected by the state of the Loopback Control bits (LOOP, MENDECL, and INTL) in CSR15. This affects whether the internal MENDEC is considered part of the internal or external loop- backpath.

The receive FCS generation logic in the PCnet-ISA II chip is used for multicast address detection. Since this FCS logic is always enabled, there are no restrictions to the use of multicast addressing while in loopback mode.

When performing an internal loopback, no frame will be transmitted to the network. However, when the PCnet-ISA II controller is configured for internal loopback the receiver will not be able to detect network traffic. External loopback tests will transmit frames onto the network if the AUI port is selected, and the PCnet-ISA II controller will receive network traffic while configured for external loopback when the AUI port is selected. Runt Packet Accept is automatically enabled when any loopback mode is invoked.

Loopback mode can be performed with any frame size. Runt Packet Accept is internally enabled (RPA bit in CSR124 is not affected) when any loopback mode is invoked. This is to be backwards compatible to the LANCE (Am7990) software.

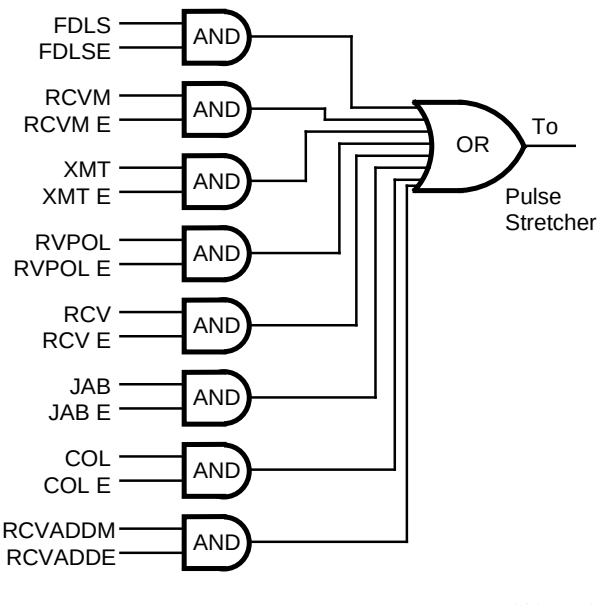
LEDs

The PCnet-ISA II controller's LED control logic allows programming of the status signals, which are displayed

on 3 LED outputs. One LED ($\overline{LED0}$) is dedicated to displaying 10BASE-T Link Status. The status signals available are Collision, Jabber, Receive, Receive Polarity, Transmit, Receive Address Match, and Full Duplex Link Status. If more than one status signal is enabled, they are ORed together. An optional pulse stretcher is available for each programmable output. This allows emulation of the TPEX (Am79C98) and TPEX+ (Am79C100) LED outputs.

Signal	Behavior
COL	Active during collision activity on the network
FDLS	Active when Full Duplex operation is enabled and functioning on the selected network port
JAB	Active when the PCnet-ISA II is jabbering on the network
LNKST	Active during Link OK Not active during Link Down
RCV	Active while receiving data
RVPOL	Active during receive polarity is OK Not active during reverse receive polarity
RCVADDM	Active during Receive with Address Match
XMT	Active while transmitting data

Each status signal is ANDed with its corresponding enable signal. The enabled status signals run to a common OR gate:



LED Control Logic

The output from the OR gate is run through a pulse stretcher, which consists of a 3-bit shift register clocked at 38 Hz. The data input of the shift register is at logic 0. The OR gate output asynchronously sets all three bits of the shift register when its output goes active. The output of the shift register controls the associated $\overline{LEDx}$ pin. Thus, the pulse stretcher provides an LED output of 52 ms to 78 ms.

Refer to the section “ISA Bus Configuration Registers” for information on LED control via the ISACSRs.