Am79C970A

PCnet™-PCI II Single-Chip Full-Duplex Ethernet Controller for PCI Local Bus Product

DISTINCTIVE CHARACTERISTICS

- **Single-chip Ethernet controller for the Peripheral Component Interconnect (PCI) local bus**
- **Supports ISO 8802-3 (IEEE/ANSI 802.3) and Ethernet standards**
- **Direct interface to the PCI local bus (Revision 2.0 compliant)**
- **High-performance 32-bit Bus Master architecture with integrated DMA buffer management unit for low CPU and bus utilization**
- **Software-compatible with AMD PCnet family, LANCE/C-LANCE, and Am79C900 ILACC register and descriptor architecture**
- **Compatible with PCnet family driver software**
- **Full-duplex operation for increased network bandwidth**
- **Big endian and little endian byte alignments supported**
- **3.3 V/5 V signaling for PCI bus interface**
- **Low-power CMOS design with two sleep modes allows reduced power consumption for critical battery-powered applications and Green PCs**
- **Integrated Magic Packet™ support for remote wake up of Green PCs**
- **Individual 272-byte transmit and 256-byte receive FIFOs provide frame 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 frame rejection
  - — Automatic selection of received collision frames
- **Microwire EEPROM interface supports jumperless design and provides through-chip programming**
- **Supports optional Boot PROM for diskless node applications**
- **Look-Ahead Packet Processing (LAPP) data handling technique reduces system overhead by allowing protocol analysis to begin before end of receive frame**
- **Integrated Manchester encoder/decoder**
- **Provides integrated attachment unit interface (AUI) and 10BASE-T transceiver with automatic port selection**
- **Automatic twisted-pair receive polarity detection and automatic correction of the receive polarity**
- **Optional byte padding to long-word boundary on receive**
- **Dynamic transmit FCS generation programmable on a frame-by-frame basis**
- **Internal/external loopback capabilities**
- **Supports the following types of network interfaces:**
  - — AUI to external 10BASE-2, 10BASE-5, 10BASE-T, or 10BASE-F MAU
  - — Internal 10BASE-T transceiver with Smart Squelch to twisted-pair medium
- **JTAG Boundary Scan (IEEE 1149.1) test access port interface and NAND Tree test mode for board-level production connectivity test**
- **Supports external address detection interface (EADI)**
- **4 programmable LEDs for status indication**
- **132-pin PQFP and 144-pin TQFP packages**
- **Support for operation in industrial temperature range (–40°C to +85°C) available in both packages**

GENERAL DESCRIPTION

The 32-bit PCnet-PCI II single-chip full-duplex Ethernet controller is a highly integrated Ethernet system solution designed to address high-performance system application requirements. It is a flexible bus-mastering device that can be used in any application, including network-ready PCs, printers, fax modems, and bridge/router designs. The bus-master architecture provides high data throughput in the system and low

Publication# **19436** Rev: **E** Amendment/**0**
Issue Date: **June 2000**

CPU and system bus utilization. The PCnet-PCI II controller is fabricated with AMD's advanced low-power CMOS process to provide low operating and standby current for power-sensitive applications.

The PCnet-PCI II controller is a complete Ethernet node integrated into a single VLSI device. It contains a bus interface unit, a DMA buffer management unit, an IEEE 802.3-compliant media access control (MAC) function, individual 272-byte transmit and 256-byte receive FIFOs, an IEEE 802.3-compliant attachment unit interface (AUI) and twisted-pair transceiver medium attachment unit (10BASE-T MAU) that can both operate in either half-duplex or full-duplex mode.

The PCnet-PCI II controller is register-compatible with the LANCE (Am7990) Ethernet controller, the C-LANCE (Am79C90) Ethernet controller, the ILACC (Am79C900) Ethernet controller, and all Ethernet controllers in the PCnet family, including the PCnet-ISA controller (Am79C960), PCnet-ISA+ controller (Am79C961), PCnet-ISA II controller (Am79C961A), PCnet-32 controller (Am79C965), PCnet-PCI controller (Am79970), and the PCnet-SCSI controller (Am79C974). The buffer management unit supports the C-LANCE, ILACC, and PCnet descriptor software models. The PCnet-PCI II controller is software compatible with the Novell® NE2100 and NE1500 Ethernet adapter card architectures.

The 32-bit multiplexed bus interface unit provides a direct interface to PCI local bus applications, simplifying the design of an Ethernet node in a PC system. The PCnet-PCI II controller provides the complete interface to an expansion ROM, allowing add-on card designs with only a single load per PCI bus interface pin. With its built-in support for both little and big endian byte alignment, this controller also addresses proprietary non-PC applications. The PCnet-PCI II controller's advanced CMOS design allows the bus interface to be connected to either a 5 V or a 3.3 V signaling environment. Both NAND Tree and JTAG test interfaces are provided.

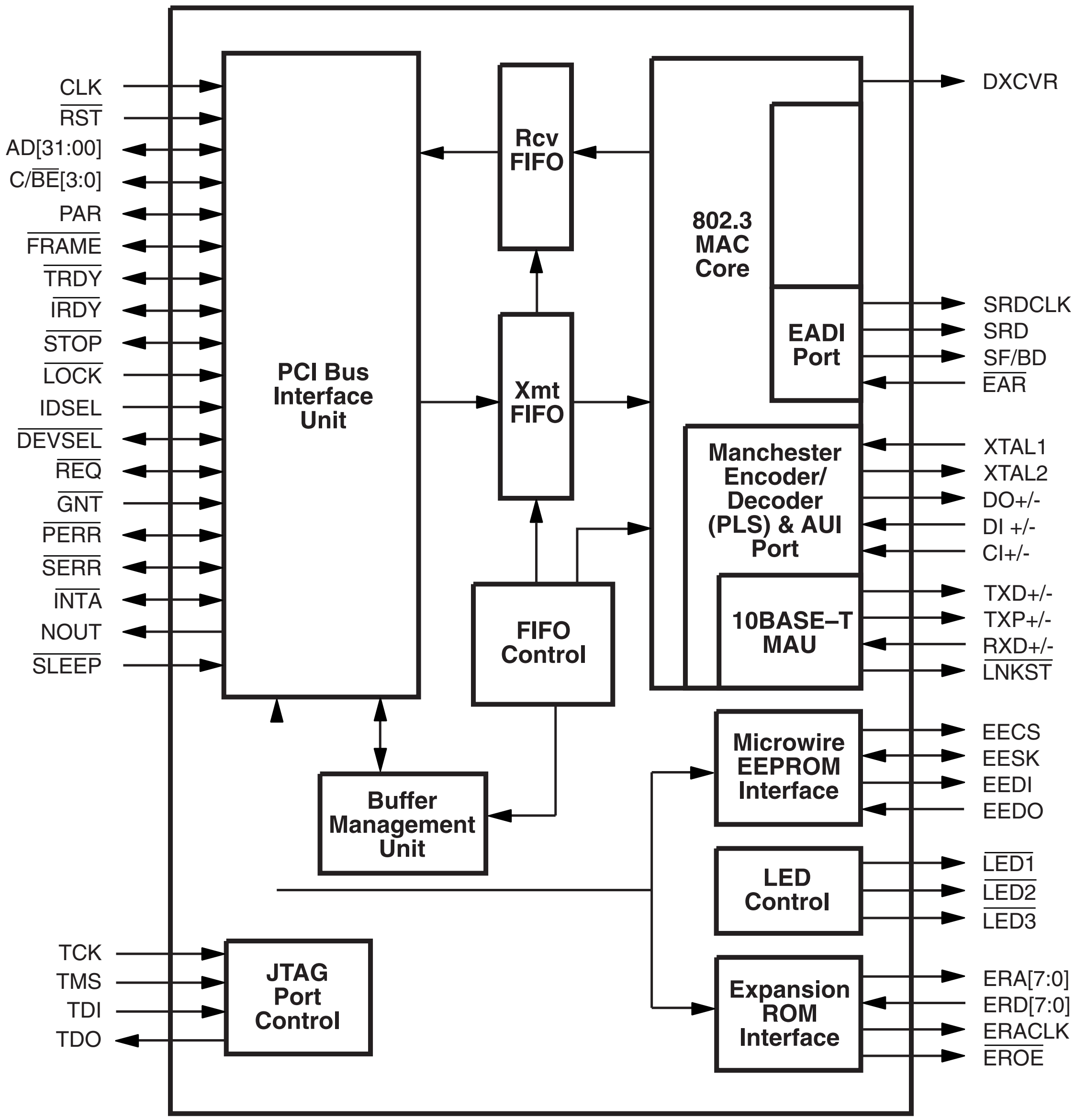
The PCnet-PCI II controller supports automatic configuration in the PCI configuration space. Additional PCnet-PCI II configuration parameters, including the unique IEEE physical address, can be read from an external non-volatile memory (microwire EEPROM) immediately following system reset.

The controller has the capability to automatically select either the AUI port or the twisted-pair transceiver. Only one interface is active at any one time. Both network interfaces can be programmed to operate in either half-duplex or full-duplex mode. The individual transmit and receive FIFOs optimize system overhead, providing sufficient latency during frame transmission and reception, and minimizing intervention during normal network error recovery. The integrated Manchester encoder/decoder (MENDEC) eliminates the need for an external serial interface adapter (SIA) in the system. In addition, the device provides programmable on-chip LED drivers for transmit, receive, collision, receive polarity, link integrity, activity, or jabber status. The PCnet-PCI II controller also provides an external address detection interface (EADI) to allow fast external hardware address filtering in internetworking applications.

For power-sensitive applications where low standby current is desired, the device incorporates two sleep functions to reduce overall system power consumption, excellent for notebooks and Green PCs. In conjunction with these low power modes, the PCnet-PCI II controller also has integrated functions to support Magic Packet, an inexpensive technology that allows remote wakeup of Green PCs.

With the rise of embedded networking applications operating in harsh environments where temperatures may exceed the normal commercial temperature window (0°C to +70°C), an industrial temperature (–40°C to +85°C) version is available in both the 132-pin PQFP and the 144-pin TQFP package. The industrial temperature version of the PCnet-PCI II Ethernet controller is characterized across the industrial temperature range (–40°C to +85°C) within the published power supply specification (4.75 V to 5.25 V; i.e., ±5% V_{CC}).

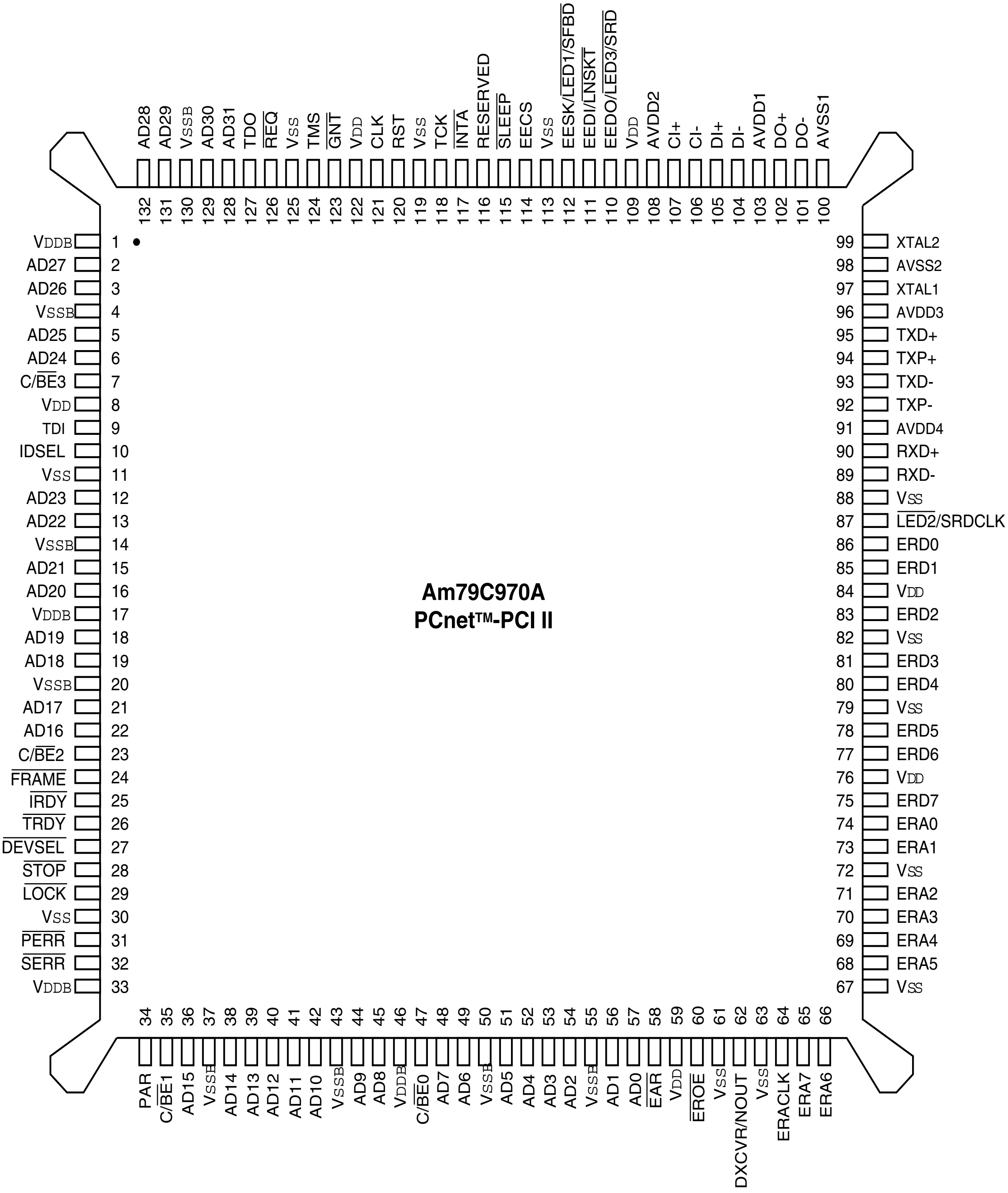
BLOCK DIAGRAM



19436C-1

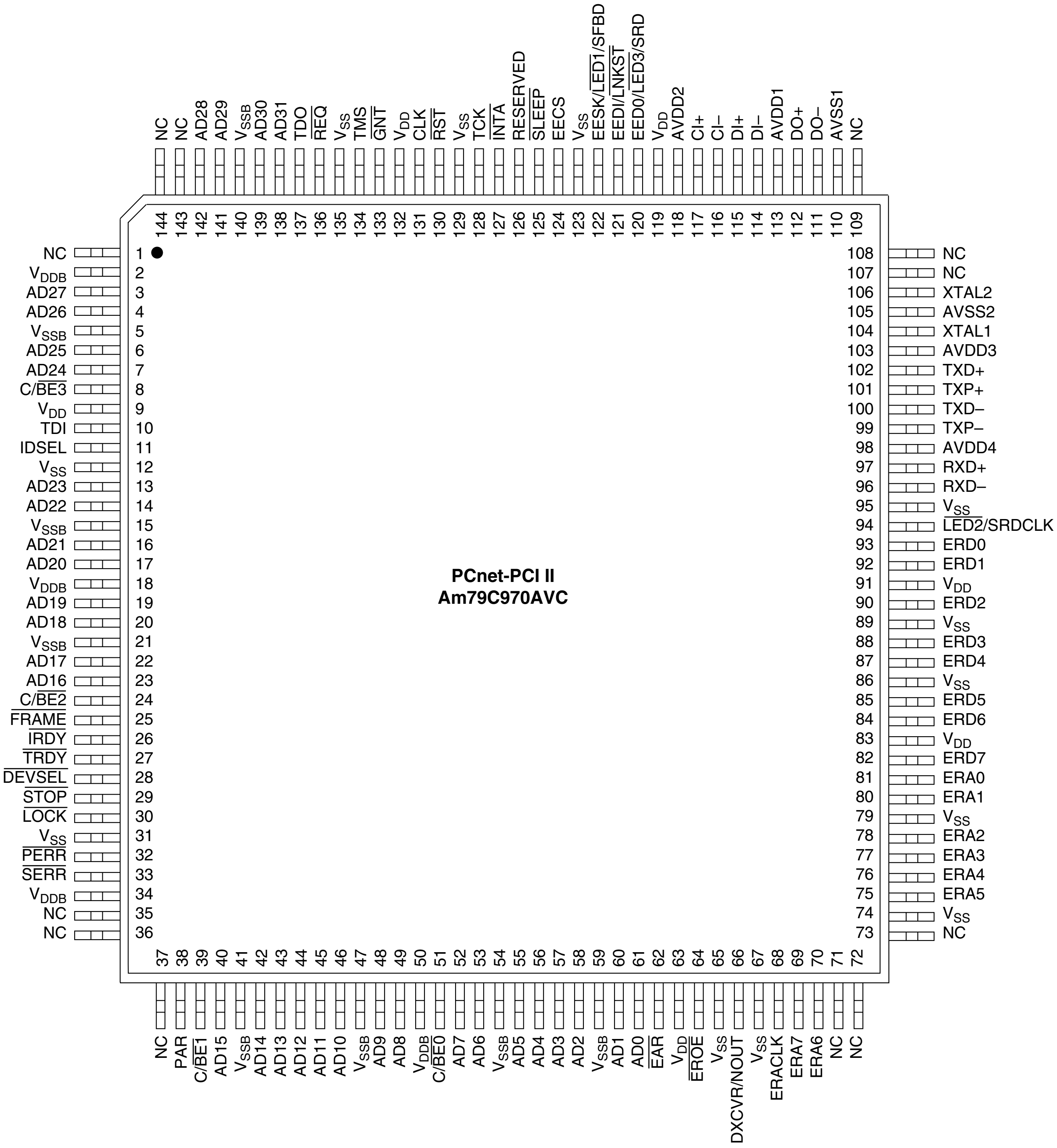
TABLE OF CONTENTS

CONNECTION DIAGRAM



19436C-2

CONNECTION DIAGRAM

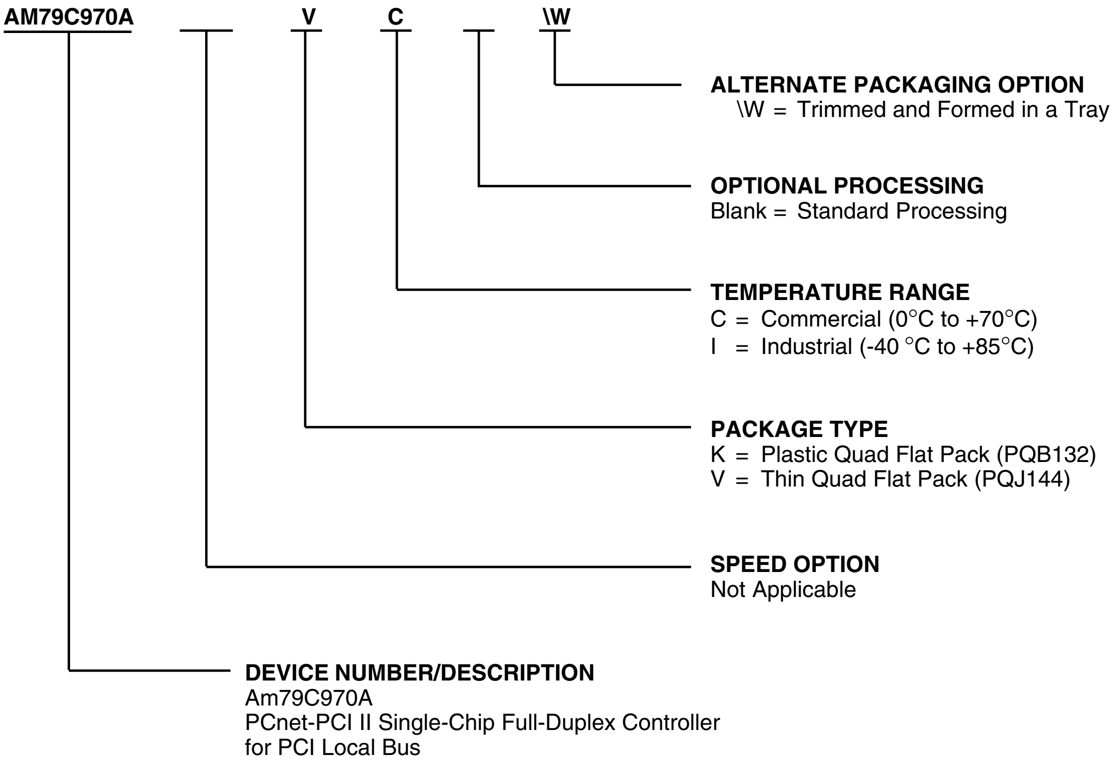


19436C-3

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:



AM79C970A ___ V C ___ \W

ALTERNATE PACKAGING OPTION
\W = Trimmed and Formed in a Tray

OPTIONAL PROCESSING
Blank = Standard Processing

TEMPERATURE RANGE
C = Commercial (0°C to +70°C)
I = Industrial (-40 °C to +85°C)

PACKAGE TYPE
K = Plastic Quad Flat Pack (PQB132)
V = Thin Quad Flat Pack (PQJ144)

SPEED OPTION
Not Applicable

DEVICE NUMBER/DESCRIPTION
Am79C970A
PCnet-PCI II Single-Chip Full-Duplex Controller
for PCI Local Bus

Valid Combinations	
AM79C970A	KC, KC\W, VC, VC\W
AM79C970A	KI, KI\W, VC, VI\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.

PIN DESIGNATIONS – 132 PIN PQFP

Listed By Pin Number

Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
1	V_{DDB}	34	PAR	67	V_{SS}	100	AVSS1
2	AD27	35	C/$\overline{\text{BE}}$1	68	ERA5	101	DO-
3	AD26	36	AD15	69	ERA4	102	DO+
4	V_{SSB}	37	V_{SSB}	70	ERA3	103	AVDD1
5	AD25	38	AD14	71	ERA2	104	DI-
6	AD24	39	AD13	72	V_{SS}	105	DI+
7	C/$\overline{\text{BE}}$3	40	AD12	73	ERA1	106	CI-
8	V_{DD}	41	AD11	74	ERA0	107	CI+
9	TDI	42	AD10	75	ERD7	108	AVDD2
10	IDSEL	43	V_{SSB}	76	V_{DD}	109	V_{DD}
11	V_{SS}	44	AD9	77	ERD6	110	EEDO/$\overline{\text{LED3}}$/SRD
12	AD23	45	AD8	78	ERD5	111	EEDI/$\overline{\text{LNKST}}$
13	AD22	46	V_{DDB}	79	V_{SS}	112	EESK/$\overline{\text{LED1}}$/SFBD
14	V_{SSB}	47	C/$\overline{\text{BE}}$0	80	ERD4	113	V_{SS}
15	AD21	48	AD7	81	ERD3	114	EECS
16	AD20	49	AD6	82	V_{SS}	115	$\overline{\text{SLEEP}}$
17	V_{DDB}	50	V_{SSB}	83	ERD2	116	RESERVED
18	AD19	51	AD5	84	V_{DD}	117	$\overline{\text{INTA}}$
19	AD18	52	AD4	85	ERD1	118	TCK
20	V_{SSB}	53	AD3	86	ERD0	119	V_{SS}
21	AD17	54	AD2	87	$\overline{\text{LED2}}$/SRDCLK	120	$\overline{\text{RST}}$
22	AD16	55	V_{SSB}	88	V_{SS}	121	CLK
23	C/$\overline{\text{BE}}$2	56	AD1	89	RXD-	122	V_{DD}
24	$\overline{\text{FRAME}}$	57	AD0	90	RXD+	123	$\overline{\text{GNT}}$
25	$\overline{\text{IRDY}}$	58	$\overline{\text{EAR}}$	91	AVDD4	124	TMS
26	$\overline{\text{TRDY}}$	59	V_{DD}	92	TXP-	125	V_{SS}
27	$\overline{\text{DEVSEL}}$	60	$\overline{\text{EROE}}$	93	TXD-	126	$\overline{\text{REQ}}$
28	$\overline{\text{STOP}}$	61	V_{SS}	94	TXP+	127	TDO
29	$\overline{\text{LOCK}}$	62	DXCVR/NOUT	95	TXD+	128	AD31
30	V_{SS}	63	V_{SS}	96	AVDD3	129	AD30
31	$\overline{\text{PERR}}$	64	ERACLK	97	XTAL1	130	V_{SSB}
32	$\overline{\text{SERR}}$	65	ERA7	98	AVSS2	131	AD29
33	V_{DDB}	66	ERA6	99	XTAL2	132	AD28

PIN DESIGNATIONS – 132 PIN PQFP

Listed By Group

Pin Name	Pin Function	Type	Driver	No. of Pins
PCI Bus Interface				
AD[31:0]	Address/Data Bus	IO	TS3	32
C/$\overline{\text{BE}}$[3:0]	Bus Command/Byte Enable	IO	TS3	4
CLK	Bus Clock	I	N/A	1
$\overline{\text{DEVSEL}}$	Device Select	IO	STS6	1
$\overline{\text{FRAME}}$	Cycle Frame	IO	STS6	1
$\overline{\text{GNT}}$	Bus Grant	I	N/A	1
IDSEL	Initialization Device Select	I	N/A	1
$\overline{\text{INTA}}$	Interrupt	IO	TS6	1
$\overline{\text{IRDY}}$	Initiator Ready	IO	STS6	1
$\overline{\text{LOCK}}$	Bus Lock	I	N/A	1
PAR	Parity	IO	TS3	1
$\overline{\text{PERR}}$	Parity Error	IO	STS6	1
$\overline{\text{REQ}}$	Bus Request	IO	TS3	1
$\overline{\text{RST}}$	Reset	1	N/A	1
$\overline{\text{SERR}}$	System Error	IO	TS6	1
$\overline{\text{STOP}}$	Stop	IO	STS6	1
$\overline{\text{TRDY}}$	Target Ready	IO	STS6	1
Board Interface				
$\overline{\text{LED1}}$	LED1	O	LED	1
$\overline{\text{LED2}}$	LED2	O	LED	1
$\overline{\text{LED3}}$	LED3	O	LED	1
$\overline{\text{SLEEP}}$	Sleep Mode	I	N/A	1
XTAL1	Crystal Input	I	N/A	1
XTAL2	Crystal Output	O	XTAL	1
Microwave EEPROM Interface				
EECS	Microwire Serial EEPROM Chip Select	O	O6	1
EEDI	Microwire Serial EEPROM Data In	O	LED	1
EEDO	Microwire Address EEPROM Data Out	I	N/A	1
EESK	Microwire Serial PROM Clock	IO	LED	1
Expansion ROM Interface				
ERA[7:0]	Expansion ROM Address Bus	O	O6	8
ERACLK	Expansion ROM Address Clock	O	O6	1
ERD[7:0]	Expansion ROM Data Bus	I	N/A	8
$\overline{\text{EROE}}$	Expansion ROM Output Enable	O	O6	1

PIN DESIGNATIONS – 132 PIN PQFP

Listed By Group

Pin Name	Pin Function	Type[1]	Driver	No. of Pins
PCI Bus Interface				
Attachment Unit Interface AUI				
CI+/CI-	AUI Collision Differential Pair	I	N/A	2
DI+/DI-	AUI Data In Differential Pair	I	N/A	2
DO+/DO-	AUI Data Out Differential Pair	O	DO	2
DXCVR	Disable Transceiver	O	O6	1
10Base-T Interface				
$\overline{\text{LNKST}}$	Link Status	O	LED	1
RXD+/RXD-	Receive Differential Pair	I	N/A	2
TXD+/TXD-	Transmit Differential Pair	O	TDO	2
TXP+/TXP-	Transmit Pre-distortion Differential Pair	O	TPO	2
External Address Detection Interface (EADI)				
$\overline{\text{EAR}}$	External Address Reject Low	I	N/A	1
SFBD	Start Frame Byte Delimiter	O	LED	1
SRD	Serial Receive Data	O	LED	1
SRDCLK	Serial Receive Data Clock	O	LED	1
IEEE 1149.1 Test Access Port Interface (JTAG)				
TCK	Test Clock	I	N/A	1
TDI	Test Data In	I	N/A	1
TDO	Test Data Out	O	TS6	1
TMS	Test Mode Select	I	N/A	1
Test Interface				
NOUT	NAND Tree Test Output	O	O6	1
Power Supplies				
AV_{DD}	Analog Power	P	N/A	4
AV_{SS}	Analog Ground	P	N/A	2
V_{DD}	Digital Power	P	N/A	6
V_{SS}	Digital Ground	P	N/A	12
V_{DDB}	I/O Buffer Power	P	N/A	4
V_{SSB}	I/O Buffer Ground	P	N/A	8

PIN DESIGNATIONS – 144-PIN TQFP

Listed By Pin Number

Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
1	NC	**37**	NC	**73**	NC	**109**	NC
2	V_{DDB}	**38**	PAR	**74**	V_{SS}	**110**	AVSS1
3	AD27	**39**	C/$\overline{\text{BE1}}$	**75**	ERA5	**111**	DO-
4	AD26	**40**	AD15	**76**	ERA4	**112**	DO+
5	V_{SSB}	**41**	V_{SSB}	**77**	ERA3	**113**	AVDD1
6	AD25	**42**	AD14	**78**	ERA2	**114**	DI-
7	AD24	**43**	AD13	**79**	V_{SS}	**115**	DI+
8	C/$\overline{\text{BE3}}$	**44**	AD12	**80**	ERA1	**116**	CI-
9	V_{DD}	**45**	AD11	**81**	ERA0	**117**	CI+
10	TDI	**46**	AD10	**82**	ERD7	**118**	AVDD2
11	IDSEL	**47**	V_{SSB}	**83**	V_{DD}	**119**	V_{DD}
12	V_{SS}	**48**	AD9	**84**	ERD6	**120**	EEDO/$\overline{\text{LED3}}$/SRD
13	AD23	**49**	AD8	**85**	ERD5	**121**	EEDI/$\overline{\text{LNKST}}$
14	AD22	**50**	V_{DDB}	**86**	V_{SS}	**122**	EESK/LED1/SFBD
15	V_{SSB}	**51**	C/$\overline{\text{BE0}}$	**87**	ERD4	**123**	V_{SS}
16	AD21	**52**	AD7	**88**	ERD3	**124**	EECS
17	AD20	**53**	AD6	**89**	V_{SS}	**125**	$\overline{\text{SLEEP}}$
18	V_{DDB}	**54**	V_{SSB}	**90**	ERD2	**126**	RESERVED
19	AD19	**55**	AD5	**91**	V_{DD}	**127**	$\overline{\text{INTA}}$
20	AD18	**56**	AD4	**92**	ERD1	**128**	TCK
21	V_{SSB}	**57**	AD3	**93**	ERD0	**129**	V_{SS}
22	AD17	**58**	AD2	**94**	$\overline{\text{LED2}}$/SRDCLK	**130**	$\overline{\text{RST}}$
23	AD16	**59**	V_{SSB}	**95**	V_{SS}	**131**	CLK
24	C/$\overline{\text{BE2}}$	**60**	AD1	**96**	RXD-	**132**	V_{DD}
25	$\overline{\text{FRAME}}$	**61**	AD0	**97**	RXD+	**133**	$\overline{\text{GNT}}$
26	$\overline{\text{IRDY}}$	**62**	$\overline{\text{EAR}}$	**98**	AVDD4	**134**	TMS
27	$\overline{\text{TRDY}}$	**63**	V_{DD}	**99**	TXP-	**135**	V_{SS}
28	$\overline{\text{DEVSEL}}$	**64**	$\overline{\text{EROE}}$	**100**	TXD-	**136**	$\overline{\text{REQ}}$
29	$\overline{\text{STOP}}$	**65**	V_{SS}	**101**	TXP+	**137**	TDO
30	$\overline{\text{LOCK}}$	**66**	DXCVR/NOUT	**102**	TXD+	**138**	AD31
31	V_{SS}	**67**	V_{SS}	**103**	AVDD3	**139**	AD30
32	$\overline{\text{PERR}}$	**68**	ERACLK	**104**	XTAL1	**140**	V_{SSB}
33	$\overline{\text{SERR}}$	**69**	ERA7	**105**	AVSS2	**141**	AD29
34	V_{DDB}	**70**	ERA6	**106**	XTAL2	**142**	AD28
35	NC	**71**	NC	**107**	NC	**143**	NC
36	NC	**72**	NC	**108**	NC	**144**	NC

NC - Indicates no connect

PIN DESIGNATIONS – 144 PIN TQFP

Listed By Group

Pin Name	Pin Function	Type	Driver	No. of Pins
PCI Bus Interface				
AD[31:0]	Address/Data Bus	IO	TS3	32
C/$\overline{\text{BE}}$[3:0]	Bus Command/Byte Enable	IO	TS3	4
CLK	Bus Clock	I	N/A	1
$\overline{\text{DEVSEL}}$	Device Select	IO	STS6	1
$\overline{\text{FRAME}}$	Cycle Frame	IO	STS6	1
$\overline{\text{GNT}}$	Bus Grant	I	N/A	1
IDSEL	Initialization Device Select	I	N/A	1
$\overline{\text{INTA}}$	Interrupt	IO	TS6	1
$\overline{\text{IRDY}}$	Initiator Ready	IO	STS6	1
$\overline{\text{LOCK}}$	Bus Lock	I	N/A	1
PAR	Parity	IO	TS3	1
$\overline{\text{PERR}}$	Parity Error	IO	STS6	1
$\overline{\text{REQ}}$	Bus Request	IO	TS3	1
$\overline{\text{RST}}$	Reset	1	N/A	1
$\overline{\text{SERR}}$	System Error	IO	TS6	1
$\overline{\text{STOP}}$	Stop	IO	STS6	1
$\overline{\text{TRDY}}$	Target Ready	IO	STS6	1
Board Interface				
$\overline{\text{LED1}}$	LED1	O	LED	1
$\overline{\text{LED2}}$	LED2	O	LED	1
$\overline{\text{LED3}}$	LED3	O	LED	1
$\overline{\text{SLEEP}}$	Sleep Mode	I	N/A	1
XTAL1	Crystal Input	I	N/A	1
XTAL2	Crystal Output	O	XTAL	1
Microwave EEPROM Interface				
EECS	Microwire Serial EEPROM Chip Select	O	O6	1
EEDI	Microwire Serial EEPROM Data In	O	LED	1
EEDO	Microwire Address EEPROM Data Out	I	N/A	1
EESK	Microwire Serial PROM Clock	IO	LED	1
Expansion ROM Interface				
ERA[7:0]	Expansion ROM Address Bus	O	O6	8
ERACLK	Expansion ROM Address Clock	O	O6	1
ERD[7:0]	Expansion ROM Data Bus	I	N/A	8
$\overline{\text{EROE}}$	Expansion ROM Output Enable	O	O6	1

PIN DESIGNATIONS — 144-PIN TQFP

Listed By Group

Pin Name	Pin Function	Type[1]	Driver	No. of Pins
PCI Bus Interface				
Attachment Unit Interface AUI				
CI+/CI-	AUI Collision Differential Pair	I	N/A	2
DI+/DI-	AUI Data In Differential Pair	I	N/A	2
DO+/DO-	AUI Data Out Differential Pair	O	DO	2
DXCVR	Disable Transceiver	O	O6	1
10Base-T Interface				
$\overline{\text{LNKST}}$	Link Status	O	LED	1
RXD+/RXD-	Receive Differential Pair	I	N/A	2
TXD+/TXD-	Transmit Differential Pair	O	TDO	2
TXP+/TXP-	Transmit Pre-distortion Differential Pair	O	TPO	2
External Address Detection Interface (EADI)				
$\overline{\text{EAR}}$	External Address Reject Low	I	N/A	1
SFBD	Start Frame Byte Delimiter	O	LED	1
SRD	Serial Receive Data	O	LED	1
SRDCLK	Serial Receive Data Clock	O	LED	1
IEEE 1149.1 Test Access Port Interface (JTAG)				
TCK	Test Clock	I	N/A	1
TDI	Test Data In	I	N/A	1
TDO	Test Data Out	O	TS6	1
TMS	Test Mode Select	I	N/A	1
Test Interface				
NOUT	NAND Tree Test Output	O	O6	1
Power Supplies				
AV_{DD}	Analog Power	P	N/A	4
AV_{SS}	Analog Ground	P	N/A	2
V_{DD}	Digital Power	P	N/A	6
V_{SS}	Digital Ground	P	N/A	12
V_{DDB}	I/O Buffer Power	P	N/A	4
V_{SSB}	I/O Buffer Ground	P	N/A	8

PIN DESIGNATIONS

Listed By Driver Type

The next table describes the various types of drivers that are used in the PCnet-PCI II controller:

Name	Type	I_{OL} (mA)	I_{OH} (mA)	Load (pF)
LED	LED	12	-0.4	50
O6	Totem Pole	6	-0.4	50
OD6	Open Drain	6	N/A	50
STS6	Sustained Tri-State™	6	-2	50
TS3	Tri-State	3	-2	50
TS6	Tri-State	6	-2	50

All I_{OL} and I_{OH} values shown in the table above apply to 5 V signaling. See the section “DC Characteristics” for the values applying to 3.3 V signaling.

A sustained tri-state signal is a low active signal that is driven high for one clock period before it is left floating.

DO, TDO and TPO are differential output drivers. The characteristic of these and the XTAL output are described in the section “DC Characteristics”.

PIN DESCRIPTION

PCI Interface

AD[31:0]

Address and Data Input/Output

Address and data are multiplexed on the same bus interface pins. During the first clock of a transaction AD[31:0] contain a physical address (32 bits). During the subsequent clocks AD[31:0] contain data. Byte ordering is little endian by default. AD[7:0] are defined as least significant byte and AD[31:24] are defined as the most significant byte. For FIFO data transfers, the PCnet-PCI II controller can be programmed for big endian byte ordering. See CSR3, bit 2 (BSWP) for more details.

During the address phase of the transaction, when the PCnet-PCI II controller is a bus master, AD[31:2] will address the active Double Word (DWord). The PCnet-PCI II controller always drives AD[1:0] to ‘00’ during the address phase indicating linear burst order. When the PCnet-PCI II controller is not a bus master, the AD[31:0] lines are continuously monitored to determine if an address match exists for slave transfers.

During the data phase of the transaction, AD[31:0] are driven by the PCnet-PCI II controller when performing bus master write and slave read operations. Data on AD[31:0] is latched by the PCnet-PCI II controller when performing bus master read and slave write operations.

When RST is active, AD[31:0] are inputs for NAND tree testing.

C/BE[3:0]

Bus Command and Byte Enables Input/Output

Bus command and byte enables are multiplexed on the same bus interface pins. During the address phase of the transaction, C/BE[3:0] define the bus command. During the data phase C/BE[3:0] are used as byte enables. The byte enables define which physical byte lanes carry meaningful data. C/BE0 applies to byte 0 (AD[7:0]) and C/BE3 applies to byte 3 (AD[31:24]). The function of the byte enables is independent of the byte ordering mode (BSWP, CSR3, bit 2).

When RST is active, C/BE[3:0] are inputs for NAND tree testing.

CLK

Clock Input

This clock is used to drive the system bus interface and the internal buffer management unit. All bus signals are sampled on the rising edge of CLK and all parameters are defined with respect to this edge. The PCnet-PCI II controller operates over a range of 0 MHz to 33 MHz. This clock is not used to drive the network functions.

When RST is active, CLK is an input for NAND tree testing.

DEVSEL

Device Select Input/Output

The PCnet-PCI II controller drives DEVSEL when it detects a transaction that selects the device as a target. The device samples DEVSEL to detect if a target claims a transaction that the PCnet-PCI II controller has initiated.

When RST is active, DEVSEL is an input for NAND tree testing.

FRAME

Cycle Frame Input/Output

FRAME is driven by the PCnet-PCI II controller when it is the bus master to indicate the beginning and duration of a transaction. FRAME is asserted to indicate a bus transaction is beginning. FRAME is asserted while data transfers continue. FRAME is deasserted before the final data phase of a transaction. When the PCnet-PCI II controller is in slave mode, it samples FRAME to determine the address phase of transaction.

When RST is active, FRAME is an input for NAND tree testing.

GNT

Bus Grant Input

This signal indicates that the access to the bus has been granted to the PCnet-PCI II controller.

The PCnet-PCI II controller supports bus parking. When the PCI bus is idle and the system arbiter asserts GNT without an active REQ from the PCnet-PCI II controller, the device will drive the AD[31:0], C/BE[3:0] and PAR lines.

When RST is active, GNT is an input for NAND tree testing.

IDSEL

Initialization Device Select Input

This signal is used as a chip select for the PCnet-PCI II controller during configuration read and write transactions.

When RST is active, IDSEL is an input for NAND tree testing.

INTA

Interrupt Request Input/Output

An attention signal which indicates that one or more of the following status flags is set: BABL, EXDINT, IDON, JAB, MERR, MISS, MFCO, MPINT, RCVCCO, RINT, SINT, SLPINT, TINT, TXSTRT and UINT. Each status flag has either a mask or an enable bit which allows for suppression of INTA assertion. The flags have the following meaning:

Table 1. Interrupt Flags

BABL	Babble
EXDINT	Excessive Deferral
IDON	Initialization Done
JAB	Jabber
MERR	Memory Error
MISS	Missed Frame
MFCO	Missed Frame Count Overflow
MPINT	Magic Packet Interrupt
RCVCCO	Receive Collision Count Overflow
RINT	Receive Interrupt
SLPINT	Sleep Interrupt
SINT	System Error
TINT	Transmit Interrupt
TXSTRT	Transmit Start
UINT	User Interrupt

By default INTA is an open-drain output. For applications that need a high-active edge sensitive interrupt signal, the INTA pin can be configured for this mode by setting INTLEVEL (BCR2, bit 7) to ONE.

When RST is active, INTA is an input for NAND tree testing.

IRDY

Initiator Ready **Input/Output**

IRDY indicates the ability of the initiator of the transaction to complete the current data phase. IRDY is used in conjunction with TRDY. Wait states are inserted until both IRDY and TRDY are asserted simultaneously. A data phase is completed on any clock when both IRDY and TRDY are asserted.

When the PCnet-PCI II controller is a bus master, it asserts IRDY during all write data phases to indicate that valid data is present on AD[31:0]. During all read data phases the device asserts IRDY to indicate that it is ready to accept the data.

When the PCnet-PCI II controller is the target of a transaction, it checks IRDY during all write data phases to determine if valid data is present on AD[31:0]. During all read data phases the device checks IRDY to determine if the initiator is ready to accept the data.

When RST is active, IRDY is an input for NAND tree testing.

LOCK

Lock **Input**

In slave mode, LOCK is an input to the PCnet-PCI II controller. A bus master can lock the device to guarantee an atomic operation that requires multiple transactions.

The PCnet-PCI II controller will never assert LOCK as a master.

When RST is active, LOCK is an input for NAND tree testing.

PAR

Parity **Input/Output**

Parity is even parity across AD[31:0] and C/BE[3:0]. When the PCnet-PCI II controller is a bus master, it generates parity during the address and write data phases. It checks parity during read data phases. When the PCnet-PCI II controller operates in slave mode, it checks parity during every address phase. When it is the target of a cycle, it checks parity during write data phases and it generates parity during read data phases.

When RST is active, PAR is an input for NAND tree testing.

PERR

Parity Error **Input/Output**

During any slave write transaction and any master read transaction, the PCnet-PCI II controller asserts PERR when it detects a data parity error and reporting of the error is enabled by setting PERREN (PCI Command register, bit 6) to ONE. During any master write transaction the PCnet-PCI II controller monitors PERR to see if the target reports a data parity error.

When RST is active, PERR is an input for NAND tree testing.

REQ

Bus Request **Input/Output**

The PCnet-PCI II controller asserts REQ pin as a signal that it wishes to become a bus master. REQ is driven high when the PCnet-PCI II controller does not request the bus.

When RST is active, REQ is an input for NAND tree testing.

RST

Reset **Input**

When RST is asserted low, then the PCnet-PCI II controller performs an internal system reset of the type H_RESET (HARDWARE_RESET). RST must be held for a minimum of 30 clock periods. While in the H_RESET state, the PCnet-PCI II controller will disable or deassert all outputs. RST may be asynchronous to CLK when asserted or deasserted. It is recommended that the deassertion be synchronous to guarantee clean and bounce free edge.

When $\overline{\text{RST}}$ is active, NAND tree testing is enabled. All PCI interface pins are in input mode. The result of the NAND tree testing can be observed on the NOUT output (pin 62).

$\overline{\text{SERR}}$

System Error **Input/Output**

During any slave transaction, the PCnet-PCI II controller asserts $\overline{\text{SERR}}$ when it detects an address parity error and reporting of the error is enabled by setting PERREN (PCI Command register, bit 6) and SERREN (PCI Command register, bit 8) to ONE.

By default $\overline{\text{SERR}}$ is an open-drain output. For component test it can be programmed to be an active-high totem-pole output.

When $\overline{\text{RST}}$ is active, $\overline{\text{SERR}}$ is an input for NAND tree testing.

$\overline{\text{STOP}}$

Stop **Input/Output**

In slave mode, the PCnet-PCI II controller drives the $\overline{\text{STOP}}$ signal to inform the bus master to stop the current transaction. In bus master mode, the PCnet-PCI II controller checks $\overline{\text{STOP}}$ to determine if the target wants to disconnect the current transaction.

When $\overline{\text{RST}}$ is active, $\overline{\text{STOP}}$ is an input for NAND tree testing.

$\overline{\text{TRDY}}$

Target Ready **Input/Output**

$\overline{\text{TRDY}}$ indicates the ability of the target of the transaction to complete the current data phase. $\overline{\text{TRDY}}$ is used in conjunction with $\overline{\text{IRDY}}$. Wait states are inserted until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted simultaneously. A data phase is completed on any clock when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted.

When the PCnet-PCI II controller is a bus master, it checks $\overline{\text{TRDY}}$ during all read data phases to determine if valid data is present on AD[31:0]. During all write data phases the device checks $\overline{\text{TRDY}}$ to determine if the target is ready to accept the data.

When the PCnet-PCI II controller is the target of a transaction, it asserts $\overline{\text{TRDY}}$ during all read data phases to indicate that valid data is present on AD[31:0]. During all write data phases the device asserts $\overline{\text{TRDY}}$ to indicate that it is ready to accept the data.

When $\overline{\text{RST}}$ is active, $\overline{\text{TRDY}}$ is an input for NAND tree testing.

Board Interface

$\overline{\text{LED1}}$

LED1 **Output**

This output is designed to directly drive an LED. By default, $\overline{\text{LED1}}$ indicates receive activity on the network. This pin can also be programmed to indicate other network status (see BCR5). The $\overline{\text{LED1}}$ pin polarity is programmable, but by default, it is active LOW.

Note that the $\overline{\text{LED1}}$ pin is multiplexed with the EESK and SFBD pins.

$\overline{\text{LED2}}$

LED2 **Output**

This output is designed to directly drive an LED. By default, $\overline{\text{LED2}}$ indicates correct receive polarity on the 10BASE-T interface. This pin can also be programmed to indicate other network status (see BCR6). The $\overline{\text{LED2}}$ pin polarity is programmable, but by default, it is active LOW.

Note that the $\overline{\text{LED2}}$ pin is multiplexed with the SRDCLK pin.

$\overline{\text{LED3}}$

LED3 **Output**

This output is designed to directly drive an LED. By default, $\overline{\text{LED3}}$ indicates transmit activity on the network. This pin can also be programmed to indicate other network status (see BCR7). The $\overline{\text{LED3}}$ pin polarity is programmable, but by default, it is active LOW.

Note that the $\overline{\text{LED3}}$ pin is multiplexed with the EEDO and SRD pins.

Special attention must be given to the external circuitry attached to this pin. When this pin is used to drive an LED while an EEPROM is used in the system, then buffering is required between the $\overline{\text{LED3}}$ pin and the LED circuit. If an LED circuit were directly attached to this pin, it would create an IOL requirement that could not be met by the serial EEPROM attached to this pin. If no EEPROM is included in the system design, then the $\overline{\text{LED3}}$ signal may be directly connected to an LED without buffering. For more details regarding LED connection, see the section “LED Support”.

$\overline{\text{SLEEP}}$

Sleep **Input**

When $\overline{\text{SLEEP}}$ is asserted, the PCnet-PCI II controller performs an internal system reset of the S_RESET type and then proceeds into a power savings mode. All PCnet-PCI II controller outputs will be placed in their normal reset condition. All PCnet-PCI II controller inputs will be ignored except for the $\overline{\text{SLEEP}}$ pin itself. Deassertion of $\overline{\text{SLEEP}}$ results in wake-up. The system must refrain from starting the network operations of the PCnet-PCI II controller device for 0.5 s following the deassertion of the $\overline{\text{SLEEP}}$ signal in order to allow internal analog circuits to stabilize.

Both CLK and XTAL1 inputs must have valid clock signals present in order for the $\overline{\text{SLEEP}}$ command to take effect.

The $\overline{\text{SLEEP}}$ pin should not be asserted during power supply ramp-up. If it is desired that $\overline{\text{SLEEP}}$ be asserted at power up time, then the system must delay the assertion of $\overline{\text{SLEEP}}$ until three clock cycles after the completion of a hardware reset operation.

The $\overline{\text{SLEEP}}$ pin must not be left unconnected. It should be tied to VDD, if the power savings mode is not used.

XTAL1

Crystal Oscillator In **Input**

The internal clock generator uses a 20 MHz crystal that is attached to the pins XTAL1 and XTAL2. The network data rate is one-half of the crystal frequency. XTAL1 may alternatively be driven using an external 20 MHz CMOS level clock signal. Refer to the section "External Crystal Characteristics" for more details.

Note that when the PCnet-PCI II controller is in coma mode, there is an internal 22 kΩ resistor from XTAL1 to ground. If an external source drives XTAL1, some power will be consumed driving this resistor. If XTAL1 is driven LOW at this time power consumption will be minimized. In this case, XTAL1 must remain active for at least 30 cycles after the assertion of $\overline{\text{SLEEP}}$ and deassertion of $\overline{\text{REQ}}$.

XTAL2

Crystal Oscillator Out **Output**

The internal clock generator uses a 20 MHz crystal that is attached to the pins XTAL1 and XTAL2. The network data rate is one-half of the crystal frequency. If an external clock source is used on XTAL1, then XTAL 2 should be left unconnected.

Microwire EEPROM Interface

EECS

EEPROM Chip Select **Output**

This pin is designed to directly interface to a serial EEPROM that uses the Microwire interface protocol. EECS is connected to the Microwire EEPROM chip select pin. It is controlled by either the PCnet-PCI II controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 2.

EEDI

EEPROM Data In **Output**

This pin is designed to directly interface to a serial EEPROM that uses the Microwire interface protocol. EEDI is connected to the Microwire EEPROM data input pin. It is controlled by either the PCnet-PCI II controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 0.

Note that the EEDI pin is multiplexed with the $\overline{\text{LNKST}}$ pin.

EEDO

EEPROM Data Out **Input**

This pin is designed to directly interface to a serial EEPROM that uses the Microwire interface protocol. EEDO is connected to the Microwire EEPROM data output pin. It is controlled by either the PCnet-PCI II controller during command portions of a read of the entire EEPROM, or indirectly by the host system by reading from BCR19, bit 0.

Note that the EEDO pin is multiplexed with the $\overline{\text{LED3}}$ and SRD pins.

EESK

EEPROM Serial clock **Input/Output**

This pin is designed to directly interface to a serial EEPROM that uses the Microwire interface protocol. EESK is connected to the Microwire EEPROM clock pin. It is controlled by either the PCnet-PCI II controller directly during a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 1.

Note that the EESK pin is multiplexed with the $\overline{\text{LED1}}$ and SFBD pins.

The EESK pin is also used during EEPROM Auto-detection to determine whether or not an EEPROM is present at the PCnet-PCI II controller Microwire interface. At the rising edge of CLK during the last clock during which $\overline{\text{RST}}$ is asserted, EESK is sampled to determine the value of the EEDET bit in BCR19. A sampled HIGH value means that an EEPROM is present, and EEDET will be set to ONE. A sampled LOW value means that an EEPROM is not present, and EEDET will be cleared to ZERO. See the section "EEPROM Auto-Detection" for more details.

If no LED circuit is to be attached to this pin, then a pull up or pull down resistor must be attached instead, in order to resolve the EEDET setting.

Expansion ROM Interface

ERA[7:0]

Expansion ROM Address **Output**

These pins provide the address to the Expansion ROM. When $\overline{\text{EROE}}$ is asserted and ERACLK is driven HIGH, ERA[7:0] contain the upper 8 bits of the Expansion ROM address. They must be latched externally. When $\overline{\text{EROE}}$ is asserted and ERACLK is low, ERA[7:0] contain the lower 8 bits of the Expansion ROM address.

All ERA outputs are forced to a constant level to conserve power while no access to the Expansion ROM is performed.

ERACLK

Expansion ROM Address Clock **Output**

When $\overline{\text{EROE}}$ is asserted and ERACLK is driven HIGH, ERA[7:0] contain the upper 8 bits of the Expansion ROM address. ERACLK is used to latch the address bits externally. Both ‘373 (transparent latch) and ‘374 (D flip-flop) types of address latch are supported.

ERD[7:0]

Expansion ROM Data **Input**

Data from the Expansion ROM is transferred on ERD[7:0]. When $\overline{\text{EROE}}$ is high, the ERD[7:0] inputs are internally disabled and can be left floating.

$\overline{\text{EROE}}$

Expansion ROM Output Enable **Output**

This signal is asserted when the Expansion ROM is read.

Attachment Unit Interface

CI±

Collision In **Input**

CI± is a differential input pair signaling the PCnet-PCI II controller that a collision has been detected on the network media, indicated by the CI± inputs being driven with a 10 MHz pattern of sufficient amplitude and pulse width to meet ISO 8802-3 (IEEE/ANSI 802.3) standards. Operates at pseudo ECL levels.

DI±

Data In **Input**

DI± is a differential input pair to the PCnet-PCI II controller carrying Manchester encoded data from the network. Operates at pseudo ECL levels.

DO±

Data Out **Output**

DO± is a differential output pair from the PCnet-PCI II controller for transmitting Manchester encoded data to the network. Operates at pseudo ECL levels.

DXCVR

Disable Transceiver **Output**

The DXCVR signal is provided to power down an external transceiver or DC-to-DC converter in designs that provide more than one network connection.

The polarity of the asserted state of the DXCVR output is controlled by DXCVRPOL (BCR2, bit 4). By default, the DXCVR output is high when asserted. When the 10BASE-T interface is the active network port, the DXCVR output is always deasserted. When the AUI interface is the active network port, the assertion of the DXCVR output is controlled by the setting of DXCVRCTL (BCR2, bit 5).

Note that the DXCVR pin is multiplexed with the NOUT pin.

Twisted Pair Interface

$\overline{\text{LNKST}}$

Link Status **Output**

This output is designed to directly drive an LED. By default, $\overline{\text{LNKST}}$ indicates an active link connection on the 10BASE-T interface. This pin can also be programmed to indicate other network status (see BCR4). The $\overline{\text{LNKST}}$ pin polarity is programmable, but by default, it is active LOW.

Note that the $\overline{\text{LNKST}}$ pin is multiplexed with the EEDI pin.

RXD±

10BASE-T Receive Data **Input**

10BASE-T port differential receivers.

TXD±

10BASE-T Transmit Data **Output**

10BASE-T port differential drivers.

TXP±

10BASE-T Pre-Distortion Control **Output**

These outputs provide transmit pre-distortion control in conjunction with the 10BASE-T port differential drivers.

External Address Detection Interface

$\overline{\text{EAR}}$

External Address Reject Low **Input**

The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be ORd with the value on the $\overline{\text{EAR}}$ pin. The $\overline{\text{EAR}}$ pin is defined as $\overline{\text{REJECT}}$. The pin value is “OR”ed with the internal address detection result to determine if the current frame should be accepted or rejected.

The $\overline{\text{EAR}}$ pin is internally pulled-up and can be left unconnected, if the EADI interface is not used.

SFBD

Start Frame—Byte Delimiter **Output**

An initial rising edge on the SFBD signal indicates that a start of frame delimiter has been detected. The serial bit stream will follow on the SRD signal, commencing with the destination address field. SFBD will go high for 4 bit times (400 ns) after detecting the second ONE in the SFD (Start of Frame Delimiter) of a received frame. SFBD will subsequently toggle every 400 ns (1.25 MHz frequency) with each rising edge indicating the first bit of each subsequent byte of the received serial bit stream. SFBD will be inactive during frame transmission.

Note that the SFBD pin is multiplexed with the EESK and $\overline{\text{LED1}}$ pins.

SRD

Serial Receive Data **Output**

SRD is the decoded NRZ data from the network. This signal can be used for external address detection. When the 10BASE-T port is selected, transitions on SRD will only occur during receive activity. When the AUI port is selected, transitions on SRD will occur during both transmit and receive activity. Note that the SRD pin is multiplexed with the EEDO and $\overline{\text{LED3}}$ pins.

SRDCLK

Serial Receive Data Clock **Output**

Serial Receive Data is synchronous with reference to SRDCLK. When the 10BASE-T port is selected, transitions on SRDCLK will only occur during receive activity. When the AUI port is selected, transitions on SRDCLK will occur during both transmit and receive activity.

Note that the SRDCLK pin is multiplexed with the $\overline{\text{LED2}}$ pin.

IEEE 1149.1 Test Access Port Interface

TCK

Test Clock **Input**

TCK is the clock input for the boundary scan test mode operation. It can operate at a frequency of up to 10 MHz. TCK has an internal pull-up resistor. The TCK input operates in the same signaling environment as the PCI bus interface.

TDI

Test Data In **Input**

TDI is the test data input path to the PCnet-PCI II controller. The pin has an internal pull-up resistor. The TDI input operates in the same signaling environment as the PCI bus interface.

TDO

Test Data Out **Output**

TDO is the test data output path from the PCnet-PCI II controller. The pin is tri-stated when the JTAG port is inactive. The TDO output operates in the same signaling environment as the PCI bus interface.

TMS

Test Mode Select **Input**

A serial input bit stream on the TMS pin is used to define the specific boundary scan test to be executed. The pin has an internal pull-up resistor. The TMS input operates in the same signaling environment as the PCI bus interface.

Test Interface

NOUT

NAND Tree Out **Output**

When $\overline{\text{RST}}$ is asserted, the results of the NAND tree testing can be observed on the NOUT pin.

Note that the NOUT pin is multiplexed with the DXCVR pin.

Power Supply Pins

AV_{DD}

Analog Power (4 Pins) **Power**

There are four analog +5 V supply pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to Appendix B and the PCnet Family Board Design and Layout Recommendations application note (PID #19595A) for details.

AV_{SS}

Analog Ground (2 Pins) **Power**

There are two analog ground pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to Appendix B and the PCnet Family Board Design and Layout Recommendations application note (PID #19595A) for details.

V_{DD}

Digital Power (6 Pins) **Power**

There are six power supply pins that are used by the internal digital circuitry. All V_{DD} pins must be connected to a +5 V supply.

V_{DDB}

I/O Buffer Power (4 Pins) **Power**

There are four power supply pins that are used by the PCI bus input/output buffer drivers. In a system with 5 V signaling environment, all VDDB pins must be connected to a +5 V supply. In a system with 3.3 V signaling environment, all V_{DDB} pins must be connected to a +3.3 V supply.

V_{SS}

Digital Ground (12 Pins) **Ground**

There are 12 ground pins that are used by the internal digital circuitry.

V_{SSB}

I/O Buffer Ground (8 Pins) **Ground**

There are 8 ground pins that are used by the PCI bus input/output buffer drivers.

BASIC FUNCTIONS

System Bus Interface Function

The PCnet-PCI II controller is designed to operate as a bus master during normal operations. Some slave I/O accesses to the PCnet-PCI II controller are required in normal operations as well. Initialization of the PCnet-PCI II controller is achieved through a combination of PCI Configuration Space accesses, bus slave accesses, bus master accesses and an optional read of a serial EEPROM that is performed by the PCnet-PCI II controller. The EEPROM read operation is performed through the Microwire interface. The ISO 8802-3 (IEEE/ANSI 802.3) Ethernet Address may reside within the serial EEPROM. Some PCnet-PCI II controller configuration registers may also be programmed by the EEPROM read operation.

The Address PROM, on-chip bus-configuration registers, and the Ethernet controller registers occupy 32 bytes of address space. Both, I/O and memory mapped I/O access are supported. Base Address registers in the PCI configuration space allow locating the address space on a wide variety of starting addresses.

For diskless stations, the PCnet-PCI II controller supports an Expansion ROM of up to 64 Kbytes in size. The host can map the Expansion ROM to any memory address that aligns to a 64K boundary by modifying the Expansion ROM Base Address register in the PCI configuration space.

Software Interface

The software interface to the PCnet-PCI II controller is divided into three parts. One part is the PCI configuration registers. They are used to identify the PCnet-PCI II controller, and are also used to setup the configuration of the device. The setup information includes the I/O or memory mapped I/O base address, mapping of the Expansion ROM and the routing of the PCnet-PCI II controller interrupt channel. This allows for a jumperless implementation.

The second portion of the software interface is the direct access to the I/O resources of the PCnet-PCI II controller. The PCnet-PCI II controller occupies 32 bytes of address space that must begin on a 32-byte block boundary. The address space can be mapped into both I/O or memory space (memory mapped I/O). The I/O Base Address Register in the PCI Configuration Space defines the start address of the address space if it is mapped to I/O space. The Memory Mapped I/O Base Address Register defines the start address of the address space if it is mapped to memory space. The 32-byte address space is used by the software to program the PCnet-PCI II controller operating mode, enable and disable various features, monitor operating status, and request particular functions to be executed by the PCnet-PCI II controller.

The third portion of the software interface is the descriptor and buffer areas that are shared between the software and the PCnet-PCI II controller during normal network operations. The descriptor area boundaries are set by the software and do not change during normal network operations. There is one descriptor area for receive activity and there is a separate area for transmit activity. The descriptor space contains relocatable pointers to the network frame data and it is used to transfer frame status from the PCnet-PCI II controller to the software. The buffer areas are locations that hold frame data for transmission or that accept frame data that has been received.

Network Interfaces

The PCnet-PCI II controller can be connected to an 802.3 network via one of three network interfaces. The Attachment Unit Interface (AUI) provides an ISO 8802-3 (IEEE/ANSI 802.3) compliant differential interface to a remote MAU or an on-board transceiver. The 10BASE-T interface provides a twisted-pair Ethernet port. While in auto-selection mode, the interface in use is determined by an auto-sensing mechanism which checks the link status on the 10BASE-T port. If there is no active link status, then the device assumes an AUI connection.

The PCnet-PCI II controller implements half or full-duplex Ethernet over all three network interfaces.

DETAILED FUNCTIONS

Slave Bus Interface Unit

The slave bus interface unit (BIU) controls all accesses to the PCI configuration space, the Control and Status Registers (CSR), the Bus Configuration Registers (BCR), the Address PROM (APROM) locations and the Expansion ROM. The table below shows the response of the PCnet-PCI II controller to each of the PCI commands in slave mode.

Table 2. Slave Commands

C[3:0]	Command	Use
0000	Interrupt Acknowledge	Not used
0001	Special Cycle	Not used
0010	I/O Read	Read of CSR, BCR, APROM
0011	I/O Write	Write to CSR, BCR, and APROM
0100	Reserved	
0101	Reserved	
0110	Memory Read	Memory mapped I/O read of CSR, BCR, APROM Read of the Expansion ROM
0111	Memory Write	Memory mapped I/O write of CSR, BCR, and APROM Dummy Write to the Expansion ROM
1000	Reserved	
1001	Reserved	
1010	Configuration Read	Read of the Configuration Space
1011	Configuration Write	Write to the Configuration Space
1100	Memory Read Multiple	Aliased to Memory Read
1101	Dual Address Cycle	Not used
1110	Memory Read Line	Aliased to Memory Read
1111	Memory Write Invalidate	Aliased to Memory Write

Slave Configuration Transfers

The host can access the PCnet-PCI II controller PCI configuration space with a configuration read or write command. The PCnet-PCI II controller will assert DEVSEL during the address phase when IDSEL is asserted, AD[1:0] are both ZERO, and the access is a configuration cycle. AD[7:2] select the DWord location in the configuration space. The PCnet-PCI II controller ignores AD[10:8], because it is a single function device. AD[31:11] are don't care.

The active bytes within a DWord are determined by the byte enable signals. 8-bit, 16-bit and 32-bit transfers are supported. DEVSEL is asserted two clock cycles after the host has asserted FRAME. All configuration cycles are of fixed length. The PCnet-PCI II controller will assert TRDY on the 4th clock of the data phase.

The PCnet-PCI II controller does not support burst transfers for access to configuration space. When the host keeps FRAME asserted for a second data phase, the PCnet-PCI II controller will disconnect the transfer.

When the host tries to access the PCI configuration space while the automatic read of the EEPROM after H_RESET is on-going, the PCnet-PCI II controller will terminate the access on the PCI bus with a disconnect/ retry response.

The PCnet-PCI II controller supports fast back-to-back transactions to different targets. This is indicated by the Fast Back-To-Back Capable bit (PCI Status register, bit 7), which is hardwired to ONE. The PCnet-PCI II controller is capable of detecting a configuration cycle even when its address phase immediately follows the data phase of a transaction to a different target without any idle state in-between. There will be no contention on the DEVSEL, TRDY and STOP signals, since the PCnet-PCI II controller asserts DEVSEL on the second clock after FRAME is asserted (medium timing).

AD31 — AD11	AD10 — AD8	AD7 — AD2	AD1	AD0
Don't care	Don't care	DWord index	0	0

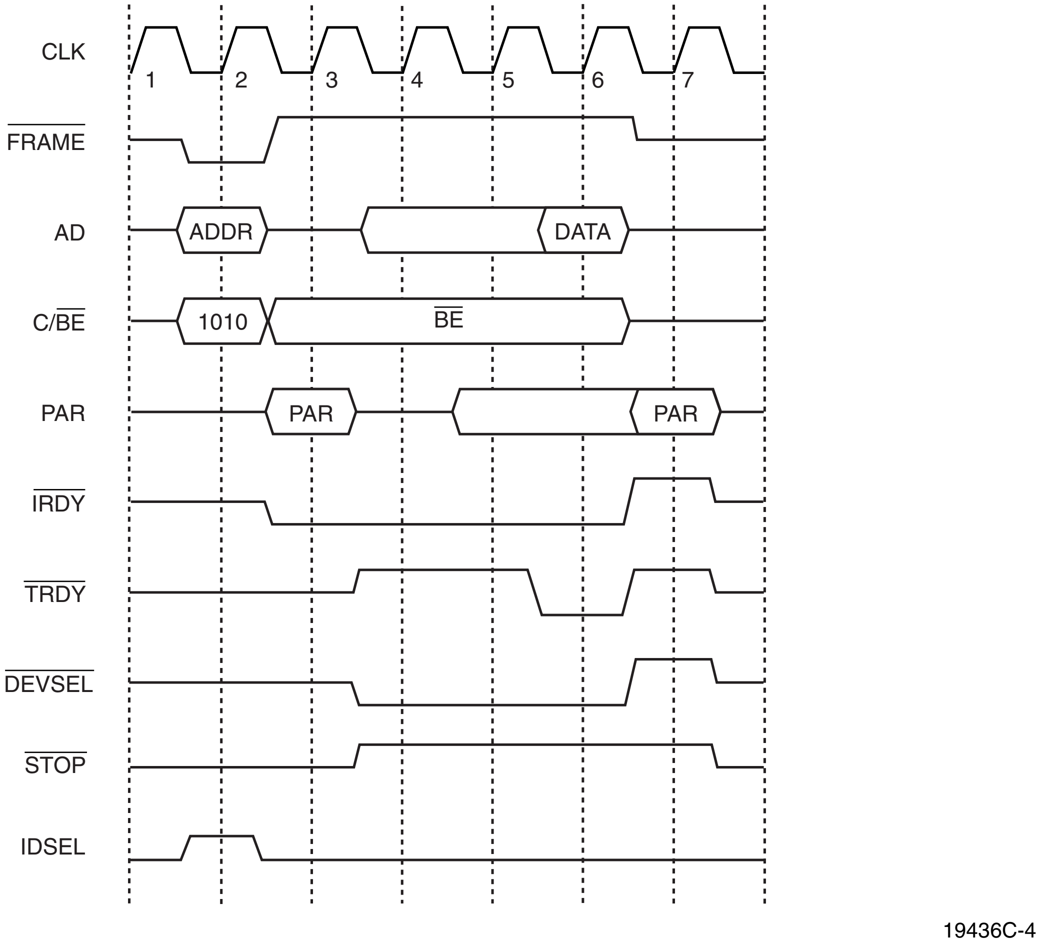
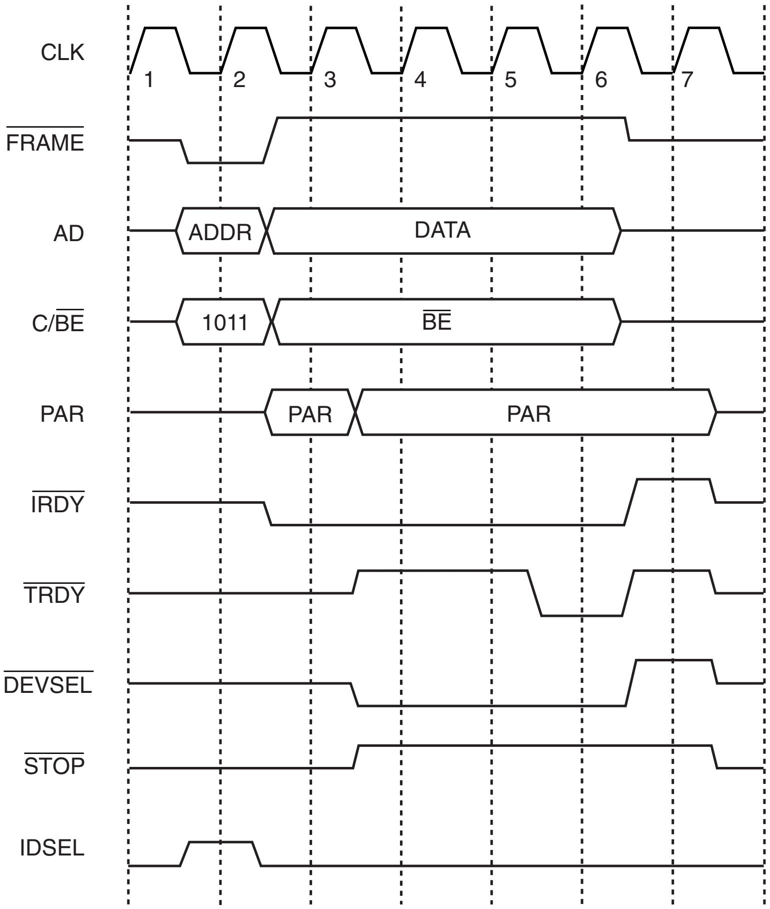


Figure 1. Slave Configuration Read



19436C-5

Figure 2. Slave Configuration Write

Slave I/O Transfers

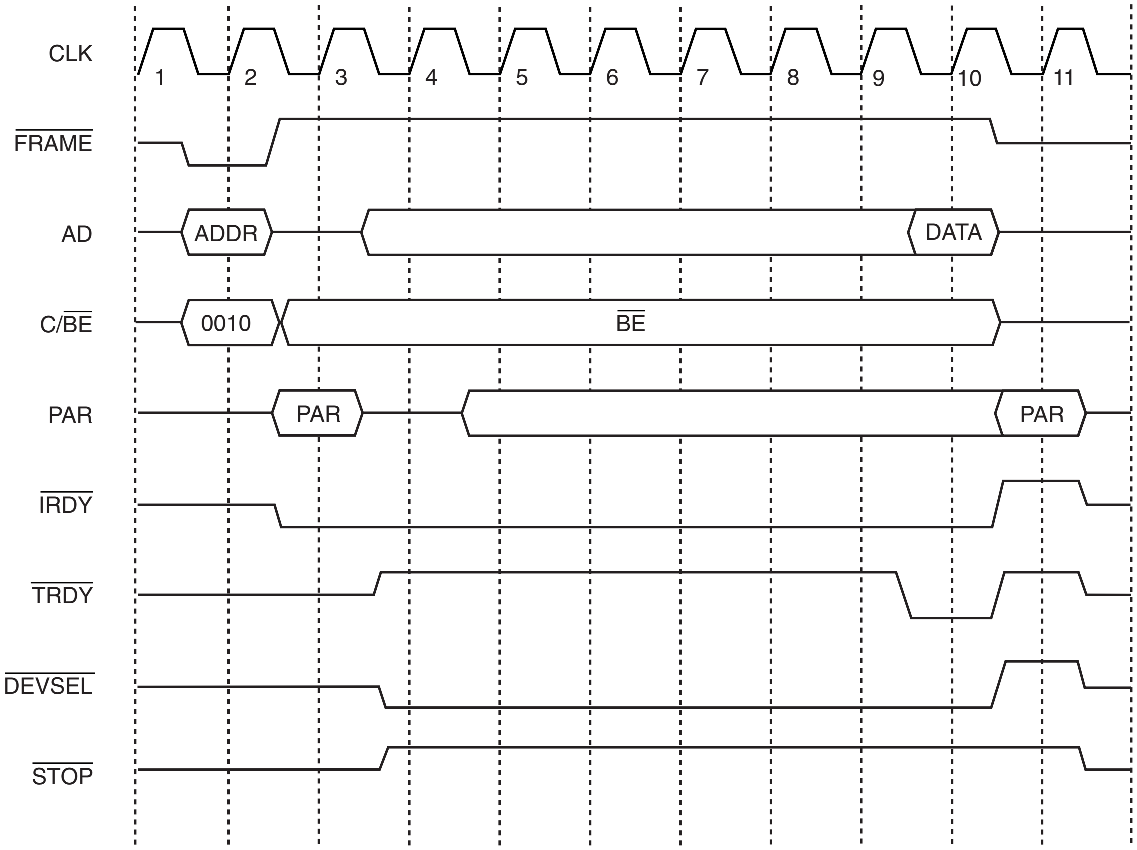
After the PCnet-PCI II controller is configured as an I/O device by setting IOEN (for regular I/O mode) or MEMEN (for memory mapped I/O mode) in the PCI Command register, it starts monitoring the PCI bus for access to its CSR, BCR or EEPROM locations. If configured for regular I/O mode, the PCnet-PCI II controller will look for an address that falls within its 32 bytes of I/O address space (starting from the I/O base address). The PCnet-PCI II controller asserts DEVSEL if it detects an address match and the access is an I/O cycle. If configured for memory mapped I/O mode, the PCnet-PCI II controller will look for an address that falls within its 32 bytes of memory address space (starting from the memory mapped I/O base address). The PCnet-PCI II controller asserts DEVSEL if it detects an address match and the access is a memory cycle. DEVSEL is asserted two clock cycles after the host has asserted FRAME. The PCnet-PCI II controller will not assert DEVSEL if it detects an address match, but the PCI command is not of the correct type. In memory mapped I/O mode, the PCnet-PCI II controller aliases all accesses to the I/O resources of the command types “Memory Read Multiple” and “Memory Read Line” to the basic Memory Read command. All accesses of the type “Memory Write and Invalidate” are aliased to the basic Memory Write command. 8-bit, 16-bit and 32-bit non-burst transactions are supported. The PCnet-PCI II controller decodes only the upper 30 address lines to determine which I/O resource is accessed.

The typical number of wait states added to a slave I/O or memory mapped I/O read or write access on the part of the PCnet-PCI II controller is 6 to 7 clock cycles, depending upon the relative phases of the internal Buffer Management Unit clock and the CLK signal, since the internal Buffer Management Unit clock is a divide-by-two version of the CLK signal.

The PCnet-PCI II controller does not support burst transfers for access to its I/O resources. When the host keeps FRAME asserted for a second data phase, the PCnet-PCI II controller will disconnect the transfer.

The PCnet-PCI II controller supports fast back-to-back transactions to different targets. This is indicated by the Fast Back-To-Back Capable bit (PCI Status register, bit

7), which is hardwired to ONE. The PCnet-PCI II controller is capable of detecting an I/O or a memory mapped I/O cycle even when its address phase immediately follows the data phase of a transaction to a different target, without any idle state in-between. There will be no contention on the $\overline{\text{DEVSEL}}$, $\overline{\text{TRDY}}$ and $\overline{\text{STOP}}$ signals, since the PCnet-PCI II controller asserts $\overline{\text{DEVSEL}}$ on the second clock after $\overline{\text{FRAME}}$ is asserted (medium timing).



19436C-6

Figure 3. Slave Read Using I/O Command

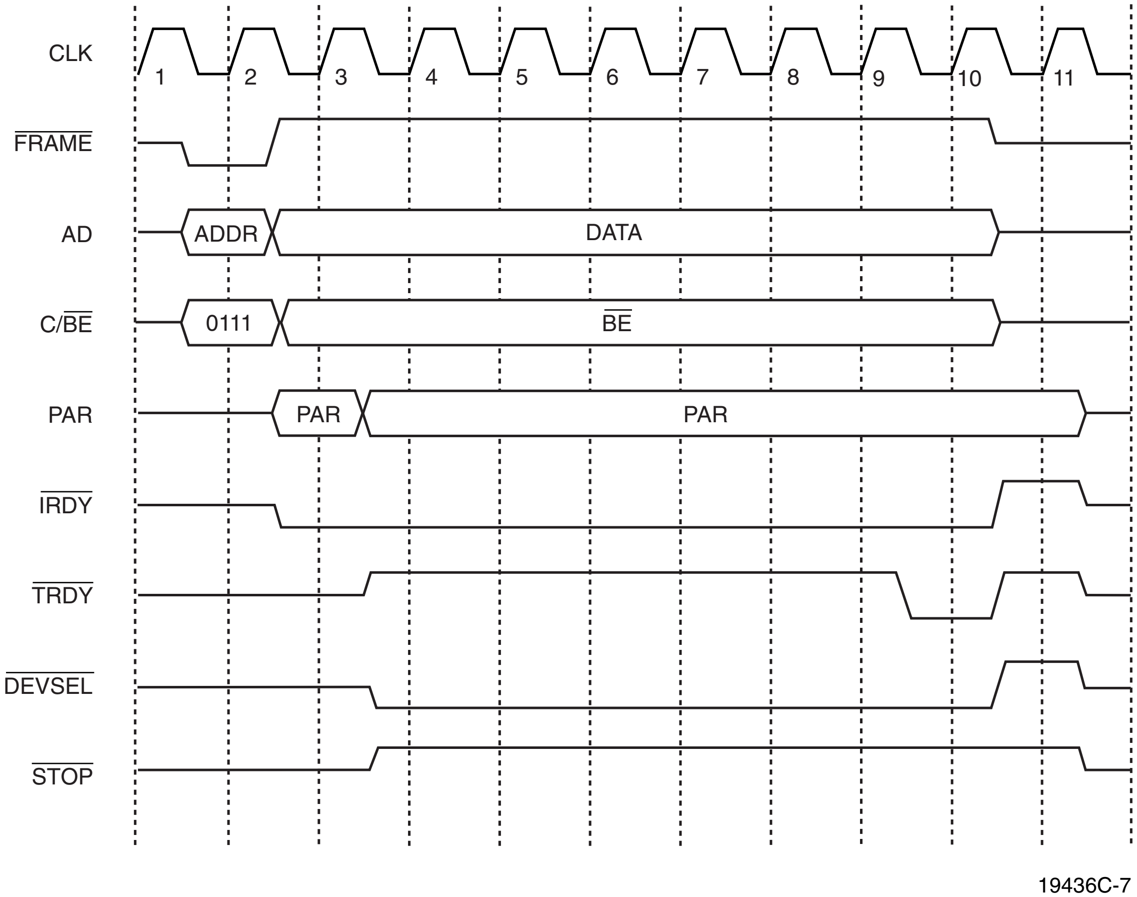


Figure 4. Slave Write Using Memory Command

Expansion ROM Transfers

The host must initialize the Expansion ROM Base Address register at offset 30h in the PCI configuration space with a valid address before enabling the access to the device. The base address must be aligned to a 64K boundary as indicated by ROMSIZE (PCI Expansion ROM Base Address register, bits 15–11). The PCnet-PCI II controller will not react to any access to the Expansion ROM until both MEMEN (PCI Command register, bit 1) and ROMEN (PCI Expansion ROM Base Address register, bit 0) are set to ONE. After the Expansion ROM is enabled, the PCnet-PCI II controller will assert DEVSEL on all memory read accesses with an address between ROMBASE and ROMBASE + 64K – 4. The PCnet-PCI II controller aliases all accesses to the Expansion ROM of the command types “Memory Read Multiple” and “Memory Read Line” to the basic Memory Read command. Eight-bit, 16-bit and 32-bit read transfers are supported.

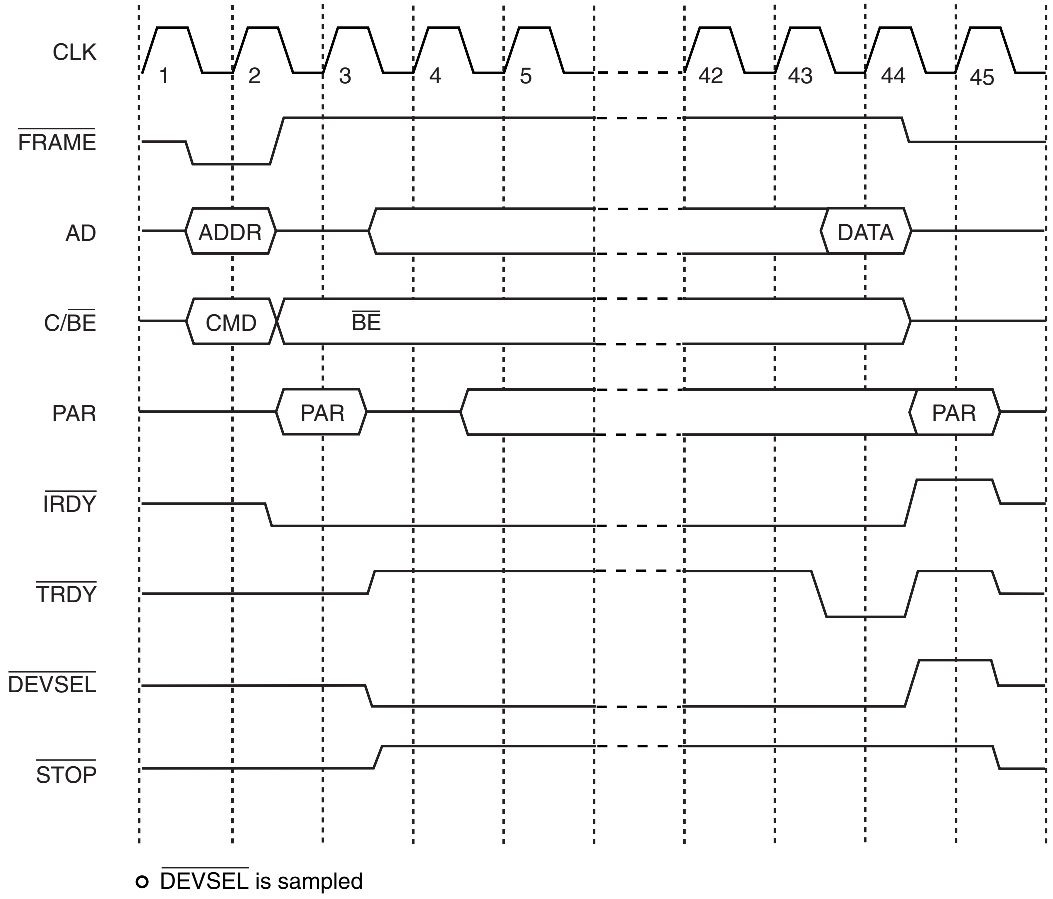
Since setting MEMEN also enables memory mapped access to the I/O resources, attention must be given the PCI Memory Mapped I/O Base Address register, before enabling access to the Expansion ROM. The host must set the PCI Memory Mapped I/O Base Address register to a value that prevents the PCnet-PCI II controller from claiming any memory cycles not intended for it.

The PCnet-PCI II controller will always read four bytes for every host Expansion ROM read access. TRDY will not be asserted until all four bytes are loaded into an internal scratch register. The cycle TRDY is asserted depends on the programming of the Expansion ROM interface timing. The following figure assumes that ROMTMG (BCR18, bits 15–12) is at its default value. Since the target latency for the Expansion ROM access is considerably long, the PCnet-PCI II controller disconnects at the second data phase, when the host tries do to perform a burst read operation of the Expansion ROM. This behavior complies with the requirements for latency issues in the PCI environment and allows other devices to get fair access to the bus.

When the host tries to write to the Expansion ROM, the PCnet-PCI II controller will claim the cycle by asserting DEVSEL. TRDY will be asserted one clock cycle later. The write operation will have no effect.

The PCnet-PCI II controller supports fast back-to-back transactions to different targets. This is indicated by the

Fast Back-To-Back Capable bit (PCI Status register, bit 7), which is hardwired to ONE. The PCnet-PCI II controller is capable of detecting a memory cycle even when its address phase immediately follows the data phase of a transaction to a different target without any idle state in-between. There will be no contention on the $\overline{\text{DEVSEL}}$, $\overline{\text{TRDY}}$ and $\overline{\text{STOP}}$ signals, since the PCnet-PCI II controller asserts $\overline{\text{DEVSEL}}$ on the second clock after $\overline{\text{FRAME}}$ is asserted (medium timing).



19436C-8

Figure 5. Expansion ROM Read

Exclusive Access

The host can lock a set of transactions to the PCnet-PCI II controller. The lock allows exclusive access to the device and can be used to guarantee atomic operations. The PCnet-PCI II controller transitions from the unlocked to the locked state when $\overline{\text{LOCK}}$ is deasserted during the address phase of a transaction that selects the device as the target. The controller stays in the locked state until both $\overline{\text{FRAME}}$ and $\overline{\text{LOCK}}$ are deasserted, or until the device signals a target abort. Note that this protocol means the device locks itself on any normal transaction. The controller will unlock automatically at the end of a normal transaction, because $\overline{\text{FRAME}}$ and $\overline{\text{LOCK}}$ will be deasserted. The lock spans over the whole slave address space. The lock only applies to slave accesses. The PCnet-PCI II controller might perform bus master cycles while being locked in slave mode. When another master tries to access the PCnet-PCI II controller while it is in the locked state, the device terminates the access with a disconnect/retry sequence.

Slave Cycle Termination

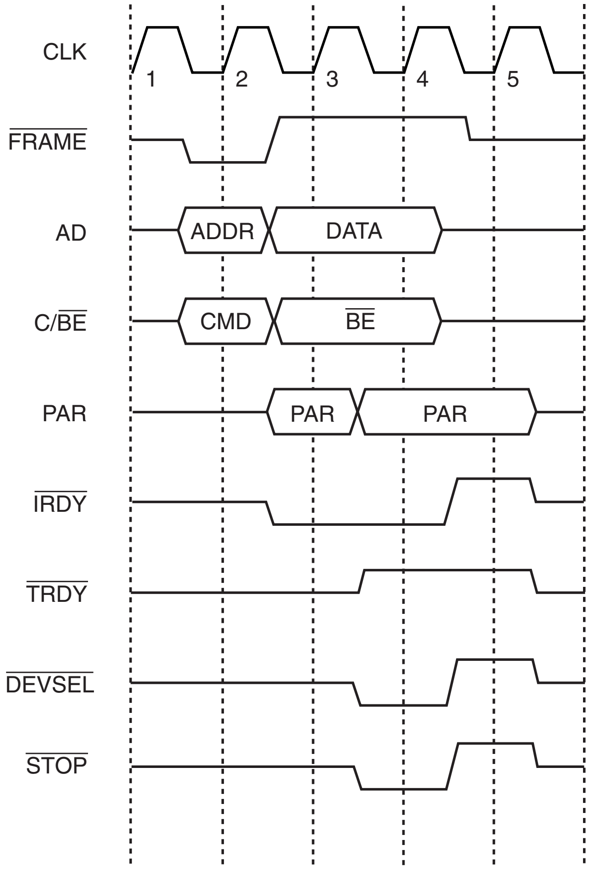
There are three scenarios besides normal completion of a transaction where the PCnet-PCI II controller is the target of a slave cycle and it will terminate the access.

Disconnect When Busy

The PCnet-PCI II controller cannot service any slave access while it is reading the contents of the Microwire EEPROM. Simultaneous access is not possible to avoid conflicts, since the Microwire EEPROM is used to initialize some of the PCI configuration space locations and most of the BCRs. The Microwire EEPROM read operation will always happen automatically after the deassertion of the RST pin. In addition, the host can start the read operation by setting the PREAD bit (BCR19, bit 14). While the EEPROM read is on-going, the PCnet-PCI II controller will disconnect any slave access where it is the target by asserting STOP together with DEVSEL, while driving TRDY high. STOP will stay asserted until the host removes FRAME.

Note that I/O and memory slave accesses will only be disconnected if they are enabled by setting the IOEN or MEMEN bit in the PCI Command register. Without the enable bit set, the cycles will not be claimed at all. Since H_RESET clears the IOEN and MEMEN bits, for the automatic EEPROM read after H_RESET the disconnect only applies to configuration cycles.

A second situation where the PCnet-PCI II controller will generate a PCI disconnect/retry cycle is when the host tries to access any of the I/O resources right after having read the Reset register. Since the access generates an internal reset pulse of about 1 µs in length, all further slave accesses will be deferred until the internal reset operation is completed.



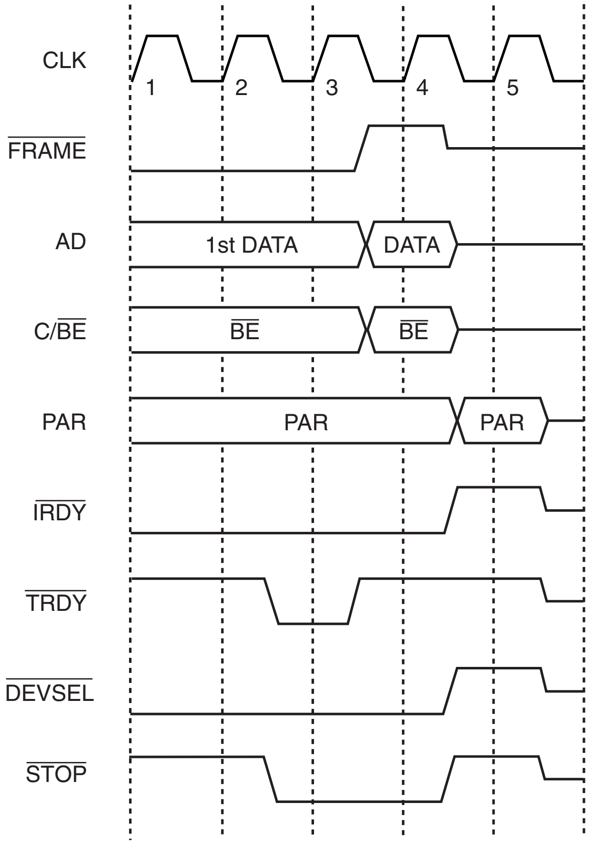
19436C-9

Figure 6. Disconnect Of Slave Cycle When Busy

Disconnect Of Burst Transfer

The PCnet-PCI II controller does not support burst access to the configuration space, the I/O resources, or to the Expansion ROM. The host indicates a burst transaction by keeping FRAME asserted during the data phase.

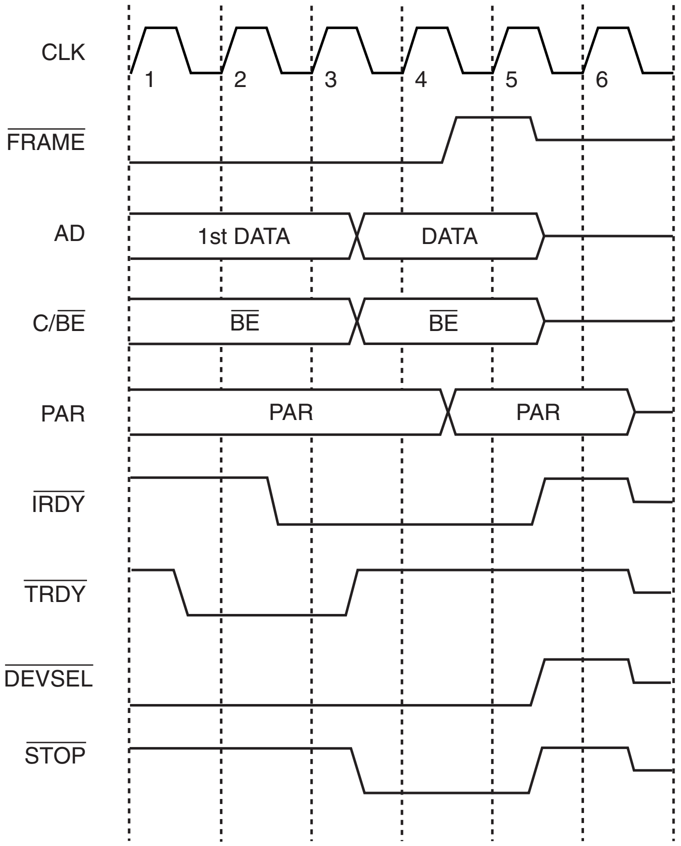
When the PCnet-PCI II controller sees FRAME and IRDY asserted in the clock cycle before it wants to asserts TRDY, it also asserts STOP at the same time. The transfer of the first data phase is still successful, since IRDY and TRDY are both asserted.



19436C-10

Figure 7. Disconnect Of Slave Burst Transfer—No Host Wait States

When the host is not yet ready when the PCnet-PCI II controller asserts TRDY, the device will wait for the host to assert IRDY. When the host asserts IRDY and FRAME is still asserted, the PCnet-PCI II controller will finish the first data phase by deasserting TRDY one clock later. At the same time, it will assert STOP to signal a disconnect to the host. STOP will stay asserted until the host removes FRAME.



19436C-11

Figure 8. Disconnect Of Slave Burst Transfer—Host Inserts Wait States

Disconnect When Locked

When the PCnet-PCI II controller is locked by one master and another master tries to access the controller, the device will disconnect the access. When the PCnet-PCI II controller is in the locked state and it sees $\overline{\text{LOCK}}$ asserted together with $\overline{\text{FRAME}}$, it knows that another master tried to access it. The PCnet-PCI II controller will respond to the access by asserting $\overline{\text{STOP}}$ together with $\overline{\text{DEVSEL}}$ while driving $\overline{\text{TRDY}}$ high, thereby disconnecting the cycle. $\overline{\text{STOP}}$ will stay asserted until the other master removes $\overline{\text{FRAME}}$.

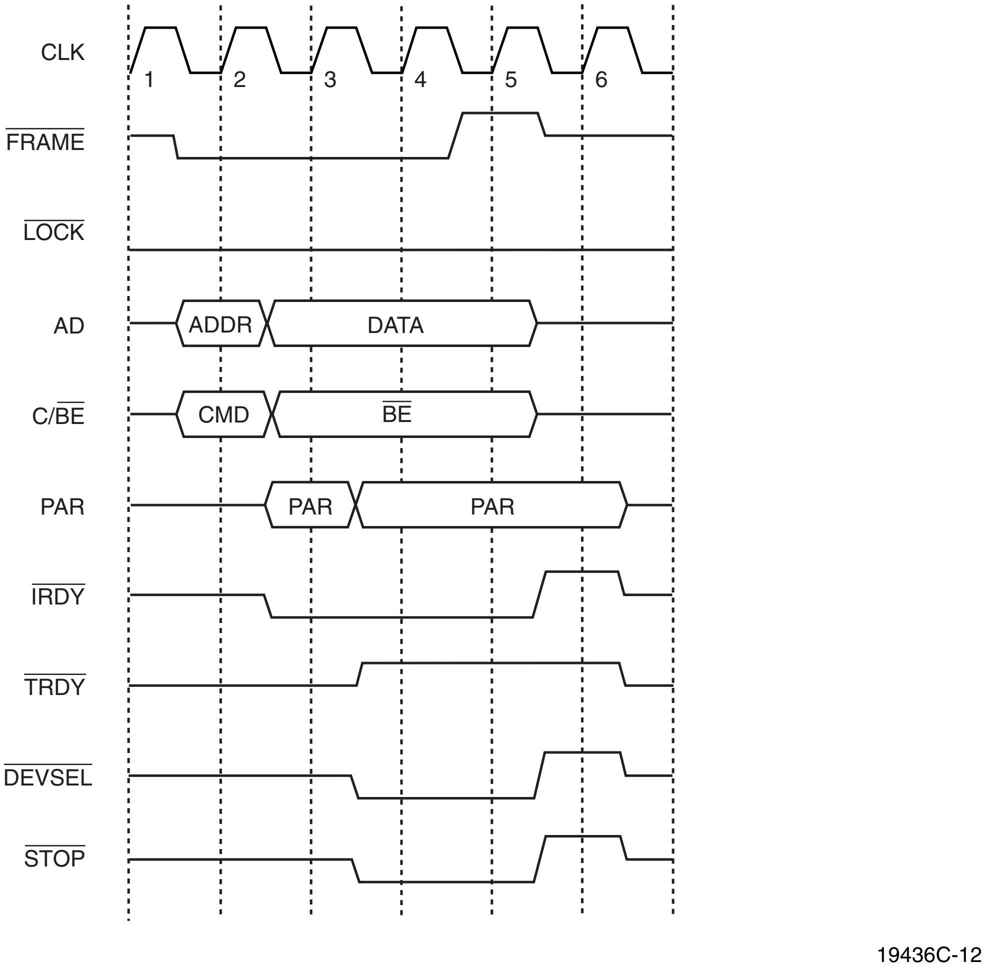


Figure 9. Disconnect Of Slave Cycle When Locked

Parity Error Response

When the PCnet-PCI II controller is not the current bus master, it samples the AD[31:0], C/$\overline{\text{BE}}$[3:0] and the PAR lines during the address phase of any PCI command for a parity error. When it detects an address parity error, the controller sets PERR (PCI Status register, bit 15) to ONE. When reporting of that error is enabled by setting SERREN (PCI Command register, bit 8) and PERREN (PCI Command register, bit 6) to ONE, the Pcnet-PCI II controller also drives the SERR signal low for one clock cycle and sets $\overline{\text{SERR}}$ (PCI Status register, bit 14) to ONE. The assertion of $\overline{\text{SERR}}$ follows the address phase by two clock cycles. The PCnet-PCI II controller will not assert $\overline{\text{DEVSEL}}$ for a PCI transaction that has an address parity error, when PERREN and SERREN are set to ONE.

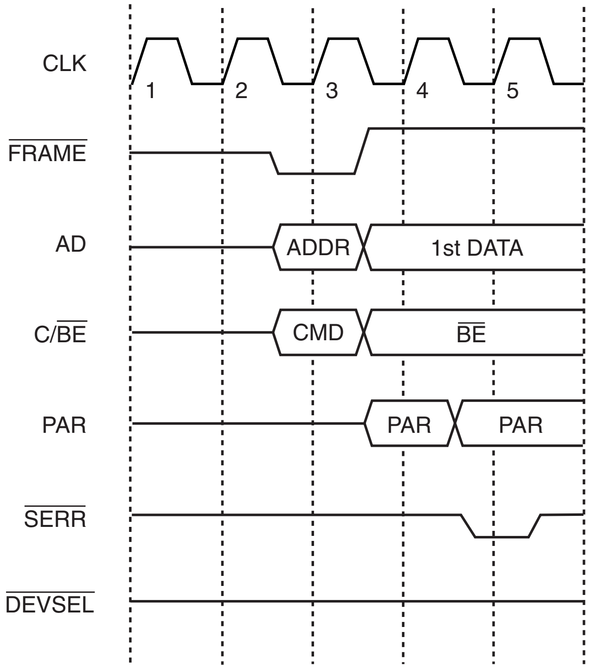


Figure 10. Address Parity Error Response

During the data phase of an I/O write, memory mapped I/O write or configuration write command that selects the PCnet-PCI II controller as target, the device samples the AD[31:0] and C/$\overline{\text{BE}}$[3:0] lines for parity on the clock edge data is transferred. PAR is sampled in the following clock cycle. If a parity error is detected and reporting of that error is enabled by setting PERREN (PCI Command register, bit 6) to ONE, $\overline{\text{PERR}}$ is asserted one clock later. The parity error will always set PERR (PCI Status register, bit 15) to ONE even when PERREN is cleared to ZERO. The PCnet-PCI II controller will finish a transaction that has a data parity error in the normal way by asserting $\overline{\text{TRDY}}$. The corrupted data will be written to the addressed location.

Figure 11 shows a transaction that suffered a parity error at the time data was transferred (clock 7, $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both asserted). $\overline{\text{PERR}}$ is driven high at the beginning of the data phase and then drops low due to the parity error on clock 9, two clock cycles after the data was transferred. After $\overline{\text{PERR}}$ is driven low, the PCnet-PCI II controller drives $\overline{\text{PERR}}$ high for one clock cycle, since $\overline{\text{PERR}}$ is a sustained tri-state signal.

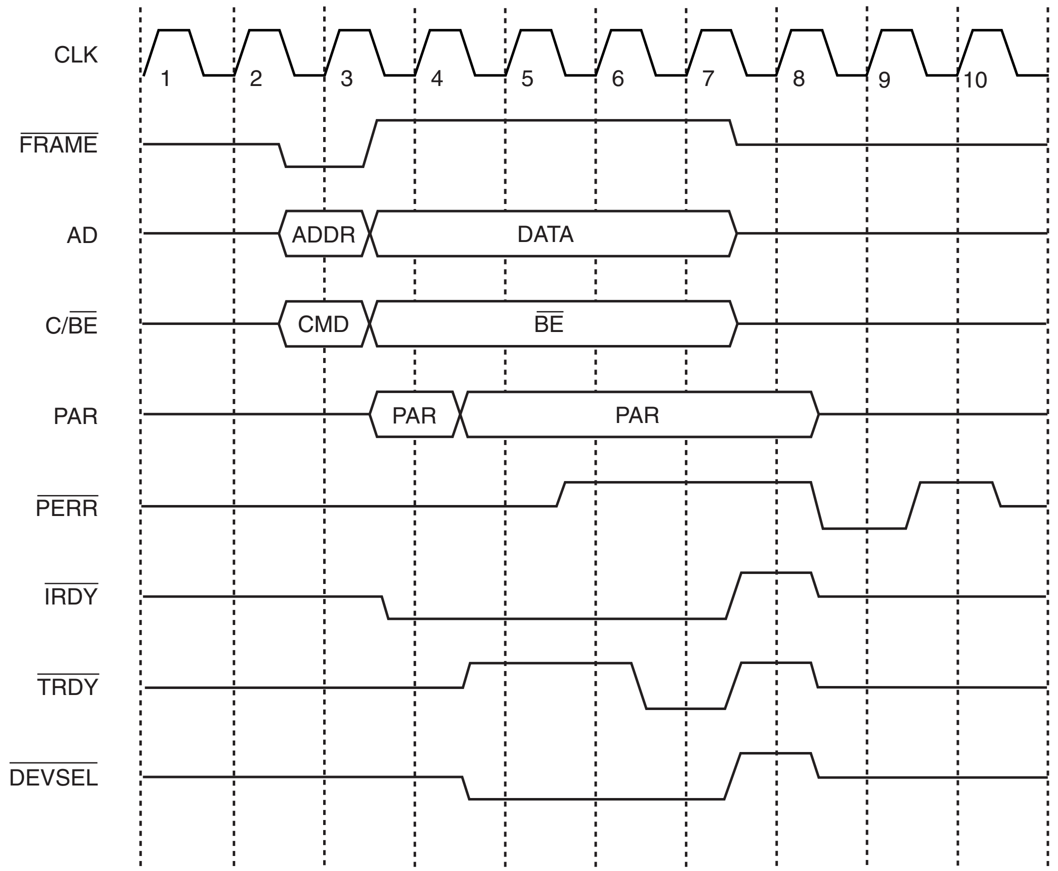


Figure 11. Slave Cycle Data Parity Error Response

Master Bus Interface Unit

The master bus interface unit (BIU) controls the acquisition of the PCI bus and all accesses to the initialization block, descriptor rings and the receive and transmit buffer memory. The table below shows the usage of PCI commands by the PCnet-PCI II controller in master mode.

Table 3. Master Commands

C[3:0]	Command	Use
0000	Interrupt Acknowledge	Not used
0001	Special Cycle	Not used
0010	I/O Read	Not used
0011	I/O Write	Not used
0100	Reserved	
0101	Reserved	
0110	Memory Read	Read of the Initialization Block and Descriptor Rings Read of the Transmit Buffer in Non-burst Mode
0111	Memory Write	Write to the Descriptor Rings and to the Receive Buffer
1000	Reserved	
1001	Reserved	
1010	Configuration Read	Not used
1011	Configuration Write	Not used
1100	Memory Read Multiple	Read of the Transmit Buffer in Burst Mode
1101	Dual Address Cycle	Not used
1110	Memory Read Line	Read of the Transmit Buffer in Burst Mode
1111	Memory Write Invalidate	Not used

Bus Acquisition

The PCnet-PCI II controller microcode will determine when a DMA transfer should be initiated. The first step in any PCnet-PCI II controller bus master transfer is to acquire ownership of the bus. This task is handled by synchronous logic within the BIU. Bus ownership is requested with the REQ signal and ownership is granted by the arbiter through the GNT signal.

Figure 12 shows the PCnet-PCI II controller bus acquisition. REQ is asserted and the arbiter returns GNT while another bus master is transferring data. The PCnet-PCI II controller waits until the bus is idle (FRAME and IRDY deasserted) before it starts driving AD[31:0] and C/BE[3:0] on clock 5. FRAME is asserted at clock 5 indicating a valid address and command on AD[31:0] and C/BE[3:0]. The PCnet-PCI II controller does not use address stepping which is reflected by ADSTEP (bit 7) in the PCI Command register being hardwired to ZERO.

In burst mode, the deassertion of REQ depends on the setting of EXTREQ (BCR18, bit 8). If EXTREQ is cleared to ZERO, REQ is deasserted at the same time as FRAME is asserted. (The PCnet-PCI II controller never performs more than one burst transaction within a single bus mastership period). If EXTREQ is set to ONE, the PCnet-PCI II controller does not deassert REQ until it starts the last data phase of the transaction.

Once asserted, REQ remains active until GNT has become active, independent of subsequent setting of the STOP (CSR0, bit 2) or SPND (CSR5, bit 0). The assertion of H_RESET or S_RESET, however, will cause REQ to go inactive immediately.

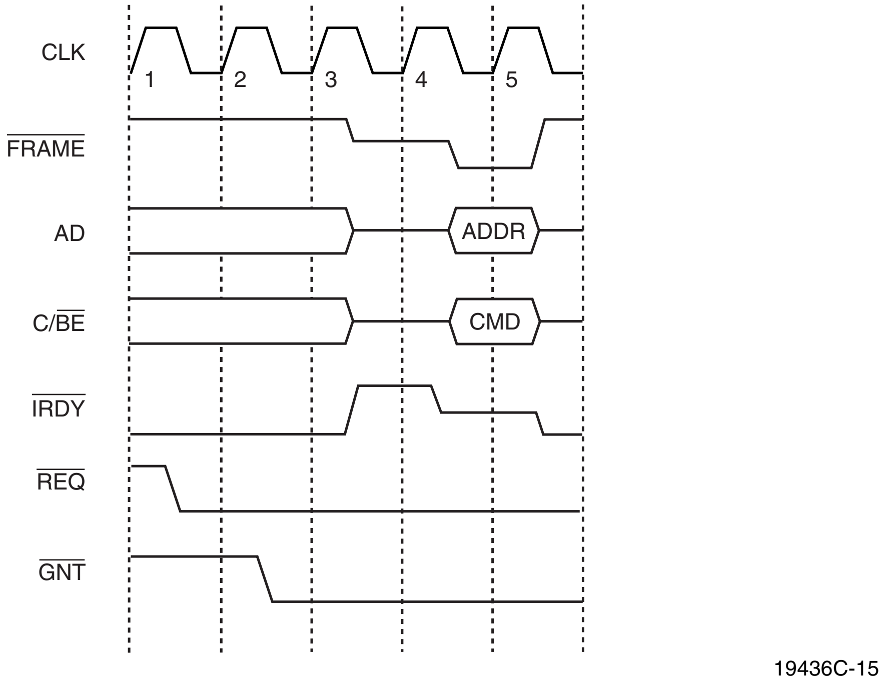


Figure 12. Bus Acquisition

Bus Master DMA Transfers

There are four primary types of DMA transfers. The PCnet-PCI II controller uses non-burst as well as burst cycles for read and write access to the main memory.

Basic Non-Burst Read Transfer

By default, the PCnet-PCI II controller uses non-burst cycles in all bus master read operations. All PCnet-PCI II controller non-burst read accesses are of the PCI command type Memory Read (type 6). Note that during a non-burst read operation, all byte lanes will always be active. The PCnet-PCI II controller will internally discard unneeded bytes.

The PCnet-PCI II controller typically performs more than one non-burst read transactions within a single bus mastership period. FRAME is dropped between consecutive non-burst read cycles. REQ however stays asserted until FRAME is asserted for the last transaction. The PCnet-PCI II controller supports zero wait state read cycles. It asserts IRDY immediately after the address phase and at the same time starts sampling DEVSEL.

The following figure shows two non-burst read transactions. The first transaction has zero wait states. In the second transaction, the target extends the cycle by asserting $\overline{\text{TRDY}}$ one clock later.

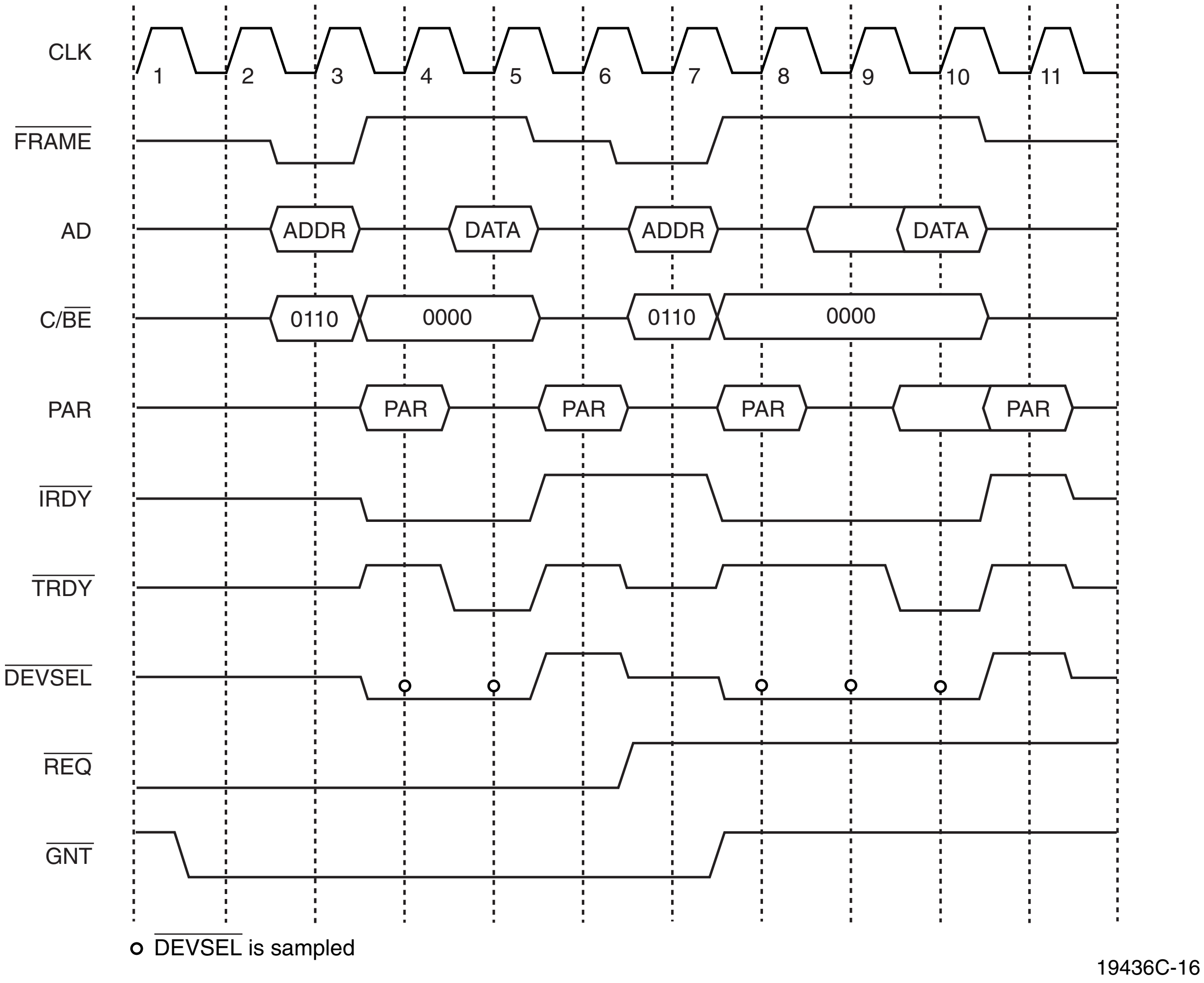


Figure 13. Non-Burst Read Transfer

Basic Burst Read Transfer

The PCnet-PCI II controller supports burst mode for all bus master read operations. The burst mode must be enabled by setting BREADE (BCR18, bit 6). To allow burst transfers in descriptor read operations, the PCnet-PCI II controller must also be programmed to use SWSTYLE THREE (BCR20, bits 7–0). All burst read accesses to the initialization block and descriptor ring are of the PCI command type Memory Read (type 6). Burst read accesses to the transmit buffer typically are longer than two data phases. When MEMCMD (BCR18, bit 9) is cleared to ZERO, all burst read accesses to the transmit buffer are of the PCI command type Memory Read Line (type 14). When MEMCMD (BCR18, bit 9) is set to ONE, all burst read accesses to the transmit buffer are of the PCI command type Memory Read Multiple (type 12). AD[1:0] will both be ZERO during the address phase indicating a linear burst order. Note that during a burst read operation, all byte lanes will always be active. The PCnet-PCI II controller will internally discard unneeded bytes.

The PCnet-PCI II controller will always perform only a single burst read transaction per bus mastership period, where transaction is defined as one address phase and one or multiple data phases. The PCnet-PCI II controller supports zero wait state read cycles. It asserts $\overline{\text{IRDY}}$ immediately after the address phase and at the same time starts sampling $\overline{\text{DEVSEL}}$. $\overline{\text{FRAME}}$ is deasserted when the next to last data phase is completed.

The following figure shows a typical burst read access. The PCnet-PCI II controller arbitrates for the bus, is granted access, and reads three 32-bit words (DWord) from the system memory and then releases the bus. In the example, the memory system extends the data phase of the each access by one wait state. The example assumes that EXTREQ (BCR18, bit 8) is cleared to ZERO, therefore, $\overline{\text{REQ}}$ is deasserted in the same cycle as $\overline{\text{FRAME}}$ is asserted.

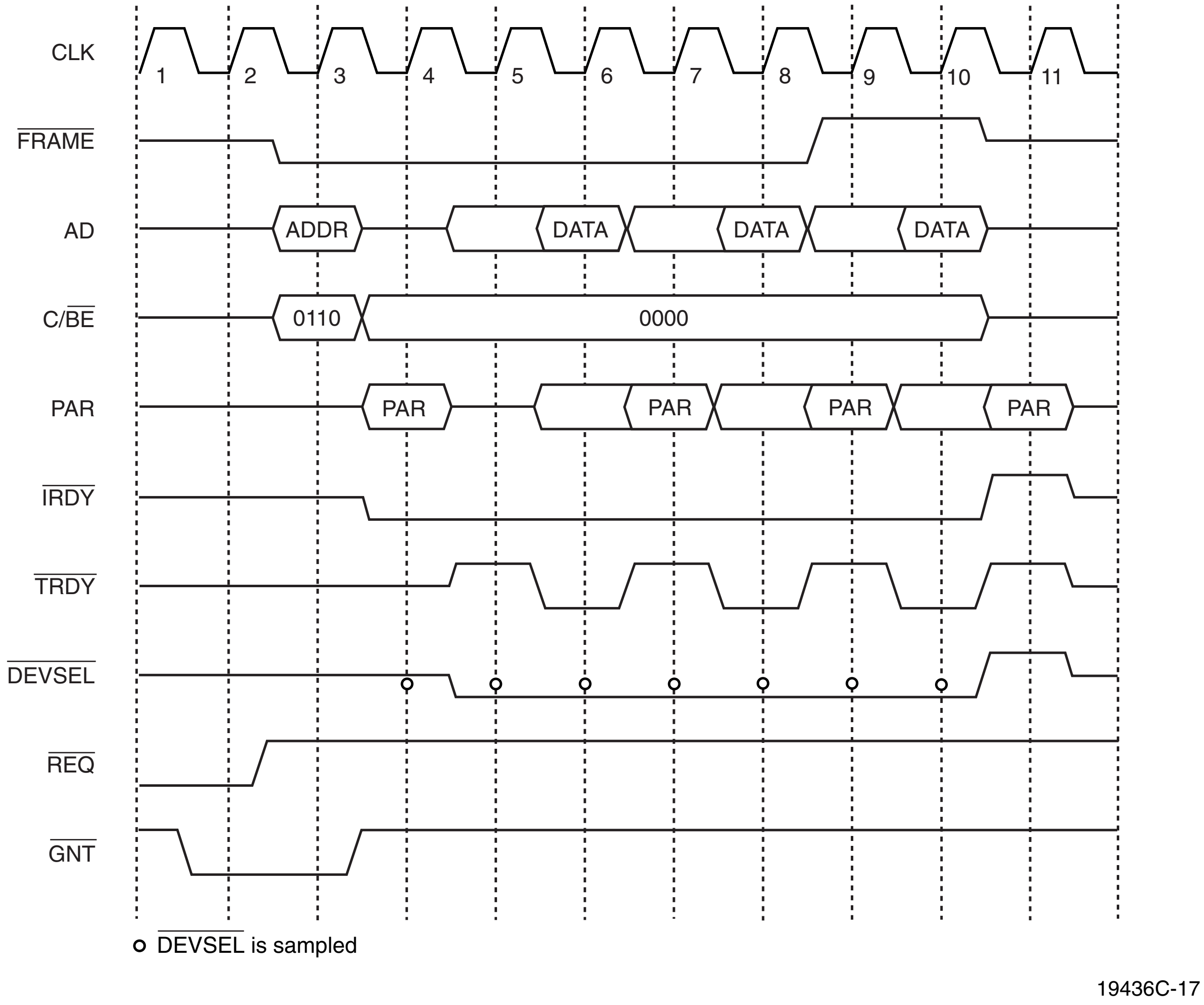


Figure 14. Burst Read Transfer (EXTREQ = 0, MEMCMD = 0)

Basic Non-Burst Write Transfer

By default, the PCnet-PCI II controller uses non-burst cycles in all bus master write operations. All PCnet-PCI II controller non-burst write accesses are of the PCI command type Memory Write (type 7). The byte enable signals indicate the byte lanes that have valid data.

The PCnet-PCI II controller typically performs more than one non-burst write transactions within a single bus mastership period. $\overline{\text{FRAME}}$ is dropped between consecutive non-burst write cycles. $\overline{\text{REQ}}$ however stays asserted until $\overline{\text{FRAME}}$ is asserted for the last transaction. The PCnet-PCI II controller supports zero wait state write cycles except with the case of descriptor write transfers. (See the section “Descriptor DMA Transfers” for the only exception.) It asserts $\overline{\text{IRDY}}$ immediately after the address phase and at the same time starts sampling $\overline{\text{DEVSEL}}$.

The following figure shows two non-burst write transactions. The first transaction has two wait states. The target inserts one wait state by asserting DEVSEL one clock late and another wait state by also asserting TRDY one clock late. The second transaction shows a zero wait state write cycle. The target asserts DEVSEL and TRDY in the same cycle as the PCnet-PCI II controller asserts IRDY.

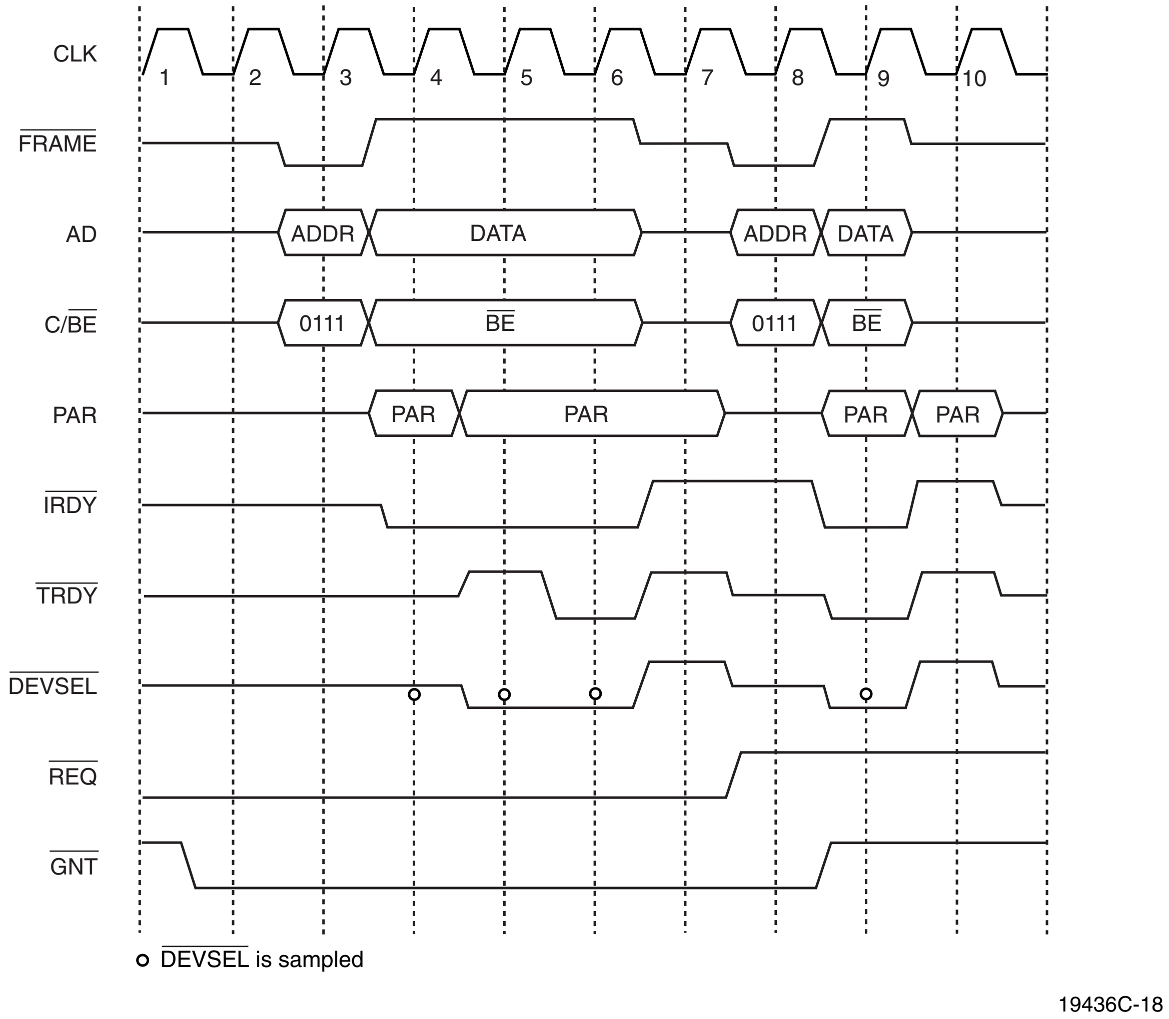


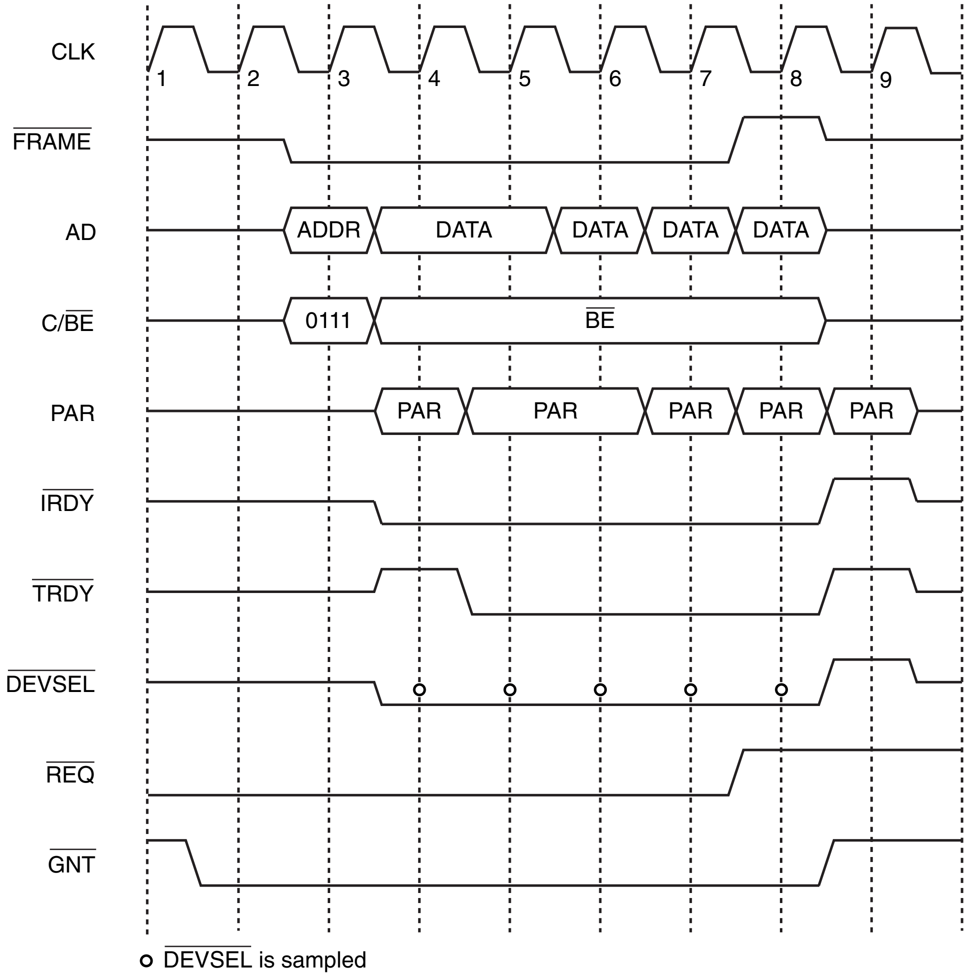
Figure 15. Non-Burst Write Transfer

Basic Burst Write Transfer

The PCnet-PCI II controller supports burst mode for all bus master write operations. The burst mode must be enabled by setting BWRITE (BCR18, bit 5). To allow burst transfers in descriptor write operations, the PCnet-PCI II controller must also be programmed to use SWSTYLE THREE (BCR20, bits 7–0). All PCnet-PCI II controller burst write transfers are of the PCI command type Memory Write (type 7). AD[1:0] will both be ZERO during the address phase indicating a linear burst order. The byte enable signals indicate the byte lanes that have valid data.

The PCnet-PCI II controller will always perform a single burst write transaction per bus mastership period, where transaction is defined as one address phase and one or multiple data phases. The PCnet-PCI II controller supports zero wait state write cycles except with the case of descriptor write transfers. (See the section “Descriptor DMA Transfers” for the only exception.) It asserts IRDY immediately after the address phase and at the same time starts sampling DEVSEL. FRAME is deasserted when the next to the last data phase is completed.

The following figure shows a typical burst write access. The PCnet-PCI II controller arbitrates for the bus, is granted access, and writes four 32-bit words (DWords) to the system memory and then releases the bus. In this example, the memory system extends the data phase of the first access by one wait state. The following three data phases take one clock cycle each, which is determined by the timing of $\overline{\text{TRDY}}$. The example assumes that EXTREQ (BCR18, bit 8) is set to ONE, therefore, $\overline{\text{REQ}}$ is not deasserted until the next to last data phase is finished.



19436C-19

Figure 16. Burst Write Transfer (EXTREQ = 1)

Target Initiated Termination

When the PCnet-PCI II controller is a bus master, the cycles it produces on the PCI bus may be terminated by the target in one of three different ways.

Disconnect With Data Transfer

The figure below shows a disconnection in which one last data transfer occurs after the target asserted STOP. STOP is asserted on clock 4 to start the termination sequence. Data is still transferred during this cycle, since both IRDY and TRDY are asserted. The PCnet-PCI II controller terminates the current transfer with the deassertion of FRAME on clock 5 and of IRDY one clock later. It finally releases the bus on clock 6. The PCnet-PCI II controller will re-request the bus after 2 clock cycles, if it wants to transfer more data. The starting address of the new transfer will be the address of the next untransferred data.

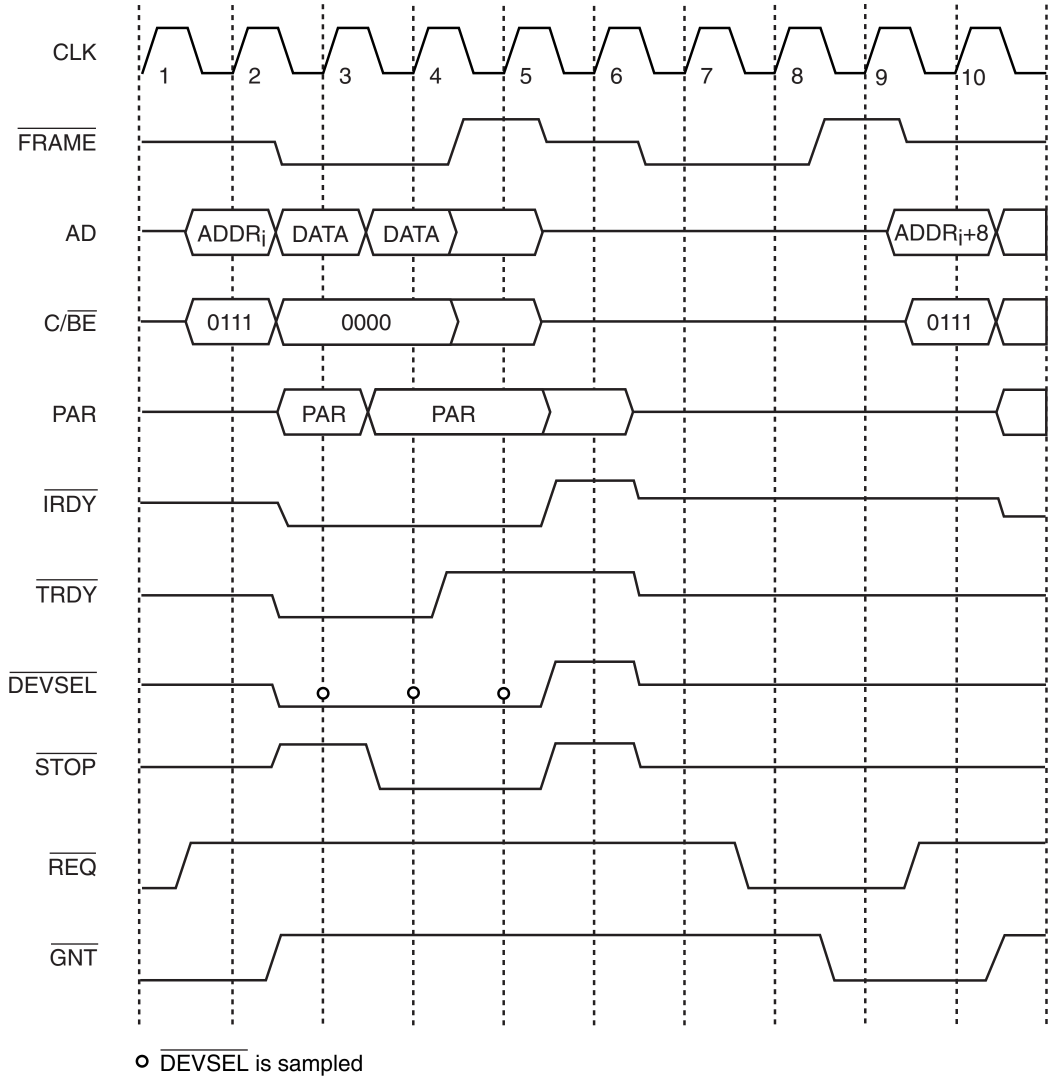


Figure 17. Disconnect With Data Transfer

Disconnect Without Data Transfer

The figure below shows a target disconnect sequence during which no data is transferred. STOP is asserted on clock 4 without TRDY being asserted at the same time. The PCnet-PCI II controller terminates the access with the deassertion of FRAME on clock 5 and of IRDY one clock cycle later. It finally releases the bus on clock 6. The PCnet-PCI II controller will re-request the bus after 2 clock cycles to retry the last transfer. The starting address of the new transfer will be the address of the last untransferred data.

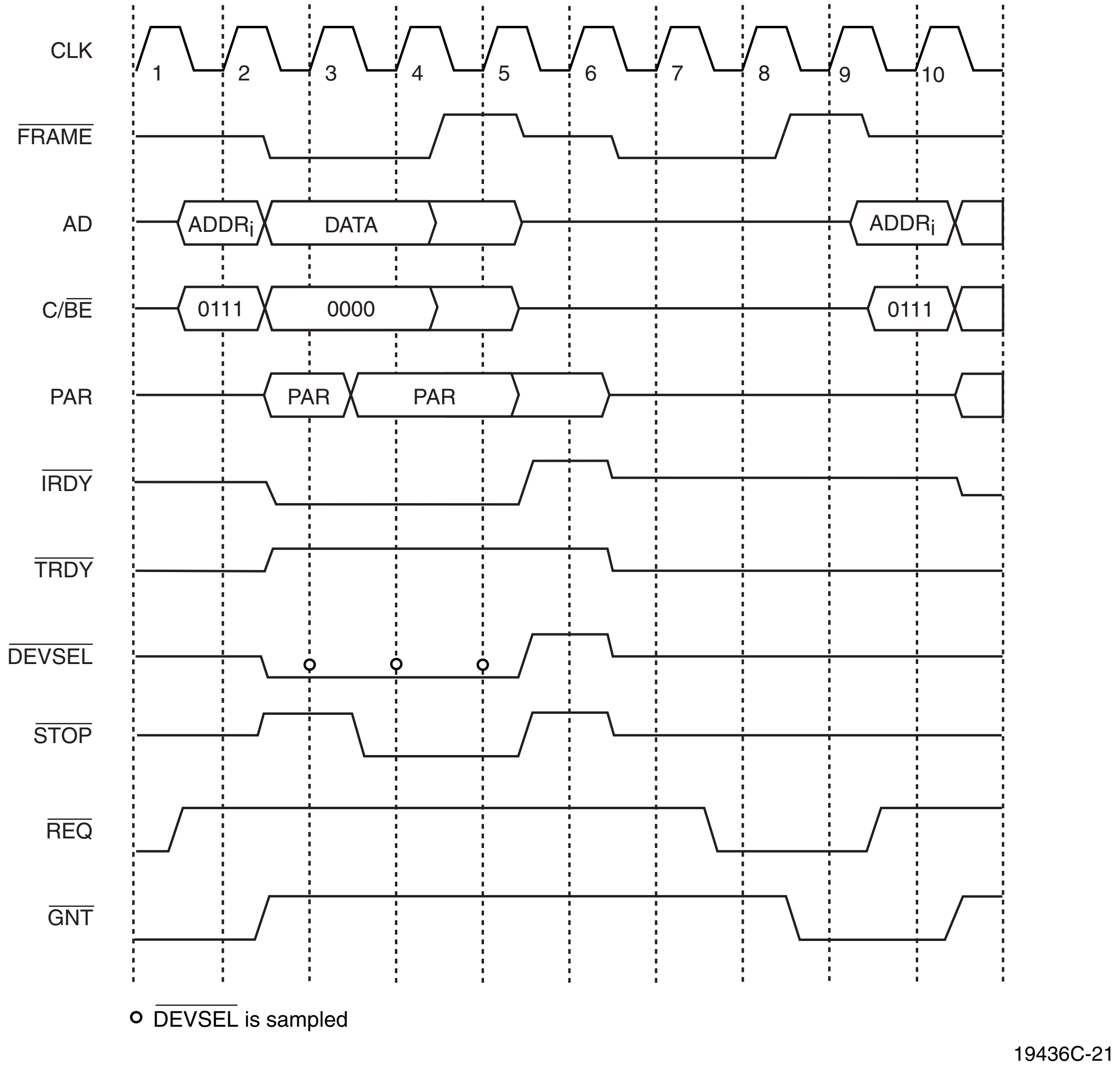


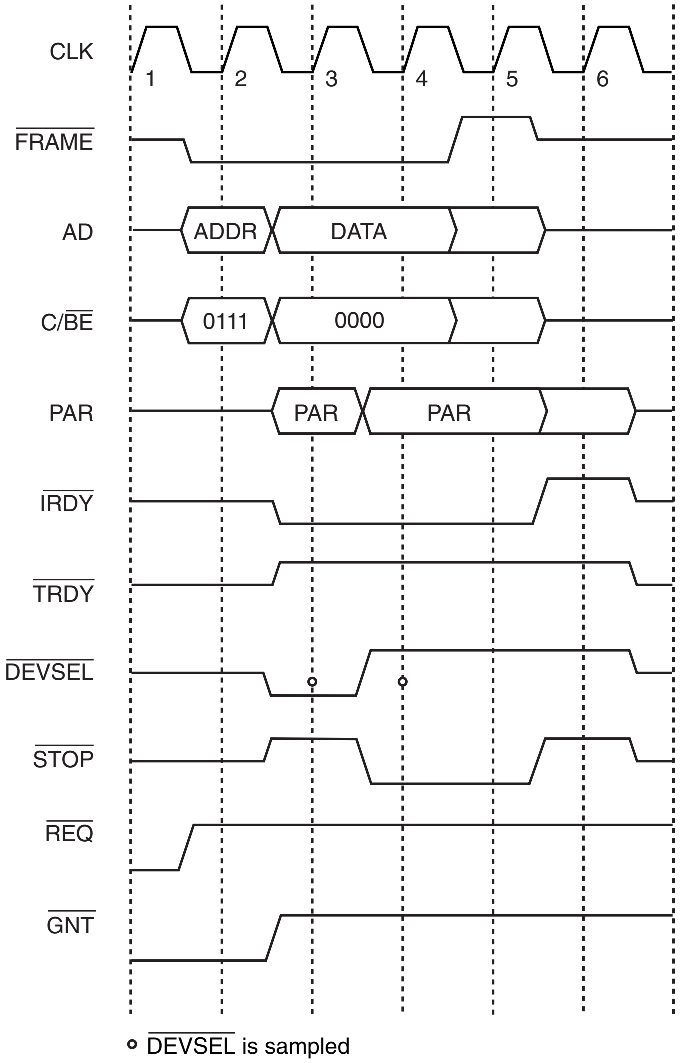
Figure 18. Disconnect Without Data Transfer

Target Abort

The figure below shows a target abort sequence. The target asserts DEVSEL for one clock. It then deasserts DEVSEL and asserts STOP on clock 4. A target can use the target abort sequence to indicate that it cannot service the data transfer and that it does not want the transaction to be retried. Additionally, the PCnet-PCI II controller cannot make any assumption about the success of the previous data transfers in the current transaction. The PCnet-PCI II controller terminates the current transfer with the deassertion of FRAME on clock 5 and of IRDY one clock cycle later. It finally releases the bus on clock 6.

Since data integrity is not guaranteed, the PCnet-PCI II controller cannot recover from a target abort event. The PCnet-PCI II controller will reset all CSR locations to their STOP_RESET values. The BCR and PCI configuration registers will not be cleared. Any on-going network transmission is terminated in an orderly sequence. If less than 512 bits have been transmitted onto the network, the transmission will be terminated immediately, generating a runt packet. If 512 bits or more have been transmitted, the message will have the current FCS inverted and appended at the next byte boundary to guarantee an FCS error is detected at the receiving station.

RTABORT (PCI Status register, bit 12) will be set to indicate that the PCnet-PCI II controller has received a target abort. In addition, SINT (CSR5, bit 11) will be set to ONE. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to ONE. This mechanism can be used to inform the driver of the system error. The host can read the PCI Status register to determine the exact cause of the interrupt.



19436C-22

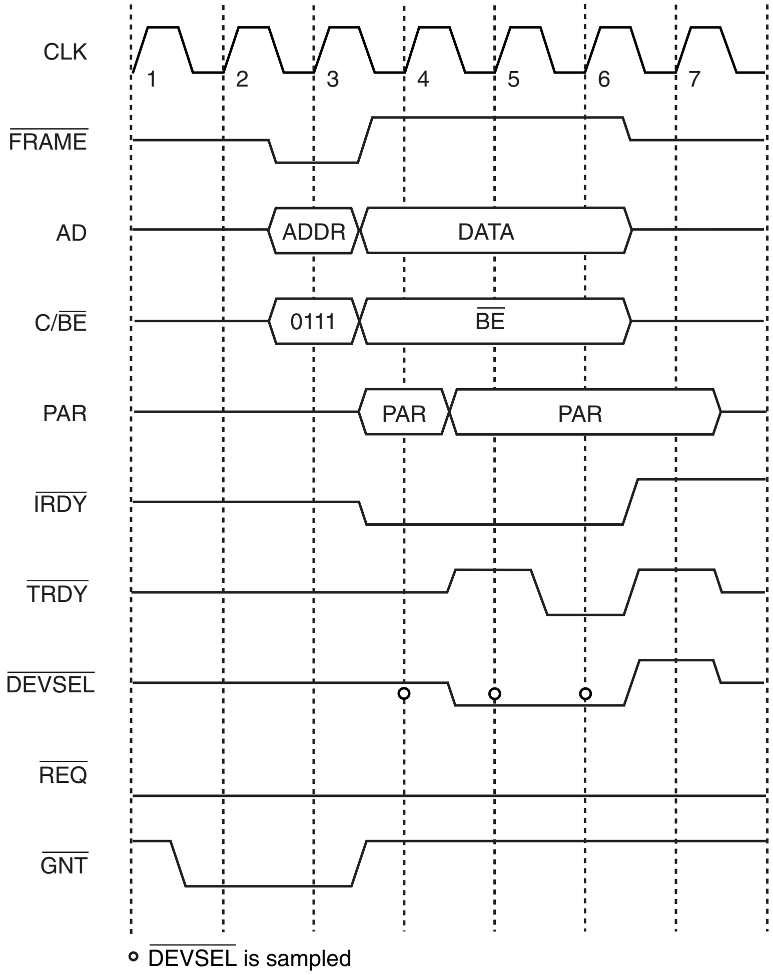
Figure 19. Target Abort

Master Initiated Termination

There are three scenarios besides normal completion of a transaction where the PCnet-PCI II controller will terminate the cycles it produces on the PCI bus.

Preemption During Non-Burst Transaction

When the PCnet-PCI II controller performs multiple non-burst transactions, it keeps $\overline{\text{REQ}}$ asserted until the assertion of $\overline{\text{FRAME}}$ for the last transaction. When $\overline{\text{GNT}}$ is removed, the PCnet-PCI II controller will finish the current transaction and then release the bus. If it is not the last transaction, $\overline{\text{REQ}}$ will remain asserted to regain bus ownership as soon as possible.



19436C-23

Figure 20. Preemption During Non-Burst Transaction

Preemption During Burst Transaction

When the PCnet-PCI II controller operates in burst mode, it only performs a single transaction per bus mastership period, where transaction is defined as one address phase and one or multiple data phases. The central arbiter can remove $\overline{\text{GNT}}$ at any time during the transaction. The PCnet-PCI II controller will ignore the deassertion of $\overline{\text{GNT}}$ and continue with data transfers, as long as the PCI Latency Timer is not expired. When the Latency Timer is ZERO and $\overline{\text{GNT}}$ is deasserted, the PCnet-PCI II controller will finish the current data phase, deassert $\overline{\text{FRAME}}$, finish the last data phase and release the bus. If EXTREQ (BCR18, bit 8) is cleared to ZERO, it will immediately assert $\overline{\text{REQ}}$ to regain bus ownership as soon as possible. If EXTREQ is set to ONE, $\overline{\text{REQ}}$ will stay asserted. When the preemption occurs after the counter has counted down to ZERO, the PCnet-PCI II controller will finish the current data phase, deassert $\overline{\text{FRAME}}$, finish the last data phase and release the bus. Note that it is important for the host to program the PCI Latency Timer according to the bus bandwidth requirement of the PCnet-PCI II controller. The host can determine this bus bandwidth requirement by reading the PCI MAX_LAT and MIN_GNT registers.

The figure below assumes that the PCI Latency Timer has counted down to ZERO on clock 7.

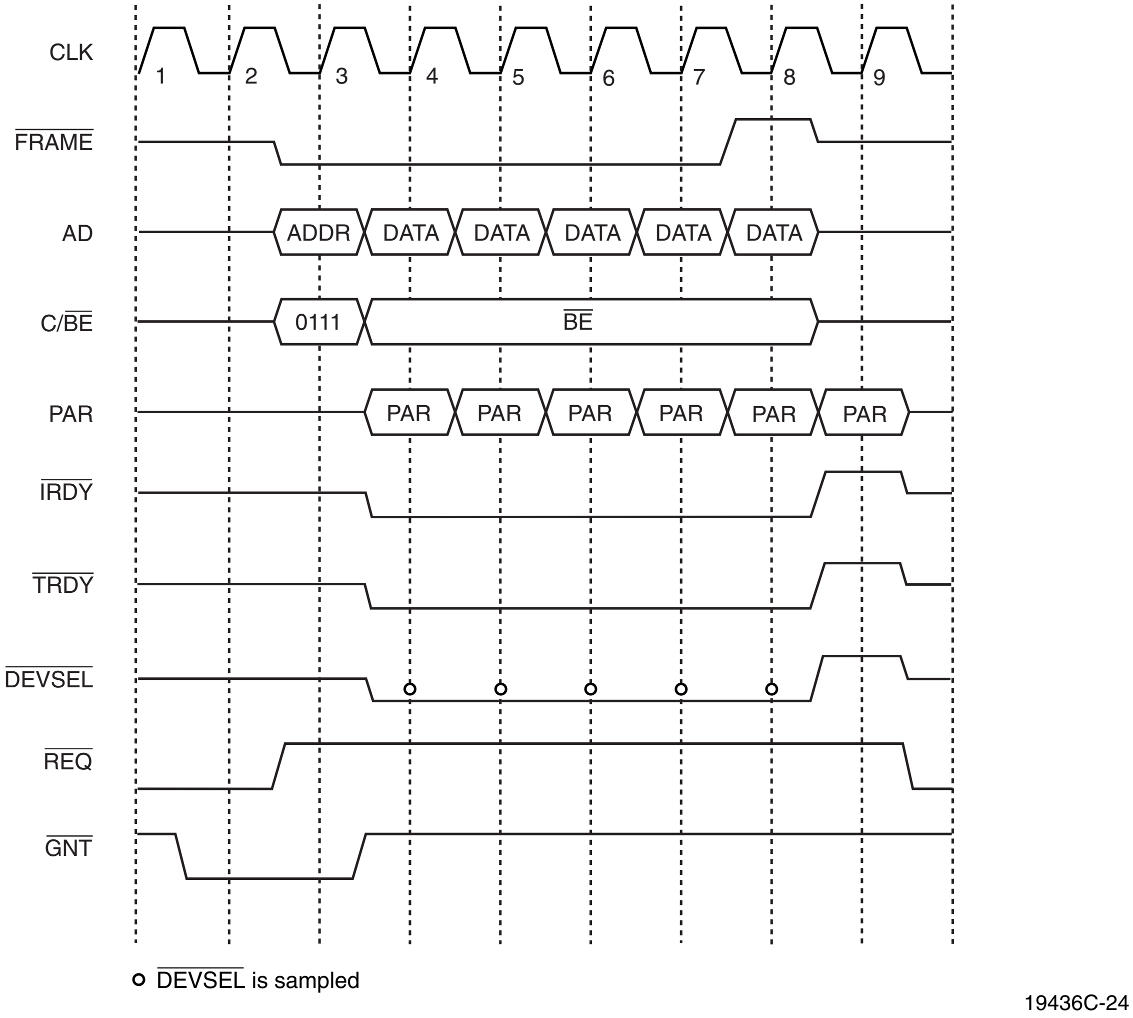


Figure 21. Preemption During Burst Transaction

Master Abort

The PCnet-PCI II controller will terminate its cycle with a Master Abort sequence if DEVSEL is not asserted within 4 clocks after FRAME is asserted. Master Abort is treated as a fatal error by the PCnet-PCI II controller. The PCnet-PCI II controller will reset all CSR locations to their STOP_RESET values. The BCR and PCI configuration registers will not be cleared. Any on-going network transmission is terminated in an orderly sequence. If less than 512 bits have been transmitted onto the network, the transmission will be terminated immediately, generating a runt packet. If 512 bits or more have been transmitted, the message will have the current FCS inverted and appended at the next byte boundary to guarantee an FCS error is detected at the receiving station.

RMABORT (in the PCI Status register, bit 13) will be set to indicate that the PCnet-PCI II controller has terminated its transaction with a master abort. In addition, SINT (CSR5, bit 11) will be set to ONE. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to ONE. This mechanism can be used to inform the driver of the system error. The host can read the PCI Status register to determine the exact cause of the interrupt.

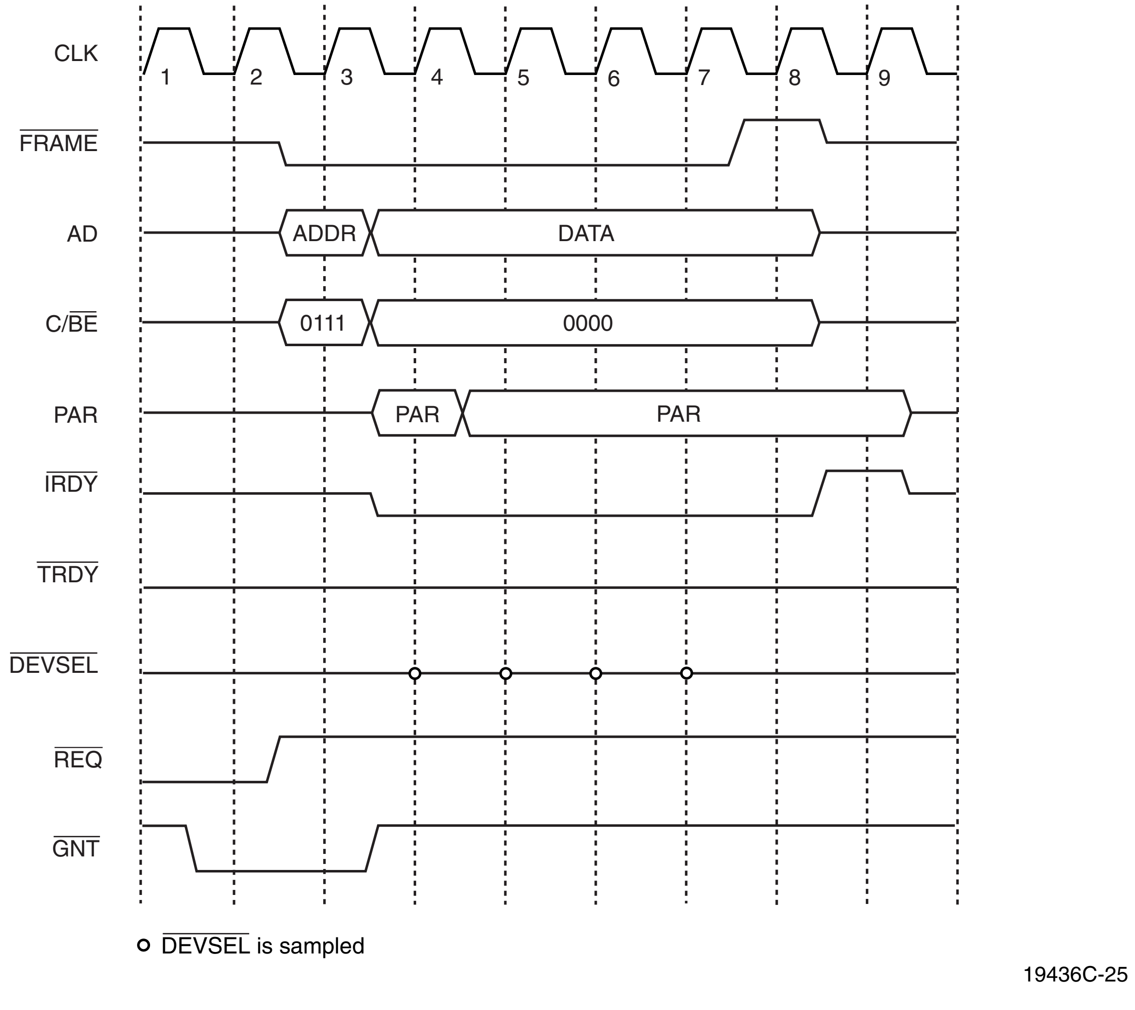


Figure 22. Master Abort

Parity Error Response

During every data phase of a DMA read operation, when the target indicates that the data is valid by asserting TRDY, the PCnet-PCI II controller samples the AD[31:0], C/BE[3:0] and the PAR lines for a data parity error. When it detects a data parity error, the controller sets PERR (PCI Status register, bit 15) to ONE. When reporting of that error is enabled by setting PERREN (PCI Command register, bit 6) to ONE, the PCnet-PCI II controller also drives the PERR signal low and sets DATAPERR (PCI Status register, bit 8) to ONE. The assertion of PERR follows the corrupted data/byte enables by two clock cycles and PAR by one clock cycle.

The figure below shows a transaction that has a parity error in the data phase. The PCnet-PCI II controller asserts PERR on clock 8, two clock cycles after data is valid. The data on clock 5 is not checked for parity, since on a read access PAR is only required to be valid one clock after the target has asserted TRDY. The PCnet-PCI II controller then drives PERR high for one clock cycle, since PERR is a sustained tri-state signal.

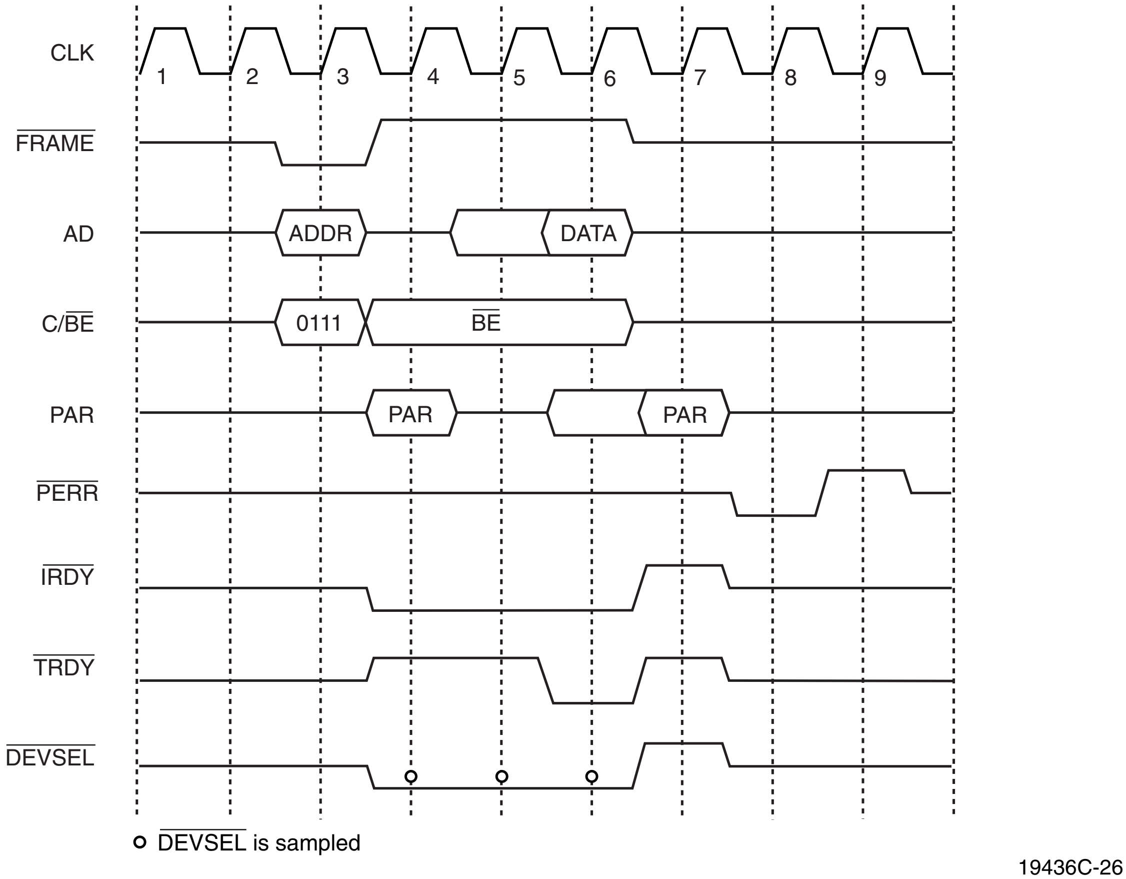


Figure 23. Master Cycle Data Parity Error Response

During every data phase of a DMA write operation, the PCnet-PCI II controller checks the PERR input to see if the target reports a parity error. When it sees the PERR input asserted, the controller sets PERR (PCI Status register, bit 15) to ONE. When PERREN (PCI Command register, bit 6) is set to ONE, the PCnet-PCI II controller also sets DATAPERR (PCI Status register, bit 8) to ONE.

Whenever the PCnet-PCI II controller is the current bus master and a data parity error occurs, SINT (CSR5, bit 11) will be set to ONE. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to ONE. This mechanism can be used to inform the driver of the system error. The host can read the PCI Status register to determine the exact cause of the interrupt. The setting of SINT due to a data parity error is not dependent on the setting of PERREN (PCI Command register, bit 6).

By default, a data parity error does not affect the state of the MAC engine. The PCnet-PCI II controller treats the data in all bus master transfers that have a parity error as if nothing has happened. All network activity continues.

Advanced Parity Error Handling

For all DMA cycles, the PCnet-PCI II controller provides a second, more advanced level of parity error handling. This mode is enabled by setting APERREN (BCR20, bit 10) to ONE.

When APERREN is set to ONE, the BPE bits (RMD1 and TMD1, bit 23) are used to indicate parity error in data transfers to the receive and transmit buffers. Note that since the advanced parity error handling uses an additional bit in the descriptor, SWSTYLE (BCR20, bits 7–0) must be set to ONE, TWO or THREE to program the PCnet-PCI II controller to use 32-bit software structures. The PCnet-PCI II controller will react in the following way when a data parity error occurs:

- Initialization block read: STOP (CSR0, bit 2) is set to ONE and causes a STOP_RESET of the device.
- Descriptor ring read: Any on-going network activity is terminated in an orderly sequence and then STOP (CSR0, bit 2) is set to ONE to cause a STOP_RESET of the device.
- Descriptor ring write: Any on-going network activity is terminated in an orderly sequence and then STOP (CSR0, bit 2) is set to ONE to cause a STOP_RESET of the device.
- Transmit buffer read: BPE (TMD1, bit 23) is set in the current transmit descriptor. Any on-going network transmission is terminated in an orderly sequence.
- Receive buffer write: BPE (RMD1, bit 23) is set in the last receive descriptor associated with the frame.

Terminating on-going network transmission in an orderly sequence means that if less than 512 bits have been transmitted onto the network, the transmission will be terminated immediately, generating a runt packet. If 512 bits or more have been transmitted, the message will have the current FCS inverted and appended at the next byte boundary to guarantee an FCS error is detected at the receiving station.

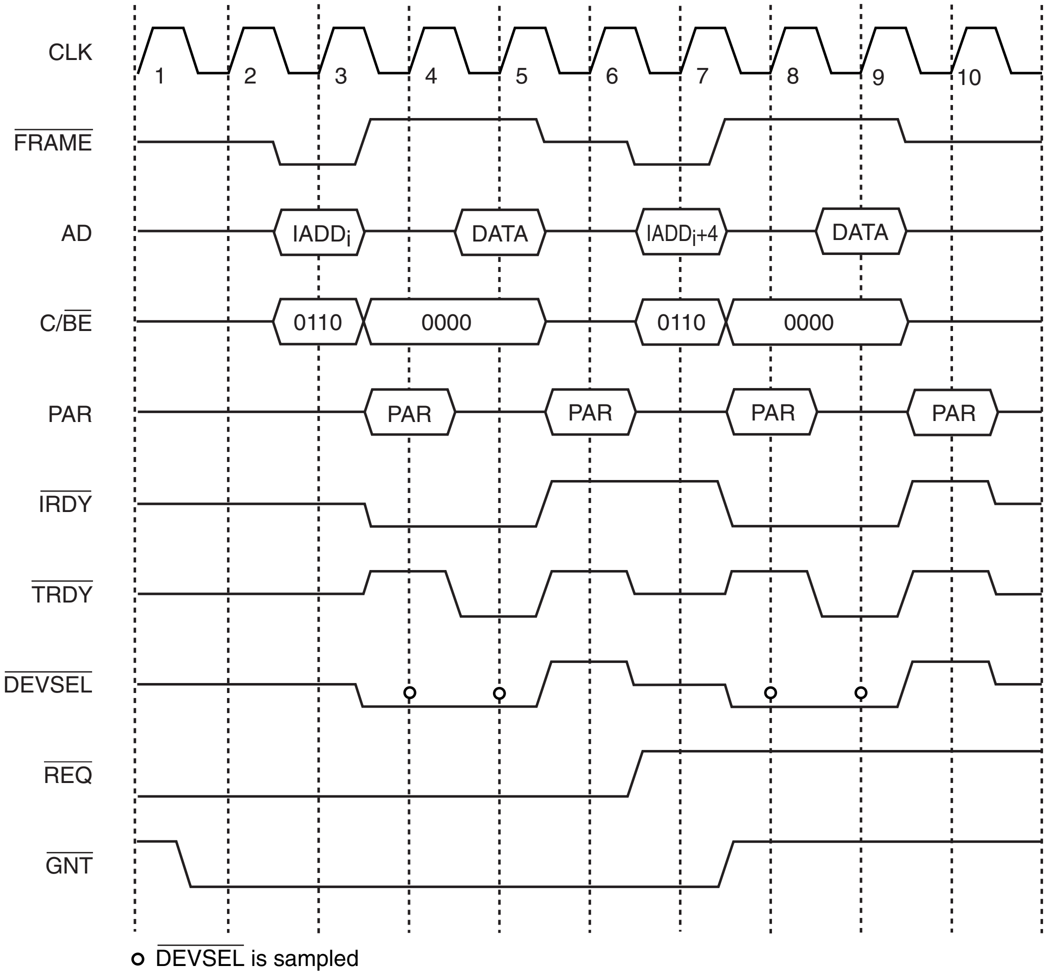
APERREN does not affect the reporting of address parity errors or data parity errors that occur when the PCnet-PCI II controller is the target of the transfer.

Initialization Block DMA Transfers

During execution of the PCnet-PCI II controller bus master initialization procedure, the PCnet-PCI II controller microcode will repeatedly request DMA transfers from the BIU. During each of these initialization block DMA transfers, the BIU will perform two data transfer cycles reading one DWord per transfer and then it will relinquish the bus. When SSIZE32 (BCR20, bit 8) is set to ONE (i.e. the initialization block is organized as 32-bit software structures), there are 7 DWords to transfer during the bus master initialization procedure, so four bus mastership periods are needed in order to complete the initialization sequence. Note that the last DWord transfer of the last bus mastership period of the initialization sequence accesses an unneeded location. Data from this transfer is discarded internally. When SSIZE32 is cleared to ZERO (i.e. the initialization block is organized as 16-bit software structures), then three bus mastership periods are needed to complete the initialization sequence.

The PCnet-PCI II controller supports two transfer modes for reading the initialization block: non-burst and burst mode; with burst mode being the preferred mode when the PCnet-PCI II controller is used in a PCI bus application.

When BREADE is cleared to ZERO (BCR18, bit 6), all initialization block read transfers will be executed in non-burst mode. There is a new address phase for every data phase. FRAME will be dropped between the two transfers. The two phases within a bus mastership period will have addresses of ascending contiguous order.



19436C-27

Figure 24. Initialization Block Read In Non-Burst Mode

When BREADE is set to ONE (BCR18, bit 6), all initialization block read transfers will be executed in burst mode. AD[1:0] will be ZERO during the address phase indicating a linear burst order.

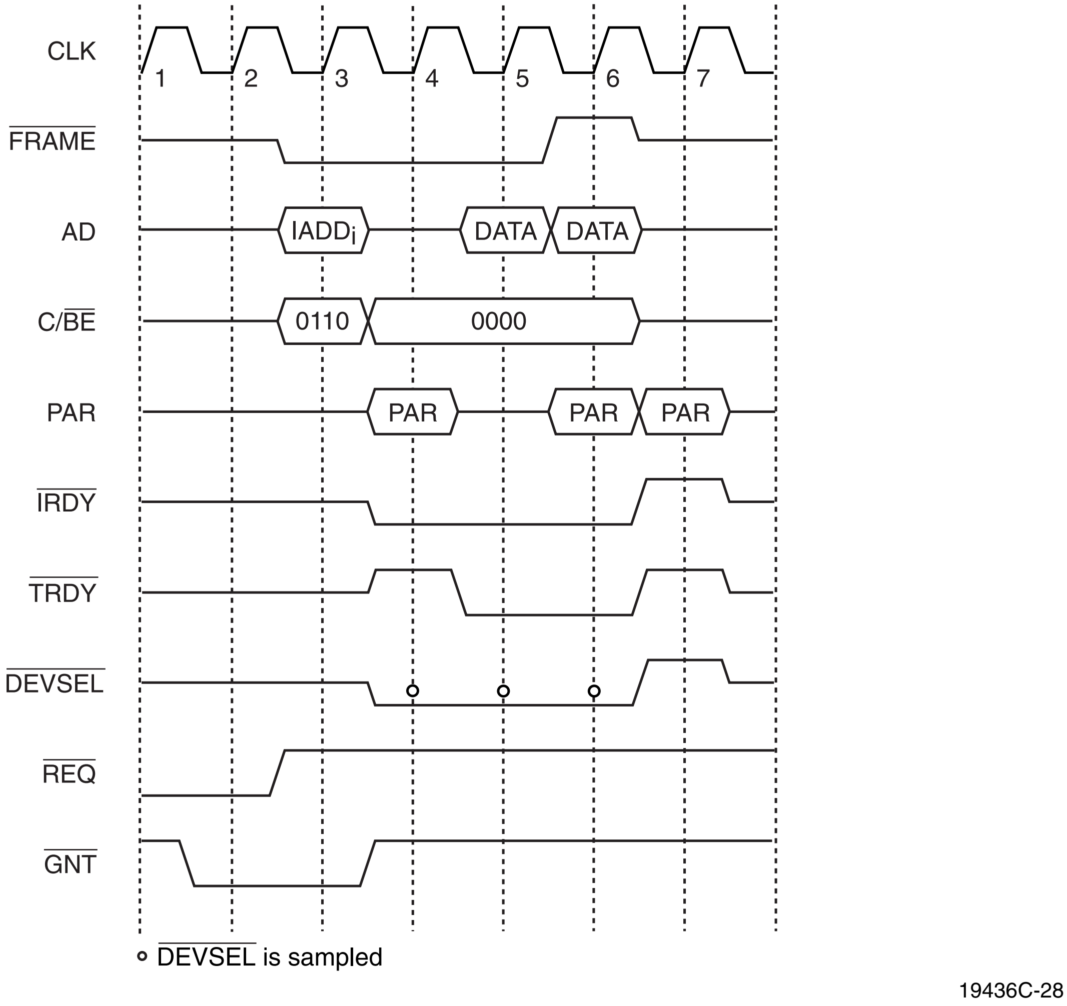


Figure 25. Initialization Block Read In Burst Mode

Descriptor DMA Transfers

PCnet-PCI II controller microcode will determine when a descriptor access is required. A descriptor DMA read will consist of two data transfers. A descriptor DMA write will consist of one or two data transfers. The descriptor DMA transfers within a single bus mastership period will always be of the same type (either all read or all write).

During descriptor read accesses, the byte enable signals will indicate that all byte lanes are active. Should some of the bytes not be needed, then the PCnet-PCI II controller will internally discard the extraneous information that was gathered during such a read.

The settings of SWSTYLE (BCR20, bits 7–0) and BREADE (BCR18, bit 6) affect the way the PCnet-PCI II controller performs descriptor read operations.

When SWSTYLE is set to ZERO, ONE or TWO, all descriptor read operations are performed in non-burst mode. The setting of BREADE has no effect in this configuration.

When SWSTYLE is set to THREE, the descriptor entries are ordered to allow burst transfers. The PCnet-PCI II controller will perform all descriptor read operations in burst mode, if BREADE is set to ONE.

Table 4. Descriptor Read Sequence

SWSTYLE BCR18[6]	BREADE BCR20[7:0]	AD Bus Sequence

0	X	Address = XXXX XX00h
		Turn around cycle
		Data = MD1[31:24], MD0[23:0]
		Idle
		Address = XXXX XX04h
		Turn around cycle
		Data = MD2[15:0], MD1[15:0]

1,2	X	Address = XXXX XX04h
		Turn around cycle
		Data = MD1[31:0]
		Idle
		Address = XXXX XX00h
		Turn around cycle
		Data = MD0[31:0]

3	0	Address = XXXX XX04h
		Turn around cycle
		Data = MD1[31:0]
		Idle
		Address = XXXX XX08h
		Turn around cycle
		Data = MD0[31:0]

3	1	Address = XXXX XX04h
		Turn around cycle
		Data = MD1[31:0]
		Data = MD0[31:0]

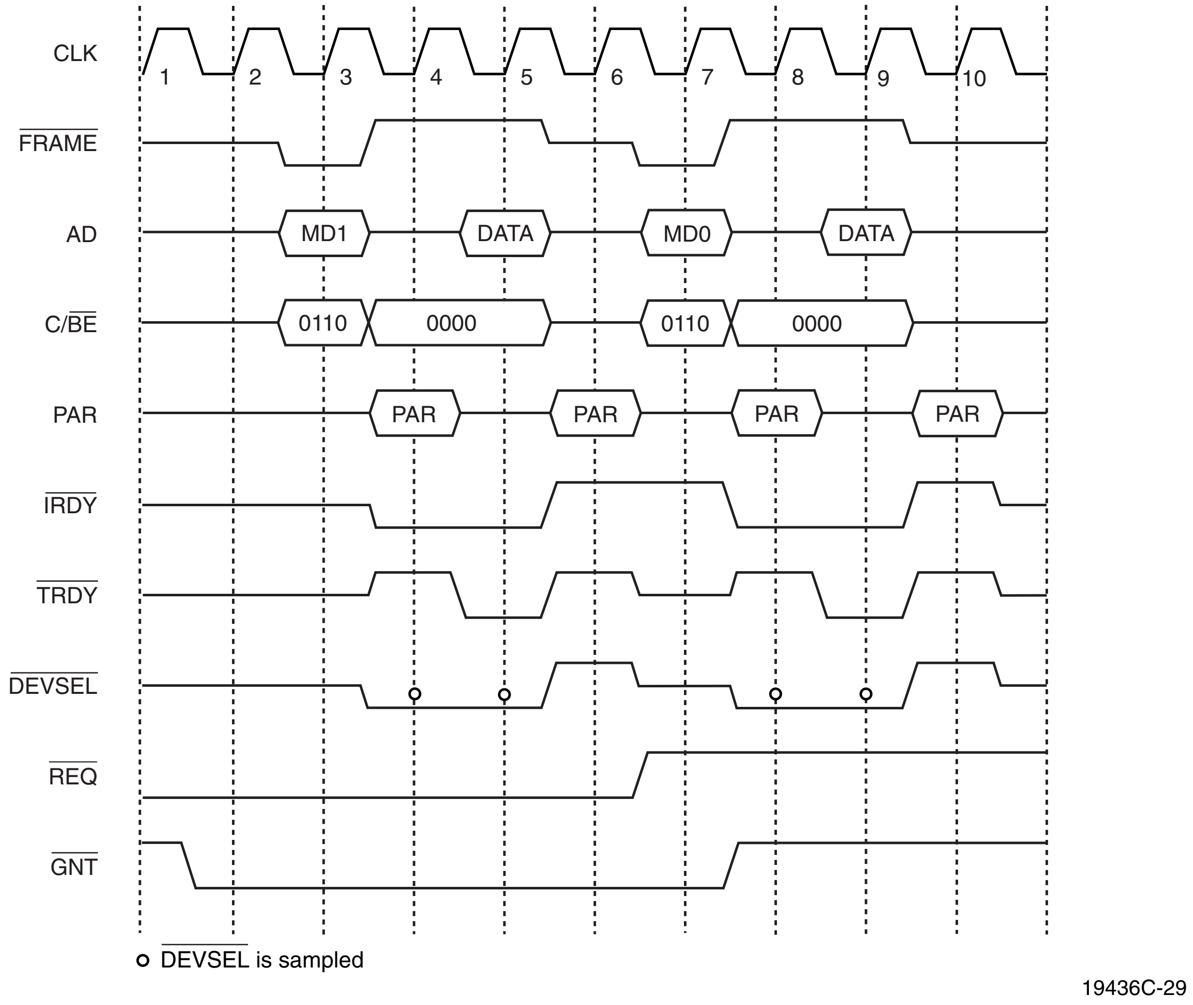


Figure 26. Descriptor Ring Read In Non-Burst Mode

During descriptor write accesses, only the byte lanes which need to be written are enabled.

If buffer chaining is used, accesses to the descriptors of all intermediate buffers consist of only one data transfer to return ownership of the buffer to the system. When SWSTYLE (BCR20, bits 7–0) is cleared to ZERO (i.e. the descriptor entries are organized as 16-bit software structures), the descriptor access will write a single byte. When SWSTYLE (BCR20, bits 7–0) is set to ONE, TWO or THREE (i.e. the descriptor entries are organized as 32-bit software structures), the descriptor access will write a single word. On all single buffer transmit or receive descriptors, as well as on the last buffer in chain, writes to the descriptor consist of two data transfers. The first one writing a DWord containing status information. The second data transfer writing a byte (SWSTYLE cleared to ZERO) or otherwise a word containing additional status and the ownership bit (i.e. MD1[31]).

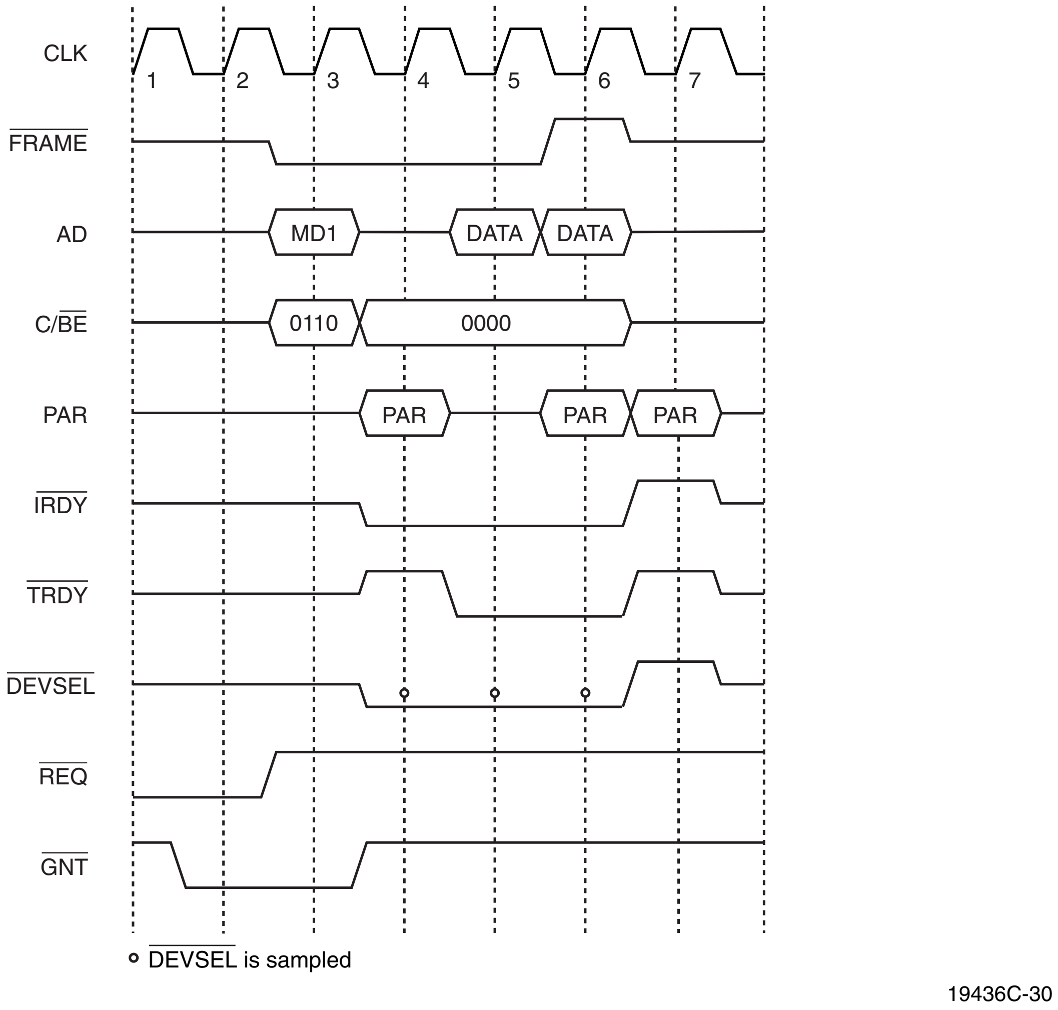


Figure 27. Descriptor Ring Read In Burst Mode

The settings of SWSTYLE (BCR20, bits 7–0) and BWRITE (BCR18, bit 5) affect the way the PCnet-PCI II controller performs descriptor write operations.

When SWSTYLE is set to ZERO, ONE or TWO, all descriptor write operations are performed in non-burst mode. The setting of BWRITE has no effect in this configuration.

When SWSTYLE is set to THREE, the descriptor entries are ordered to allow burst transfers. The PCnet-PCI II controller will perform all descriptor write operations in burst mode, if BWRITE is set to ONE.

A write transaction to the descriptor ring entries is the only case where the PCnet-PCI II controller inserts a wait state when being the bus master. Every data phase in non-burst and burst mode is extended by one clock cycle, during which IRDY is deasserted.

Table 5. Descriptor Write Sequence

SWSTYLE BCR20[7:0]	BWRITE BCR18[5]	AD Bus Sequence

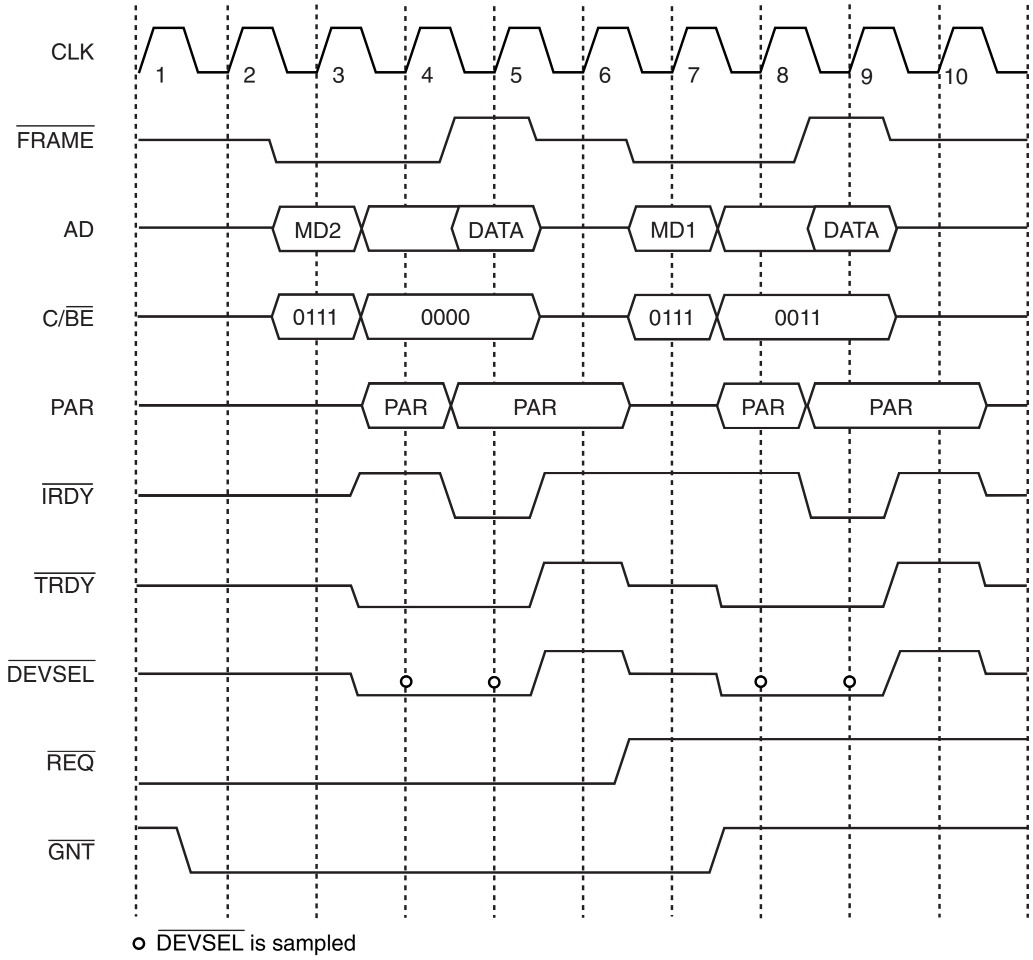
0	X	Address = XXXX XX04h
		Data = MD2[15:0], MD1[15:0]
		Idle
		Address = XXXX XX00h
		Data = MD1[31:24]

1,2	X	Address = XXXX XX08h
		Data = MD2[31:0]
		Idle
		Address = XXXX XX04h
		Data = MD1[31:16]

3	0	Address = XXXX XX00h
		Data = MD2[31:0]
		Idle
		Address = XXXX XX04h
		Data = MD1[31:16]

3	1	Address = XXXX XX00h
		Data = MD2[31:0]
		Data = MD1[31:16]

Note that the figure below assumes that the PCnet-PCI II controller is programmed to use 32-bit software structures (SWSTYLE = 1, 2, or 3). The byte enable signals for the second data transfer would be 0111b, if the device was programmed to use 16-bit software structures (SWSTYLE = 0).



19436C-31

Figure 28. Descriptor Ring Write In Non-Burst Mode

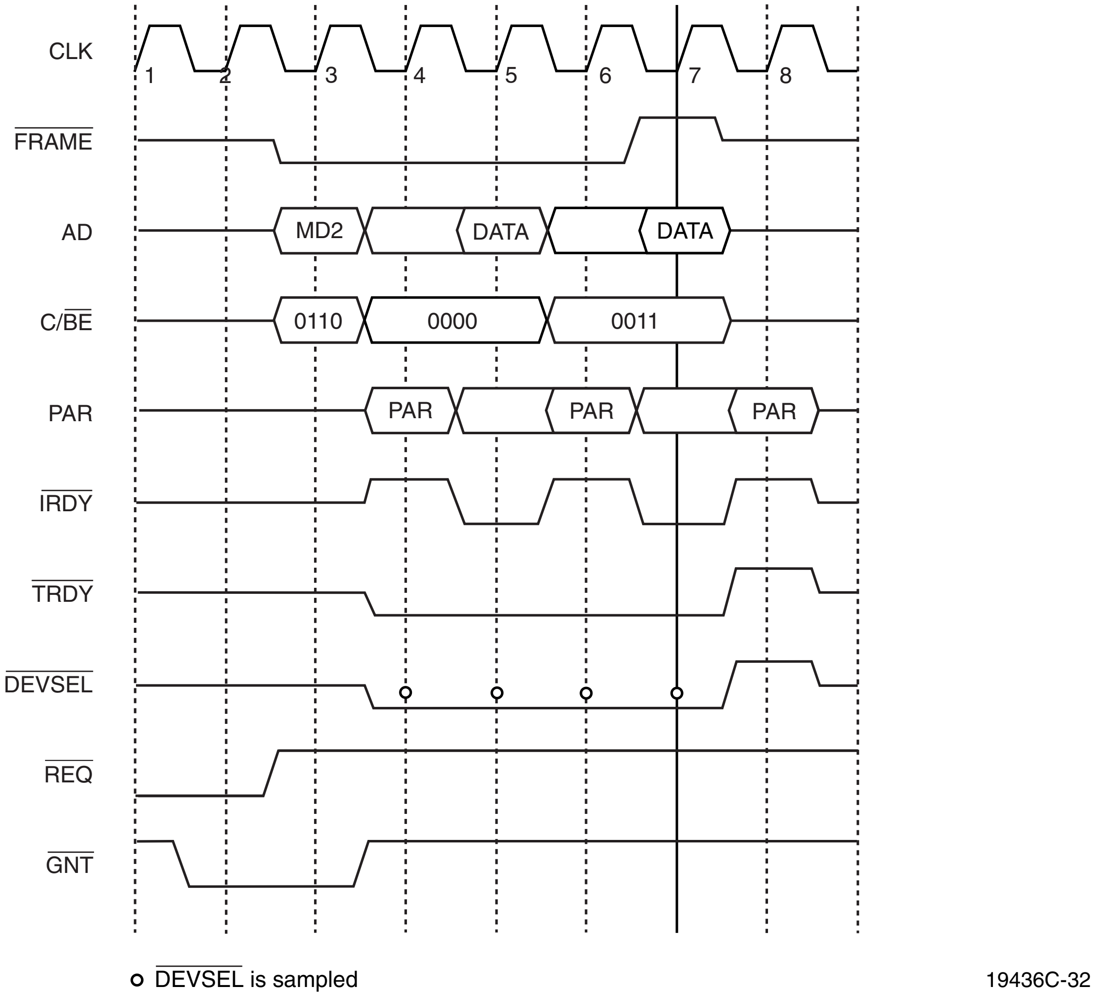


Figure 29. Descriptor Ring Write In Burst Mode

FIFO DMA Transfers

PCnet-PCI II controller microcode will determine when a FIFO DMA transfer is required. This transfer mode will be used for transfers of data to and from the PCnet-PCI II controller FIFOs. Once the PCnet-PCI II controller BIU has been granted bus mastership, it will perform a series of consecutive transfer cycles before relinquishing the bus. All transfers within the master cycle will be either read or write cycles, and all transfers will be to contiguous, ascending addresses. Both non-burst and burst cycles are used, with burst mode being the preferred mode when the device is used in a PCI bus application.

Non-Burst FIFO DMA Transfers

In the default mode the PCnet-PCI II controller uses non-burst transfers to read and write data when accessing the FIFOs. Each non-burst transfer will be performed sequentially, with the issue of an address, and the transfer of the corresponding data with appropriate output signals to indicate selection of the active data bytes during the transfer. FRAME will be deasserted after every address phase. The number of data transfer cycles contained within a single bus mastership period is in general dependent on the programming of the DMAPLUS option (CSR4, bit 14). Several other factors will also affect the length of the bus mastership period. The possibilities are as follows:

If DMAPLUS is cleared to ZERO, a maximum of 16 transfers will be performed by default. This default value may be changed by writing to the DMA Transfer Counter (CSR80). Note that DMAPLUS = 0 merely sets a maximum value. The minimum number of transfers in the bus mastership period will be determined by all of the following variables: the settings of the FIFO watermarks (CSR80), the conditions of the FIFOs, the value of the DMA Transfer Counter (CSR80), the value of the DMA Bus Timer (CSR82), and any occurrence of preemption that takes place during the bus mastership period.

If DMAPLUS is set to ONE, bus cycles 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), or until the DMA Bus Activity Timer (CSR82) has expired. The exact number of total transfer cycles in the bus mastership period is dependent on all of the following variables: the settings of the FIFO watermarks, the conditions of the FIFOs, the latency of the system bus to the PCnet-PCI II controller's bus request, the speed of bus operation and bus preemption events. The DMA Transfer Counter is disabled when DMAPLUS is set to ONE. The $\overline{\text{TRDY}}$ response time of the memory device will also affect the number of transfers, 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. For example, on a receive operation, a slower $\overline{\text{TRDY}}$ response will allow additional data to accumulate inside of the FIFO. If the accesses are slow enough, a complete DWord may become available before the end of the bus mastership period and thereby increase the number of transfers in that period. The general rule is that the longer the Bus Grant latency, the slower the bus transfer operations, the slower the clock speed, the higher the transmit watermark or the lower the receive watermark, the longer the bus mastership period will be.

Note that the PCI Latency Timer is not significant during non-burst transfers.

Burst FIFO DMA Transfers

Bursting is only performed by the PCnet-PCI II controller if the BREADE and/or BWRITE bits of BCR18 are set. These bits individually enable/disable the ability of the PCnet-PCI II controller to perform burst accesses during master read operations and master write operations, respectively.

A burst transaction will start with an address phase, followed by one or more data phases. AD[1:0] will always be ZERO during the address phase indicating a linear burst order.

During FIFO DMA read operations, all byte lanes will always be active. The PCnet-PCI II controller will internally discard unused bytes. During the first and the last data phases of a FIFO DMA burst write operation, one or more of the byte enable signals may be inactive. All other data phases will always write a complete DWord.

The following figure shows the beginning of a FIFO DMA write with the beginning of the buffer not aligned to a DWord boundary. The PCnet-PCI II controller starts off by writing only three bytes during the first data phase. This operation aligns the address for all other data transfers to a 32-bit boundary so that the PCnet-PCI II controller can continue bursting full DWords.

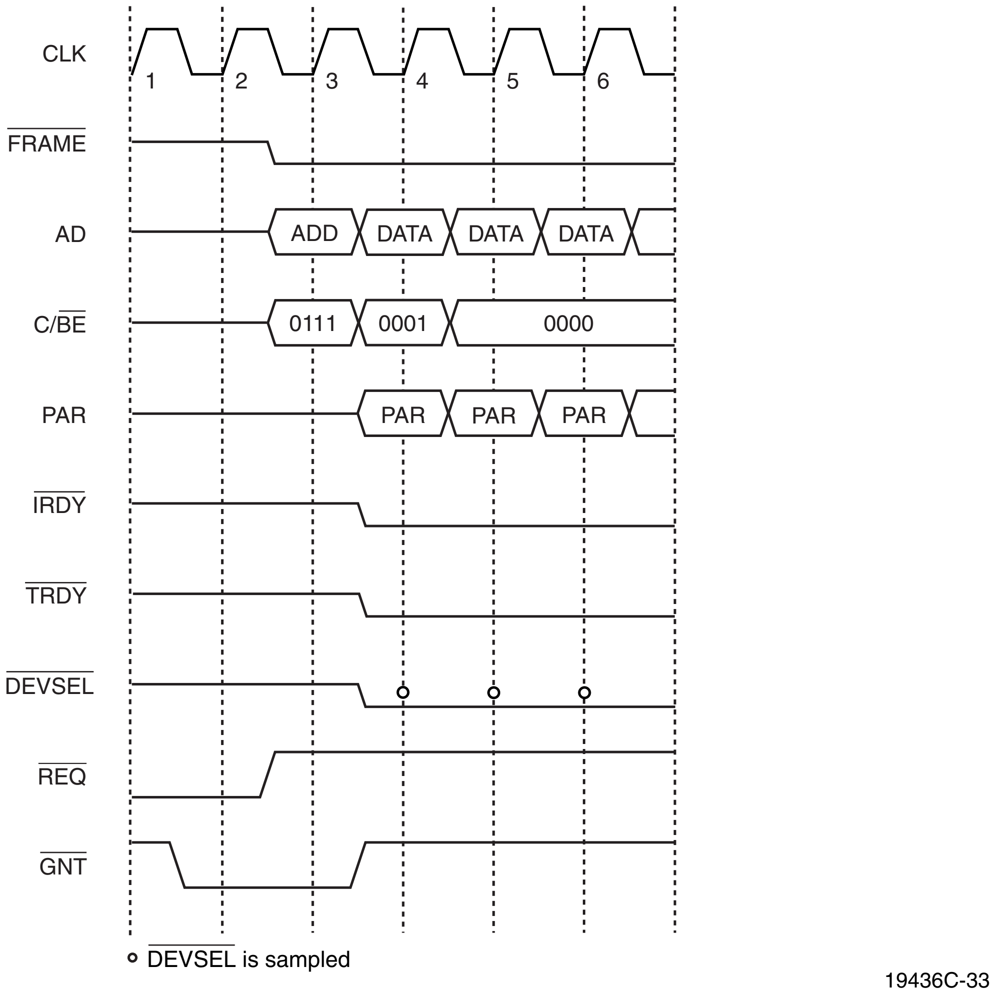


Figure 30. FIFO Burst Write At Start Of Unaligned Buffer

If a receive buffer does not end on a DWord boundary, the PCnet-PCI II controller will perform a non-DWord write on the last transfer to the buffer. The following figure shows the final three FIFO DMA transfers to a receive buffer. Since there were only nine bytes of space left in the receive buffer, the PCnet-PCI II controller burst three data phases. The first two data phases write a full DWord, the last one only writes a single byte.

Note that the PCnet-PCI II controller will always perform a DWord transfer as long as it owns the buffer space, even when there are less then four bytes to write. For example, if there is only one byte left for the current receive frame, the PCnet-PCI II controller will write a full DWord, containing the last byte of the receive frame in the least significant byte position (BSWP is cleared to ZERO, CSR3, bit 2). The content of the other three bytes is undefined. The message byte count in the receive descriptor always reflects the exact length of the received frame.

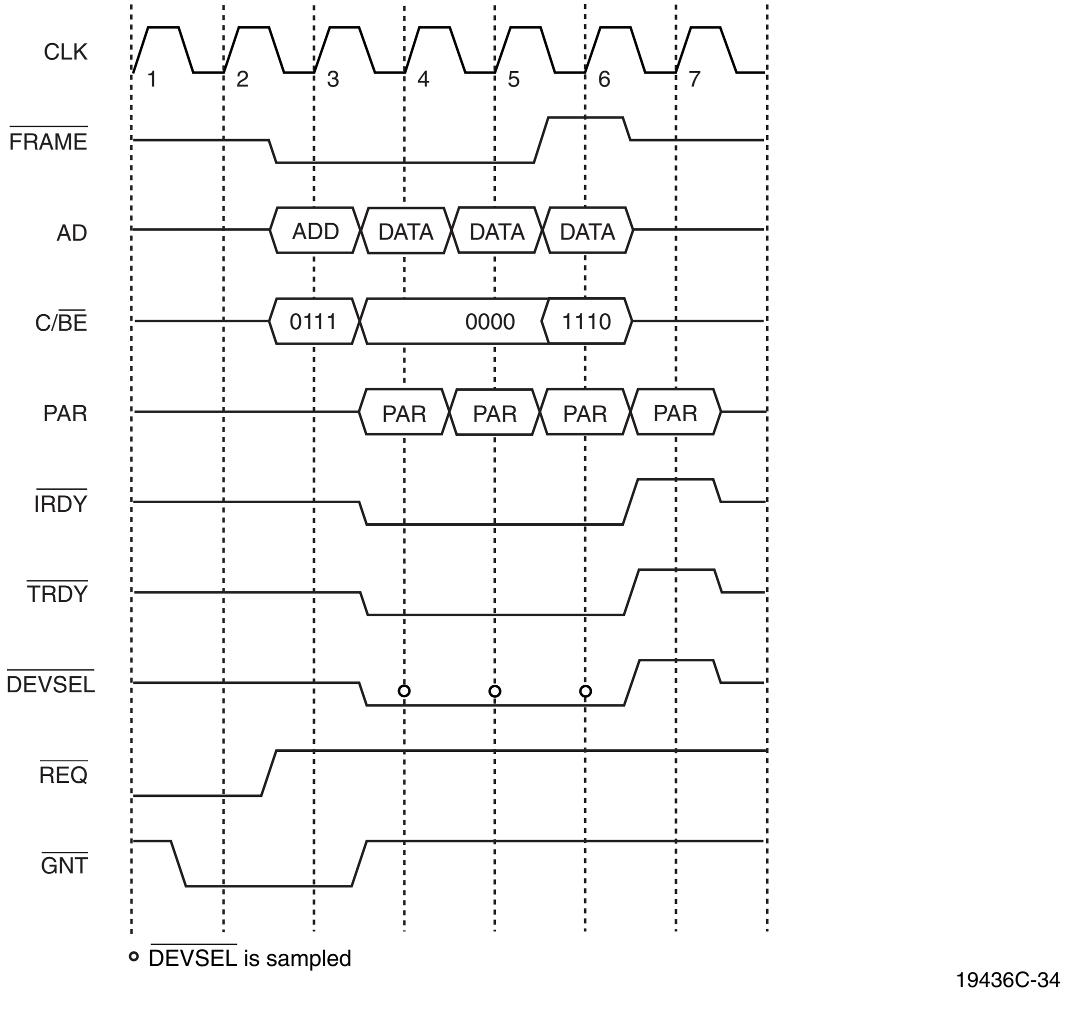


Figure 31. FIFO Burst Write At End Of Unaligned Buffer

In a PCI bus application the PCnet-PCI II controller should be set up to have the length of a bus mastership period be controlled only by the PCI Latency Timer. The Timer bit (CSR4, bit 13) should remain at its default value of ZERO so that the DMA Bus Activity Timer (CSR82) is not enabled. The DMA Transfer Counter (CSR80) should be disabled by setting DMAPLUS (CSR4, bit 14) to ONE. In this mode, the PCnet-PCI II controller will continue transferring FIFO data until the transmit FIFO is filled to its high threshold (read transfers) or the receive FIFO is emptied to its low threshold (write transfers), or the PCnet-PCI II controller is preempted, and the PCI Latency Timer is expired. The host should use the values in the PCI MIN_GNT and MAX_LAT registers to determine the value for the PCI Latency Timer.

In applications that don't use the PCI Latency Timer or that don't support preemption the following rules apply to limit the time the PCnet-PCI II controller takes up on the bus.

If DMAPLUS is cleared to ZERO, a maximum of 16 transfers will be performed by default. This default value may be changed by writing to the DMA Transfer Counter (CSR80). Note that DMAPLUS = 0 merely sets a maximum value. The minimum number of transfers in the bus mastership period will be determined by all of the following variables: the settings of the FIFO watermarks (CSR80), the conditions of the FIFOs, the value of the DMA Transfer Counter (CSR80) and the value of the DMA Bus Activity Timer (CSR82).

If DMAPLUS is set to ONE, bursting 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), or until the DMA Bus Activity Timer (CSR82) has expired. The exact number of total transfer cycles in the bus mastership period is dependent on all of the following variables: the settings of the FIFO watermarks, the conditions of the FIFOs, the latency of the system bus to the PCnet-PCI II controller's bus request, and the speed of bus operation. The DMA Transfer Counter is disabled when DMAPLUS is set to ONE. The $\overline{\text{TRDY}}$ response time of the memory device will also affect the number of transfers, 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. For example, on a receive operation, a slower $\overline{\text{TRDY}}$ response will allow additional data to accumulate inside of the FIFO. If the accesses are slow enough, a complete DWord may become available before the end of the bus mastership period and thereby increase the number of transfers in that period. The general rule is that the longer the Bus Grant latency, the slower the bus transfer operations, the slower the clock speed, the higher the transmit watermark or the lower the receive watermark, the longer the total burst length will be.

When a FIFO DMA burst operation is preempted, the PCnet-PCI II controller will not relinquish bus ownership until the PCI Latency Timer expires. The DMA Transfer Counter will freeze at the current value while the PCnet-PCI II controller is waiting to regain bus ownership. It will continue counting when the FIFO DMA burst operation restarts. The Bus Activity Timer will be reset to its starting value when the PCnet-PCI II controller regains bus ownership.

The PCI Latency Timer cannot be disabled. Systems that support preemption and that want to control the duration of the PCnet-PCI II controller bus mastership period with the DMA Transfer Counter or the Bus Activity Timer must program the PCI Latency Timer with a high value so that it does not expire before the other two registers do.

BUFFER MANAGEMENT UNIT

The Buffer Management Unit (BMU) is a microcoded state machine which implements the initialization procedure and manages the descriptors and buffers. The buffer management unit operates at half the speed of the CLK input.

Initialization

PCnet-PCI II controller initialization includes the reading of the initialization block in memory to obtain the operating parameters. The initialization block can be organized in two ways. When SSIZE32 (BCR20, bit 8) is at its default value of ZERO, all initialization block entries are logically 16-bits wide to be backwards compatible with the Am79C90 C-LANCE and Am79C96x PCnet-ISA family. When SSIZE32 (BCR20, bit 8) is set to ONE, all initialization block entries are logically 32-bits wide. Note that the PCnet-PCI II controller always performs 32-bit bus transfers to read the initialization block entries. 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. Once the initialization block has been completely read in and internal registers have been updated, IDON will be set in CSR0, generating an interrupt (if IENA is set).

The PCnet-PCI II controller obtains the start address of the initialization block from the contents of CSR1 (least significant 16 bits of address) and CSR2 (most significant 16 bits of address). The host must write CSR1 and CSR2 before setting the INIT bit. The initialization block contains the user defined conditions for PCnet-PCI II controller operation, together with the base addresses and length information of the transmit and receive descriptor rings.

There is an alternate method to initialize the PCnet-PCI II controller. Instead of initialization via the initialization block in memory, data can be written directly into the appropriate registers. Either method or a combination of the two may be used at the discretion of the programmer. Please refer to Appendix C for details on this alternate method.

Re-Initialization

The transmitter and receiver sections of the PCnet-PCI II controller can be turned on via the initialization block (DTX, DRX, CSR15, bits 1–0). The states of the transmitter and receiver are monitored by the host through CSR0 (RXON, TXON bits). The PCnet-PCI II controller should be re-initialized 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 was shut off due to the detection of an error condition (MERR, UFLO, TX BUFF error).

Re-initialization 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-PCI II controller as in the CLANCE. In particular, upon restart, the PCnet-PCI II controller reloads the transmit and receive descriptor pointers with their respective base addresses. This means that the software must clear the descriptor OWN bits and reset its descriptor ring pointers before restarting the PCnet-PCI II controller. The reload of descriptor base addresses is performed in the CLANCE only after initialization, so a restart of the CLANCE without initialization leaves the CLANCE pointing at the same descriptor locations as before the restart.

Suspend

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

The host requests the PCnet-PCI II controller to enter the suspend mode by setting SPND (CSR5, bit 0) to ONE. When the host sets SPND to ONE, the PCnet-PCI 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.The host must poll SPND until it reads back ONE to determine that the PCnet-PCI II controller has entered the suspend mode. In suspend mode, all of the CSR and BCR registers are accessible. As long as the PCnet-PCI II controller is not reset while in suspend mode (by H_RESET, S_RESET or by setting the STOP bit), no re-initialization of the device is required after the device comes out of suspend mode. When the host clears SPND, the PCnet-PCI II controller will leave the suspend mode and will continue at the transmit and receive descriptor ring locations, where it had left off.

Buffer Management

Buffer management is accomplished through message descriptor entries organized as ring structures in memory. There are two descriptor rings, one for transmit and one for receive. Each descriptor describes a single buffer. A frame may occupy one or more buffers. If multiple buffers are used, this is referred to as buffer chaining.

Descriptor Rings

Each descriptor ring must occupy a contiguous area of memory. During initialization the user-defined base address for the transmit and receive descriptor rings, as well as the number of entries contained in the descriptor rings are set up. The programming of the software style (SWSTYLE, BCR20, bits 7–0) affects the way the descriptor rings and their entries are arranged.

When SWSTYLE is at its default value of ZERO, the descriptor rings are backwards compatible with the Am79C90 C-LANCE and Am79C96x PCnet-ISA family. The descriptor ring base addresses must be aligned to an 8-byte boundary and a maximum of 128 ring entries is allowed when the ring length is set through the TLEN and RLEN fields of the initialization block. Each ring entry contains a subset of the three 32-bit transmit or receive message descriptors (TMD, RMD) that are organized as four 16-bit structures (SSIZE (BCR20, bit 8) is set to ZERO). Note that even though the PCnet-PCI II controller treats the descriptor entries as 16-bit structures, it will always perform 32-bit bus transfers to access the descriptor entries. The value of CSR2, bits 15–8 is used as the upper 8-bits for all memory addresses during bus master transfers.

When SWSTYLE is set to ONE, TWO or THREE, the descriptor ring base addresses must be aligned to a 16-byte boundary and a maximum of 512 ring entries is allowed when the ring length is set through the TLEN and RLEN fields of the initialization block. Each ring entry is organized as three 32-bit message descriptors (SSIZE32 (BCR20, bit 8) is set to ONE). The fourth DWord is reserved. When SWSTYLE is set to THREE, the order of the message descriptors is optimized to allow read and write access in burst mode.

For any software style, the ring lengths can be set beyond this range (up to 65535) by writing the transmit and receive ring length registers (CSR76, CSR78) 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

To permit the queuing and de-queuing of message buffers, ownership of each buffer is allocated to either the PCnet-PCI II controller or the host. The OWN bit within the descriptor status information, either TMD or RMD, is used for this purpose. When OWN is set to ONE, it signifies that the PCnet-PCI II controller currently has ownership of this ring descriptor and its associated buffer. 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. A device may, however, read from a descriptor that it does not currently own. Software should always read descriptor entries in sequential order. When software finds that the current descriptor is owned by the PCnet-PCI II controller, then the software must not read ahead to the next descriptor. The software should wait at a descriptor it does not own until the PCnet-PCI II controller sets OWN to ZERO to release ownership to the software. (When LAPPEN (CSR3, bit 5) is set to ONE, this rule is modified. See the LAPPEN description.)

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

The following figure illustrates the relationship between the initialization base address, the initialization block, the receive and transmit descriptor ring base addresses, the receive and transmit descriptors and the receive and transmit data buffers, when SSIZE32 is cleared to ZERO.

Note that the value of CSR2, bits 15–8 is used as the upper 8-bits for all memory addresses during bus master transfers.

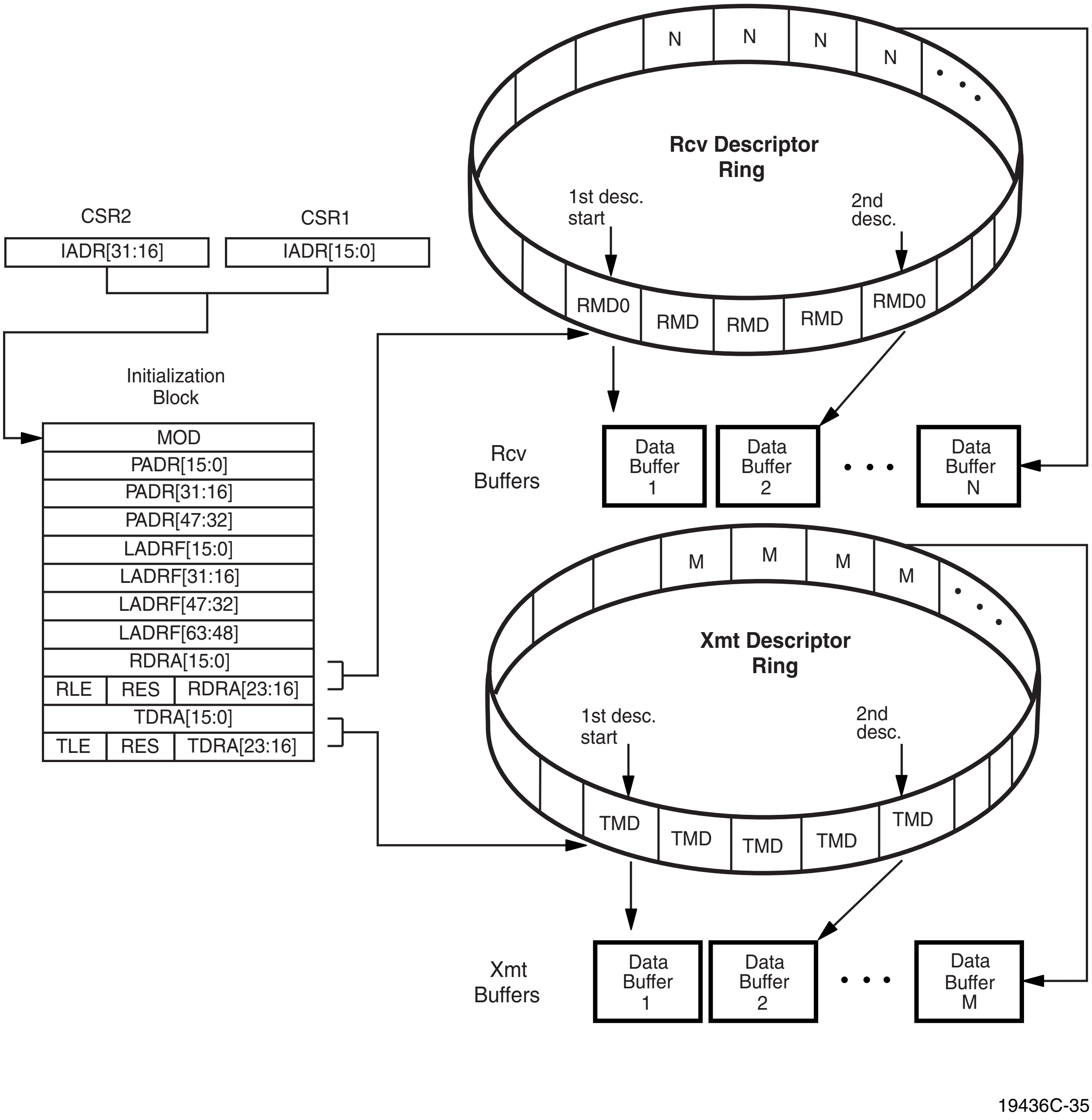
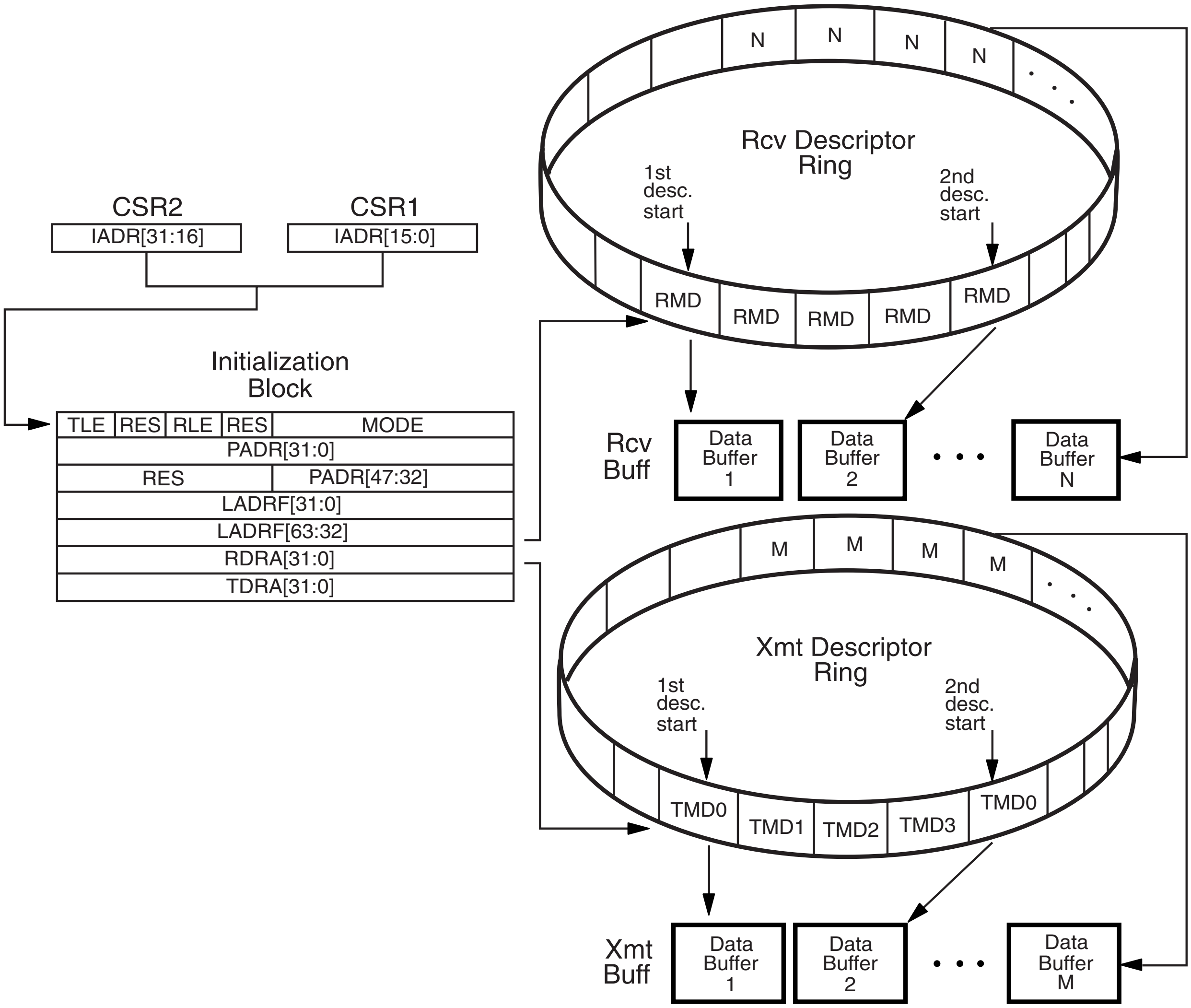


Figure 32. 16-Bit Software Model

The following figure illustrates the relationship between the initialization base address, the initialization block, the receive and transmit descriptor ring base addresses, the receive and transmit descriptors and the receive and transmit data buffers, when SSIZE32 is set to ONE.



19436C-36

Figure 33. 32-bit Software Model

Polling

If there is no network channel activity and there is no pre- or post-receive or pre- or post-transmit activity being performed by the PCnet-PCI II controller, then the PCnet-PCI 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-PCI 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). The accesses will be made in the following order: RMD1, then RMD0 of the current RDTE during one bus arbitration, and after that, TMD1, then TMD0 of the current TDTE during a second bus arbitration. All information collected during polling activity will be stored internally in the appropriate CSRs, if the OWN bit is set. (i.e. CSR18, CSR19, CSR20, CSR21, CSR40, CSR42, CSR50, CSR52).

A typical receive poll is the product of the following conditions:

1. PCnet-PCI II controller does not own the current DTE and the poll time has elapsed and RXON = 1 (CSR0, bit 5), or
2. PCnet-PCI II controller does not own the next RDTE and there is more than one receive descriptor in the ring and the poll time has elapsed and RXON = 1.

If RXON is cleared to ZERO, the PCnet-PCI II controller will never poll RDTE locations.

In order to avoid missing frames the system should have at least on RDTE available. To minimize poll activity two RDTEs should be available. In this case, the poll operation will only consist of the check of the status of the current TDTE.

A typical transmit poll is the product of the following conditions:

1. PCnet-PCI II controller does not own the current TDTE and DPOLL = 0 (CSR4, bit 12) and TXON = 1 (CSR0, bit 4) and the poll time has elapsed, or
2. PCnet-PCI II controller does not own the current TDTE and DPOLL = 0 and TXON = 1 and a frame has just been received, or
3. PCnet-PCI II controller does not own the current TDTE and DPOLL = 0 and TXON = 1 and a frame has just been transmitted.

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. If the microcode is not executing the poll counting code when the TDMD bit is set, then the demanded poll of the TDTE will be delayed until the microcode returns to the poll counting code.

The user may change the poll time value from the default of 65,536 clock periods by modifying the value in the Polling Interval register (CSR47).

Transmit Descriptor Table Entry

If, after a Transmit Descriptor Table Entry (TDTE) access, the PCnet-PCI II controller finds that the OWN bit of that TDTE is not set, the PCnet-PCI 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 the Start of Packet (STP) bit is not set, the PCnet-PCI II controller will immediately request the bus in order to clear the OWN bit of this descriptor. (This condition would normally be found following a late collision (LCOL) or retry (RTRY) error that occurred in the middle of a transmit frame chain of buffers.) After resetting the OWN bit of this descriptor, the PCnet-PCI 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 cleared. In the C-LANCE the buffer length of 0 is interpreted as a 4096-byte buffer. A zero length buffers is acceptable as long as it is not the last buffer in a chain (STP = 0 and ENP = 1).

If the OWN bit and STP are set, then microcode control proceeds to a routine that will enable transmit data transfers to the FIFO. The PCnet-PCI II controller will look ahead to the next transmit descriptor after it has performed at least one transmit data transfer from the first buffer.

If the PCnet-PCI II controller does not own the next TDTE (i.e. the second TDTE for this frame), it will complete transmission of the current buffer and update the status of the current (first) TDTE with the BUFF and UFLO bits being set. If DXSUFLO (CSR3, bit 6) is cleared to ZERO, the underflow error will cause the transmitter to be disabled (CSR0, TXON = 0). The PCnet-PCI II controller will have to be re-initialized to restore the transmit function. Setting DXSUFLO to ONE enables the PCnet-PCI II controller to gracefully recover from an underflow error. The device will scan the transmit descriptor ring until it finds either the start of a new frame or a TDTE it does not own. To avoid an underflow situation in a chained buffer transmission, the system should always set the transmit chain descriptor own bits in reverse order.

If the PCnet-PCI 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 of the first descriptor (clear the OWN bit in TMD1), 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.)

It is imperative that the host system never reads the TDTE OWN bits out of order. The PCnet-PCI II controller normally clears OWN bits in strict FIFO order. However, the PCnet-PCI II controller can queue up to two frames in the transmit FIFO. When the second frame uses buffer chaining, the PCnet-PCI II controller might return ownership out of normal FIFO order. The OWN bit for last (and maybe only) buffer of the first frame is not cleared until transmission is completed. During the transmission the PCnet-PCI II controller will read in buffers for the next frame and clear their OWN bits for all but the last one. The first and all intermediate buffers of the second frame can have their OWN bits cleared before the PCnet-PCI II controller returns ownership for the last buffer of the first frame.

If an error occurs in the transmission before all of the bytes of the current buffer have been transferred, transmit status of the current buffer will be immediately updated. If the buffer does not contain the end of packet, the PCnet-PCI II controller will skip over the rest of the frame which experienced the error. This is done by returning to the polling microcode where the PCnet-PCI II controller will clear the OWN bit for all descriptors with OWN = 1 and STP = 0 and continue in like manner until a descriptor with OWN = 0 (no more transmit frames in the ring) or OWN = 1 and STP = 1 (the first buffer of a new frame) is reached.

At the end of any transmit operation, whether successful or with errors, immediately following the completion of the descriptor updates, the PCnet-PCI II controller will always perform another polling operation. As described earlier, this polling operation will begin with a check of the current RDTE, unless the PCnet-PCI II controller already owns that descriptor. Then the PCnet-PCI II controller will poll the next TDTE. If the transmit descriptor OWN bit has a ZERO value, the PCnet-PCI II controller will resume incrementing the poll time counter. If the transmit descriptor OWN bit has a value of ONE, the PCnet-PCI II controller will begin filling the FIFO with transmit data and initiate a transmission. This end-of-operation poll coupled with the TDTE lookahead operation allows the PCnet-PCI II controller to avoid inserting poll time counts between successive transmit frames.

By default, whenever the PCnet-PCI II controller completes a transmit frame (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 cleared. The PCnet-PCI II controller provides two modes to reduce the number of transmit interrupts. The interrupt of a successfully transmitted frame can be suppressed by setting TINTOKD (CSR5, bit 15) to ONE. Another mode, which is enabled by setting LTINTEN (CSR5, bit 14) to ONE, allows suppression of interrupts for successful transmissions for all but the last frame in a sequence.

Receive Descriptor Table Entry

If the PCnet-PCI II controller does not own both the current and the next Receive Descriptor Table Entry (RDTE) then the PCnet-PCI II controller will continue to poll according to the polling sequence described above. If the receive descriptor ring length is one, then there is no next descriptor to be polled.

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

When receive activity is present on the channel, the PCnet-PCI II controller waits for the complete address of the message to arrive. It then decides whether to accept or reject the frame based on all active addressing schemes. If the frame is accepted the PCnet-PCI II controller checks the current receive buffer status register CRST (CSR41) to determine the ownership of the current buffer.

If ownership is lacking, the PCnet-PCI II controller will immediately perform a final poll of the current RDTE. If ownership is still denied, the PCnet-PCI II controller has no buffer in which to store the incoming message. The MISS bit will be set in CSR0 and the Missed Frame Counter (CSR112) will be incremented. An interrupt will be generated if IENA (CSR0, bit 6) is set to ONE and MISSM (CSR3, bit 12) is cleared to ZERO. Another poll of the current RDTE will not occur until the frame has finished.

If the PCnet-PCI II controller sees that the last poll (either a normal poll, or the final effort described in the above paragraph) of the current RDTE shows valid ownership, 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-PCI II controller will continue to perform receive data DMA transfers to the first buffer. If the frame length exceeds the length of the first buffer, and the PCnet-PCI 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 frame length exceeds the length of the first (current) buffer, and the PCnet-PCI 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. The OWN bit is the only bit modified in the descriptor. Receive data transfers to the second buffer may occur before the PCnet-PCI 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 OWN bit has been updated in 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.

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

Media Access Control

The Media Access Control (MAC) 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 operating 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 section “Full-Duplex Operation’’.

The MAC engine provides programmable enhanced features designed to minimize host supervision, bus utilization, and pre- or post- message processing. These include the ability to disable retries after a collision, dynamic FCS generation on a frame-by-frame basis, automatic pad field insertion and deletion to enforce minimum frame size attributes, automatic re-transmission without reloading the FIFO, and automatic deletion of collision fragments.

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 frames. When APAD_XMT (CSR, bit 11) is set to ONE, 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 (CSR4, bit 10) is set to ONE, 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 frame data) messages to be transmitted and/or received. The use of this feature reduces bus utilization because the pad bytes are not transferred into or out of main memory.

Framing

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 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 frame. The data portion of the frame consists of destination address, source address, length/type, and frame data. The user is responsible for the correct ordering and content in each of these fields in the frame.

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 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, all frame bytes including FCS will be passed unmodified to the receive buffer, regardless of the actual frame length.

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. The PCnet-PCI II controller has the ability to accept runt packets for diagnostics purposes and proprietary networks.

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 (unicast), logical (multicast) and broadcast address reception.

Error Detection

The MAC engine provides several facilities which report and recover from errors on the medium. In addition, it protects the network 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 Transmit Message Descriptor (TMD) and Control and Status Register (CSR) areas:

- The number of transmission retry attempts (ONE, MORE, RTRY, and TRC).
- Whether the MAC engine had to Defer (DEF) due to channel activity.
- Excessive deferral (EXDEF), indicating that the transmitter has experienced Excessive Deferral on this transmit frame, where Excessive Deferral is defined in ISO 8802-3 (IEEE/ANSI 802.3).
- Loss of Carrier (LCAR), 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 first 4 µs after a transmission was 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 it 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 Receive Message Descriptor (RMD) and CSR areas. All received frames are passed to the host regardless of any error. The FRAM error will only be reported if an FCS error is detected and there are a non integral number of bytes in the message.

During the reception, the FCS is generated on every serial bit (including the dribbling bits) coming from the cable, although the internally saved FCS value is only updated on the eighth bit (on each byte boundary). The MAC engine will ignore up to 7 additional bits at the end of a message (dribbling bits), which can occur under normal network operating conditions. The framing error is reported to the user as follows:

- If the number of dribbling bits are 1 to 7 and there is no FCS error, then there is no Framing error (FRAM = 0).
- If the number of dribbling bits are 1 to 7 and there is a FCS error, then there is also a Framing error (FRAM = 1).
- If the number of dribbling bits is ZERO, then there is no Framing error. There may or may not be a FCS error.
- If the number of dribbling bits is EIGHT, then there is no Framing error. FCS error will be reported and the receive message count will indicated one extra byte.

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

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

Media Access Management

The basic requirement for all stations on the network is to provide fairness of channel allocation. The 802.3/Ethernet protocols define 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) after the last activity, before transmitting on the media. The channel is a multidrop communications media (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

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

The ISO 8802-3 (IEEE/ANSI 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 interFrame 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 InterFrame Spacing Part 1 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 interFrame gap following reception, reset the interFrame gap timing if carrier Sense becomes true during the first 2/3 of the interFrame 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-PCI II controller will perform the two part defferral 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 deasserted. During the first part deferral (Inter-Frame Spacing Part1 – IFS1) the PCnet-PCI II controller will defer any pending transmit frame and respond to the receive message. The IPG counter will be cleared to ZERO continuously until the carrier deasserts, 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-PCI II controller will begin timing the second part deferral (Inter-Frame Spacing Part2 – IFS2) of 3.6 µs. Once IFS1 has completed, and IFS2 has commenced, the PCnet-PCI II controller will not defer to a receive frame if a transmit frame is pending. This means that the PCnet-PCI II controller will not attempt to receive the receive frame, since it will start to transmit, and generate a collision at 9.6 µs. The PCnet-PCI II controller will complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

This transmit two part deferral algorithm is implemented as an option which can be disabled using the DXMT2PD bit in 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 0.6–1.6 µs after the transmission ceases). During the time period in which the SQE Test message is expected the PCnet-PCI II controller will not respond to receive carrier sense.

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

Note: *“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-PCI II controller implements a carrier sense “blinding'' period of 4.0 µs length starting from the deassertion 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, the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the PCnet-PCI II controller will defer its transmission. If carrier is detected within the 4.0 to 6.0 µs IFS1 period, the PCnet-PCI II controller will not restart the “blinding” period, but only restart IFS1.

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 ZEROs 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 to a time determined by the random backoff algorithm. If a single retry was required, the ONE bit will be set in the transmit frame status. If more than one retry was required, the MORE bit will be set. If all 16 attempts experienced collisions, the RTRY bit 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 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 transmit message will be flushed from the FIFO.

The ISO 8802-3 (IEEE/ANSI 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:

Note: *"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:*

$$0 \leq r < 2k$$

where

$$k = \min(n,10)."$$

The PCnet-PCI 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 aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. It effectively accelerates the increase in the backoff time in busy networks, and allows nodes not involved in the collision to access the channel whilst 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.

This modified backoff algorithm is enabled when EMBA (CSR3, bit 3) is set to ONE.

TRANSMIT OPERATION

The transmit operation and features of the PCnet-PCI II controller are controlled by programmable options. The PCnet-PCI II controller offers a 272-byte transmit FIFO to provide frame buffering for increased system latency, automatic re-transmission with no FIFO reload, and automatic transmit padding.

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.

The disable FCS generation/transmission feature can be programmed as a static feature or dynamically on a frame by frame basis.

Transmit FIFO Watermark (XMTFW) in CSR80 sets the point at which the BMU requests more data from the transmit buffers for the FIFO. A minimum of XMTFW empty spaces must be available in the transmit FIFO before the BMU will request the system bus in order to transfer transmit frame data into the transmit FIFO.

Transmit Start Point (XMTSP) in CSR80 sets the point when the transmitter actually attempts to transmit a frame onto the media. A minimum of XMTSP bytes must be written to the transmit FIFO for the current frame be- fore transmission of the current frame will begin. (When automatically padded packets are being sent, it is conceivable that the XMTSP is not reached when all of the data has been transferred to the FIFO. In this case, the transmission will begin when all of the frame data has been placed into the transmit FIFO.) The default value of XMTSP is 01b, meaning there has to be 64 bytes in the transmit FIFO to start a transmission.

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 (CSR15, bit 3) or ADD_FCS/NO_FCS (TMD1, bit 29). The transmit frame will be padded by bytes with the value of 00h. The default value of APAD_XMT is 0, which will disable automatic pad generation after H_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 frame (length field as defined in the ISO 8802-3 (IEEE/ANSI 802.3) standard). The length value contained in the message is not used by the PCnet-PCI II controller to compute the actual number of pad bytes to be inserted. The PCnet-PCI 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-PCI 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.

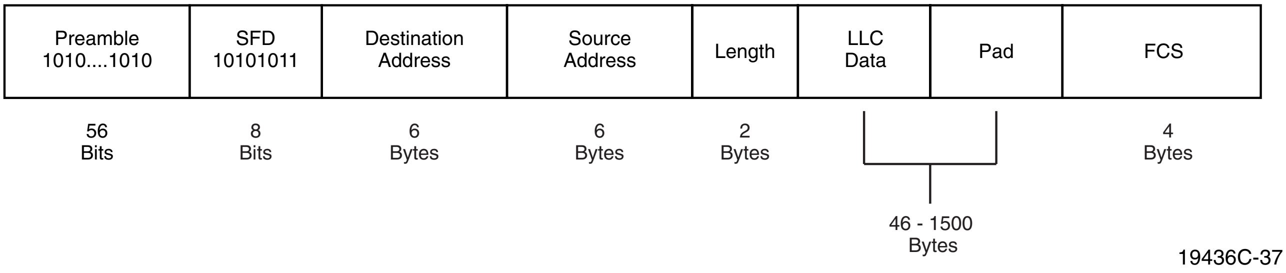


Figure 34. ISO 8802-3 (IEEE/ANSI 802.3) Data Frame

The 544 bit count is derived from the following :

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

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

Preamble/SFD + (Min Frame Size – FCS)

64 + (512 – 32) = 544 bits

A minimum length transmit frame from the PCnet-PCI 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 (CSR15, bit 3). If DXMTFCS is cleared to ZERO, 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-PCI II controller regardless of the state of DXMTFCS or ADD_FCS/NO_FCS (TMD1, bit 29). Note that the calculated FCS is transmitted most significant bit first. The default value of DXMTFCS is 0 after H_RESET.

ADD_FCS (TMD1, bit 29) allows the automatic generation and transmission of FCS on a frame by frame basis. DXMTFCS should be cleared to ZERO in this mode. To generate FCS for a frame, ADD_FCS must be set in the first descriptor of a frame (STP is set to ONE). Note that bit 29 of TMD1 has the function of ADD_FCS if SWSTYLE (BCR20, bits 7–0) is programmed to ZERO, TWO or THREE.

When SWSTYLE is set to ONE for ILACC backwards compatibility, bit 29 of TMD1 changes its function to NO_FCS. When DXMTFCS is cleared to ZERO and NO_FCS is set to ONE in the last descriptor of a frame (ENP is set to ONE), the PCnet-PCI II controller will not generate and append an FCS to a transmit frame.

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-PCI II controller include collisions within the slot time with automatic retry. The PCnet-PCI 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 preamble plus address, length and data fields have been transmitted onto the network without encountering a collision. Note that if DRTY (CSR15, bit 5) is set to ONE or if the network interface is operating in full-duplex mode, no collision handling is required, and any byte of frame data in the FIFO can be overwritten as soon as it is transmitted.

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

Abnormal network conditions include:

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

These conditions should not occur on a correctly configured 802.3 network operating in half-duplex mode, and will be reported if they do. None of these conditions will occur on a network operating in full-duplex mode. (See the section "Full-Duplex Operation" for more detail.)

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 cleared until the STP (the next frame) is found.

Loss of Carrier

When operating in half-duplex mode, a loss of carrier condition will be reported if the PCnet-PCI II controller cannot observe receive activity whilst it is transmitting on the AUI port. In AUI mode, after the PCnet-PCI 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 last bit is transmitted on DO±. 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 retried on the basis of an LCAR error.

When the 10BASE-T port is selected, LCAR will be reported for every frame transmitted while the network interface is in the Link Fail state.

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-PCI II controller will abandon the transmit process for that frame, set Late Collision (LCOL) in the associated TMD2, and process the next transmit frame in the ring. Frames experiencing a late collision will not be retried. 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 10MHz sequence) to be returned via the CI± pair within a 40 network bit-time period after DI± goes inactive (this does not apply if the 10BASE-T port is selected). If the CI± input is not asserted within the 40 network bit-time period following the completion of transmission, then the PCnet-PCI 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 $\overline{\text{INTA}}$ to be activated. It will, however, set the ERR bit CSR0.

Receive Operation

The receive operation and features of the PCnet-PCI II controller are controlled by programmable options. The PCnet-PCI II controller offers a 256-byte receive FIFO to provide frame buffering for increased system latency, automatic flushing of collision fragments (runt packets), automatic receive pad stripping and a variety of address match 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. Acceptance of unicast and broadcast frames can be individually turned off by setting the DRCVPA or DRCVBC bits in CSR15. The Physical Address register (CSR12 to CSR14) stores the address the PCnet-PCI II controller compares to the destination address of the incoming frame for a unicast address match. The Logical Address Filter register (CSR8 to CSR11) serves as a hash filter for multicast address match.

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 H_RESET is 01b which sets the watermark flag at 64 bytes filled.

For test purposes, the PCnet-PCI II controller can be programmed to accept runt packets by setting RPA in CSR124.

Address Matching

The PCnet-PCI II controller supports three types of address matching: unicast, multicast, and broadcast. The normal address matching procedure can be modified by programming three bits in CSR15, the mode register (PROM, DRCVPA, and DRCVBC).

If the first bit received after the start of frame delimiter (the least significant bit of the first byte of the destination address field) is 0, the frame is unicast, which indicates that the frame is meant to be received by a single node. If the first bit received is 1, the frame is multicast, which indicates that the frame is meant to be received by a group of nodes. If the destination address field contains all ONEs, the frame is broadcast, which is a special type of multicast. Frames with the broadcast address in the destination address field are meant to be received by all nodes on the local area network.

When a unicast frame arrives at the PCnet-PCI II controller, the controller will accept the frame if the destination address field of the incoming frame exactly matches the 6-byte station address stored in the Physical Address registers (PADR, CSR12 to CSR14). The byte ordering is such that the first byte received from the network (after the SFD) must match the least significant byte of CSR12 (PADR[7:0]), and the sixth byte received must match the most significant byte of CSR14 (PADR[47:40]).

When DRCVPA (CSR15, bit 13) is set to ONE, the PCnet-PCI II controller will not accept unicast frames.

If the incoming frame is multicast, the PCnet-PCI II controller performs a calculation on the contents of the destination address field to determine whether or not to accept the frame. This calculation is explained in the section that describes the Logical Address Filter (LADRF).

When all bits of the LADRF registers are 0, no multicast frames are accepted, except for broadcast frames.

Although broadcast frames are classified as special multicast frames, they are treated differently by the PCnet-PCI II controller hardware. Broadcast frames are always accepted, except when DRCVBC (CSR15, bit 14) is set.

None of the address filtering described above applies when the PCnet-PCI II controller is operating in the promiscuous mode. In the promiscuous mode, all properly formed packets are received, regardless of the contents of their destination address fields. The promiscuous mode overrides the Disable Receive Broadcast bit (DRCVBC bit l4 in the MODE register) and the Disable Receive Physical Address bit (DRCVPA, CSR15, bit 13).

The PCnet-PCI II controller operates in promiscuous mode when PROM (CSR15, bit 15) is set.

In addition, the PCnet-PCI II controller provides the External Address Detection Interface (EADI) to allow external address filtering. See the section “External Address Detection Interface” for further detail.

The receive descriptor entry RMD1 contains three bits that indicate which method of address matching caused the PCnet-PCI II controller to accept the frame. Note that these indicator bits are only available when the PCnet-PCI II controller is programmed to use 32-bit structures for the descriptor entries (BCR20, bit 7–0, SWSTYLE is set to ONE, TWO or THREE).

PAM (RMD1, bit 22) is set by the PCnet-PCI II controller when it accepted the received frame due to a match of the frame’s destination address with the content of the physical address register.

LAFM (RMD1, bit 21) is set by the PCnet-PCI II controller when it accepted the received frame based on the value in the logical address filter register.

BAM (RMD1, bit 20) is set by the PCnet-PCI II controller when it accepted the received frame because the frame’s destination address is of the type “Broadcast”.

If DRCVBC (CSR15, bit 14) is cleared to ZERO, only BAM, but not LAFM will be set when a Broadcast frame is received, even if the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter. If DRCVBC is set to ONE and the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter, LAFM will be set on the reception of a Broadcast frame.

When the PCnet-PCI II controller operates in promiscuous mode and none of the three match bits is set, it is an indication that the PCnet-PCI II controller only accepted the frame because it was in promiscuous mode.

When the PCnet-PCI II controller is not programmed to be in promiscuous mode, but the EADI interface is enabled, then when none of the three match bits is set, it is an indication that the PCnet-PCI II controller only accepted the frame because it was not rejected by driving the $\overline{\text{EAR}}$ pin LOW within 64 bytes after SFD.

Table 6. Receive Address Match

PAM	LAFM	BAM	DRCVBC	Comment
0	0	0	X	Frame accepted due to PROM = 1 or no EADI reject
1	0	0	X	Physical Address Match
0	1	0	0	Logical Address Filter Match; Frame is not of Type Broadcast
0	1	0	1	Logical Address Filter Match; Frame can be of Type Broadcast
0	0	1	0	Broadcast Frame

Automatic Pad Stripping

During reception of an 802.3 frame the pad field can be stripped automatically. Setting ASTRP_RCV (CSR4, bit 0) to ONE 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 ISO 8802-3 (IEEE/ANSI 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.

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

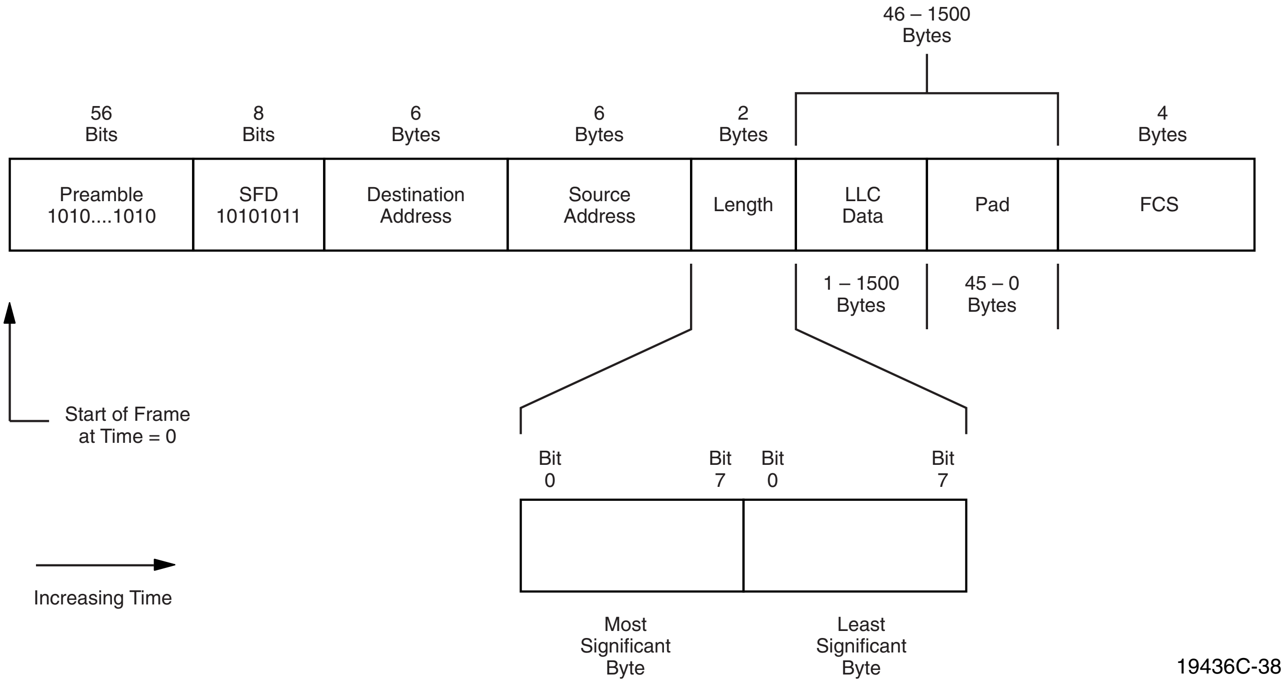


Figure 35. 802.3 Frame And Length Field Transmission Order

Since any valid Ethernet Type field value will always be greater than a normal 802.3 Length field (≥ 46), the PCnet-PCI 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 if pad stripping is enabled.

Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the PCnet-PCI II controller. Note that if the Automatic Pad Stripping feature is enabled, the FCS for padded frames will be verified against the value computed for the incoming bit stream including pad characters, but the FCS value for a padded frame will not be passed to the host. If an FCS error is detected in any frame, the error will be reported in 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-PCI II controller are basically collisions within the slot time and automatic runt packet rejection. The PCnet-PCI 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 and the network interface is operating in half-duplex mode. This criterion 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

Host related receive exception conditions include MISS, BUFF, and OFLO. These are described in the section "Buffer Management Unit''.

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 basic types of loopback. In internal loopback mode, the transmitted data is looped back to the receiver 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, in external loopback mode, data can be transmitted to and received from the external network.

Loopback operation is enabled by setting LOOP (CSR15, bit 2) to ONE. The mode of loopback operation is dependent on the active network port and on the settings of the control bits INTL (CSR15, bit 6), MENDECL (CSR15, bit 10) and TMAULOOP (BCR2, bit 14). The setting of the full-duplex control bits in BCR9 has no effect on the loopback operation.

AUI Loopback Modes

When AUI is the active network port there are three modes of loopback operation: internal with and without MENDEC and external loopback. The setting of TMAULOOP has no effect for this port.

When INTL and MENDECL are set to ONE, internal loopback without MENDEC is selected. Data coming out of the transmit FIFO is fed directly to the receive FIFO. The AUI transmitter is disabled and signals on the receive and collision inputs are ignored.

When INTL is set to ONE and MENDECL is cleared to ZERO, internal loopback including the MENDEC is selected. Data is routed from the transmit FIFO through the MENDEC back to the receive FIFO. No data is transmitted to the network. All signals on the receive and collision inputs are ignored.

External loopback operation is selected by setting INTL to ZERO. The programming of MENDECL has no effect in this mode. The AUI transmitter is enabled and data is transmitted to the network. The PCnet-PCI II controller expects data to be looped back to the receive inputs outside the chip. Collision detection is active in this mode.

T-MAU Loopback Modes

When T-MAU is the active network port there are four modes of loopback operation: internal loopback with and without MENDEC and two external loopback modes.

When INTL and MENDECL are set to ONE, internal loopback without MENDEC is selected. Data coming out of the transmit FIFO is fed directly to the receive FIFO. The T-MAU does not transmit any data to the network, but it continues to send link pulses. All signals on the receive inputs are ignored. LCAR (TMD2, bit 27) will always read ZERO, regardless of the link state. The programming of TMAULOOP has no effect.

When INTL is set to ONE and MENDECL is cleared to ZERO, internal loopback including the MENDEC is selected. Data is routed from the transmit FIFO through the MENDEC back to the receive FIFO. The T-MAU does not transmit any data to the network, but it continues to send link pulses. All signals on the receive inputs are ignored. LCAR (TMD2, bit 27) will always read ZERO, regardless of the link state. The programming of TMAULOOP has no effect.

External loopback operation works slightly different when the T-MAU is the active network port. In a 10BASE-T network, the hub does not generate a receive carrier back to the PCnet-PCI II controller while the chip is transmitting. The T-MAU provides this function internally. A true external loopback covering all the components on the printed circuit board can only be performed by using a special connector that connects the transmit pins of the RJ-45 jack to its receive pins, namely, pin 1 connected to pin 3 and pin 2 connected to pin 6. When INTL is cleared to ZERO and TMAULOOP is set to ONE, data is transmitted to the network and is expected to be routed back to the chip. Collision detection is disabled in this mode. The link state machine is forced into the link pass state. LCAR will always read ZERO. The programming of MENDECL has no effect in this mode.

The PCnet-PCI II controller provides a special external loopback mode that allows the device to be connected to a live 10BASE-T network. The virtual external loopback mode is invoked by setting INTL and TMAULOOP to ZERO. In this mode, data coming out of the transmit FIFO is fed directly into the receive FIFO. Additionally, all transmit data is output to the network. The link state machine is active as is the collision detection logic. The programming of MENDECL has no effect in this mode.

Miscellaneous Loopback Features

All transmit and receive function programming, such as automatic transmit padding and receive pad stripping, operates identically in loopback as in normal operation.

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 C-LANCE (Am79C90) software.

Since the PCnet-PCI II controller has two FCS generators there are no more restrictions on FCS generation or checking or on testing multicast address detection as they exist in the half-duplex PCnet family devices and in the C-LANCE and ILACC. On receive the PCnet-PCI II controller now provides true FCS status. The descriptor for a frame with an FCS error will have the FCS bit (RMD1, bit 27) set to ONE. The FCS generator on the transmit side can still be disabled by setting DXMTFCS (CSR15, bit 3) to ONE.

In internal loopback operation the PCnet-PCI II controller provides a special mode to test the collision logic. When FCOLL (CSR15, bit 4) is set to ONE, a collision is forced during every transmission attempt. This will result in a Retry error.

Magic Packet Mode

Magic Packet mode is enabled by performing three steps. First, the PCnet-PCI II controller must be put into suspend mode (see description of CSR5, bit 0), allowing any current network activity to finish. Next, MPMODE (CSR5, bit 1) must be set to ONE if it has not been set already. Finally, either SLEEP must be asserted (hardware control) or MPEN (CSR5, bit 2) must be set to ONE (software control).

In Magic Packet mode, the PCnet-PCI II controller remains fully powered-up (all V_{DD} and V_{DDB} pins must remain at their supply levels). The device will not generate any bus master transfers. No transmit operations will be initiated on the network. The device will continue to receive frames from the network, but all frames will be automatically flushed from the receive FIFO. Slave accesses to the PCnet-PCI II controller are still possible. Magic Packet mode can be disabled at any time by deasserting SLEEP or clearing MPEN.

A Magic Packet frame is a frame that is addressed to the PCnet-PCI II controller and contains a data sequence in its data field made up of sixteen consecutive physical addresses (PADR[47:0]). The PCnet-PCI II controller will search incoming frames until it finds a Magic Packet frame. The device starts scanning for the sequence after processing the Length field of the frame. The data sequence can begin anywhere in the data field of the frame, but must be detected before the PCnet-PCI II controller reaches the frame's FCS field. The PCnet- PCI II controller is designed such that it does not need the synchronization sequence (6 bytes of all ONEs ("FFFFFFFFFFFFh") at the beginning of the data field), to correctly recognize the proper data sequence. However, any deviation of the incoming frame's Magic Packet data sequence from the required physical address sequence, even by a single bit, will prevent the detection of that frame as a Magic Packet frame.

The PCnet-PCI II controller supports two different modes of address detection for a Magic Packet frame. If MPPLBA (CSR5, bit 5) is at its default value of ZERO, the PCnet-PCI II controller will only detect a Magic Packet frame if the destination address of the frame matches the content of the physical address register (PADR). If MPPLBA is set to ONE, the destination address of the Magic Packet frame can be unicast, multicast, or broadcast. Note that the setting of MPPLBA only effects the address detection of the Magic Packet frame. The Magic Packet data sequence must be made up of sixteen consecutive physical addresses (PADR[47:0]), even if the packet contains a valid destination address that is not the physical address.

When the PCnet-PCI II controller detects a Magic Packet frame, it sets MPINT (CSR5, bit 4) to ONE. If INEA (CSR0, bit 6) and MPINTE (CSR5, bit 3) are set to ONE, INTA will be asserted. The interrupt signal can be used wake up the system. As an alternative, one of the four LED pins can be programmed to indicate that a Magic Packet frame has been received. MPSE (BCR4–7, bit 9) must be set to ONE to enable that function. Note that the polarity of the LED pin can be programmed to be active High by setting LEDPOL (BCR4–7, bit 14) to ONE.

Once a Magic Packet frame is detected, the PCnet-PCI II controller will discard the frame internally, but will not resume normal transmit and receive operations until SLEEP is deasserted or MPEN is cleared, disabling Magic Packet mode. Once either of these events has occurred indicating that the system has detected the assertion of INTA or an LED pin and is now "awake", the controller will continue polling the receive and transmit descriptor rings where it left off. Reinitialization should not be performed.

If Magic Packet mode is disabled by the deassertion of SLEEP, then in order to immediately reenable Magic Packet mode, the SLEEP pin must remain deasserted for at least 200 ns before it is reasserted. If Magic Packet mode is disabled by clearing MPEN, then it may be immediately reenabled by setting MPEN back to ONE.

The bus interface clock (CLK) must continue running if INTA is used to indicate the detection of a magic packet. A system that wants to stop the clock during Magic Packet mode should use one of the LED pins as an indicator of Magic Packet frame detection. It should also stop the clock after enabling Magic Packet mode, other- wise PCI bus activity, including accessing CSR5 to set MPMODE and possibly MPEN to a ONE, could be affected. The clock should be restarted before

Magic Packet mode is disabled if MPEN is being cleared or the clock must be restarted right after magic packet mode is disabled if $\overline{\text{SLEEP}}$ is being deasserted. Otherwise, the receive FIFO may overflow if new frames arrive. The network clock (XTAL1) must continue running at all times while in Magic Packet mode.

MANCHESTER ENCODER/DECODER

The integrated Manchester Encoder/Decoder (MENDEC) provides the PLS (Physical Layer Signaling) functions required for a fully compliant ISO 8802-3 (IEEE/ANSI 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-PCI 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 following crystal specification may be used to ensure less than ±0.5 ns jitter at DO±:

Table 7. Crystal Characteristics

Parameter	Min	Nom	Max	Units
1. Parallel Resonant Frequency		20		MHz
2. Resonant Frequency Error	–50		+50	PPM
3. Change in Resonant Frequency With Respect To Temperature (0 – 70 C)*	–40		+40	PPM
4. Crystal Load Capacitance	20		50	pF
5. Motional Crystal Capacitance (C1)		0.022		pF
6. Series Resistance			35	ohm
7. Shunt Capacitance			7	pF
8. Drive Level			TBD	mW

** Requires trimming specification, not trim is 50 PPM total.*

External Clock Drive Characteristics

When driving the oscillator from a CMOS level 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±.

Table 8. External Clock Source Characteristics

Clock Frequency:	20 MHz ±0.01%
Rise/Fall Time (tR/tF):	<= 6 ns from 0.5 V to VDD –0.5 V
XTAL1 HIGH/LOW Time (tHIGH/tLOW):	20 ns min.
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-PCI II controller. The crystal frequency is divided by two to create the internal transmit clock reference. Both the 10 MHz and 20 MHz clocks are fed into the Manchester Encoder. The internal transmit clock is used by the MENDEC to synchronize the Internal Transmit Data (ITXDAT) and Internal Transmit Enable (ITXEN) from the controller. 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 0.01% timing reference. If an external crystal is used, the accuracy requirements are tighter because allowance for the on-board parasitics 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):

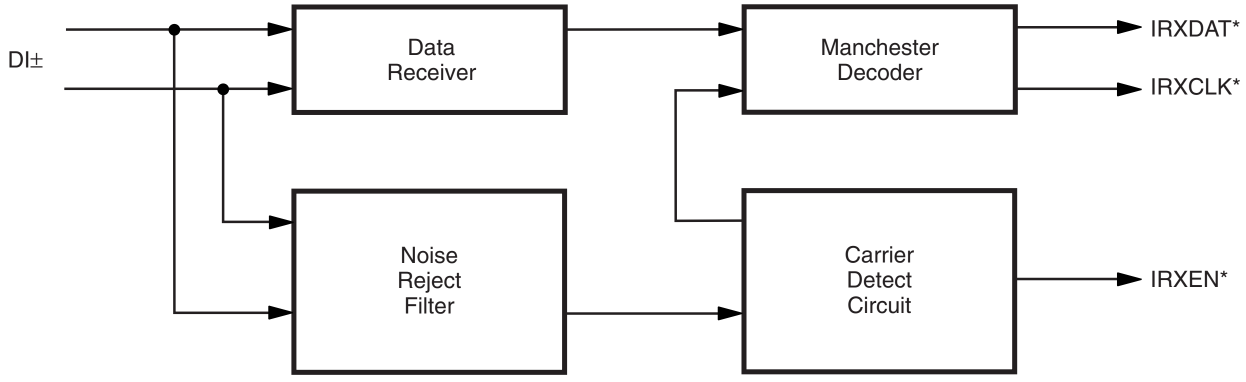
Table 9. TSEL Effect

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).

Receiver Path

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

The receiver section (see the figure below) 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.



19436C-39

*Internal signal

Figure 36. Receiver Block Diagram

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.

The Carrier Detection circuitry detects the presence of an incoming data frame 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±, the internal enable signal from the MENDEC to controller (IRXEN) is asserted and 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 the internal transmit clock. The first negative clock transition (bit cell center of first valid Manchester ZERO) after IRXEN is asserted interrupts the receive oscillator. The oscillator is then restarted at the second Manchester ZERO (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 1010b Manchester bit pattern.

IRXCLK and 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 into bit cell 5, the controller portion of the PCnet-PCI II controller sees the first IRXCLK transition. This also strobes in the incoming fifth bit to the MENDEC as Manchester ONE. IRXDAT may make a transition after the IRXCLK rising edge in bit cell 5, but its state is still undefined. The Manchester ONE 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. Hence, input data jitter is reduced in IRXCLK by 10 to 1.

Carrier Tracking and End of Message

The carrier detection circuit monitors the DI± inputs after IRXEN is asserted for an end of message. IRXEN deasserts 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 IRXEN 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.

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).

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. The clock is phase-locked to the negative transition at the bit cell center of the second ZERO 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 criterion 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

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

After the PCnet-PCI 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 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 (TMD2, bit 27) after the frame has been transmitted.

Differential Input Termination

The differential input for the Manchester data (DI±) is externally terminated by two 40.2 Ω resistors and one optional common-mode bypass capacitor, as shown in the diagram below. The differential input impedance, ZIDF, and the common-mode input impedance, ZICM, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. If SIP devices are used, 39 ohms is also a suitable value. The CI± differential inputs are terminated in exactly the same way as the DI± pair.

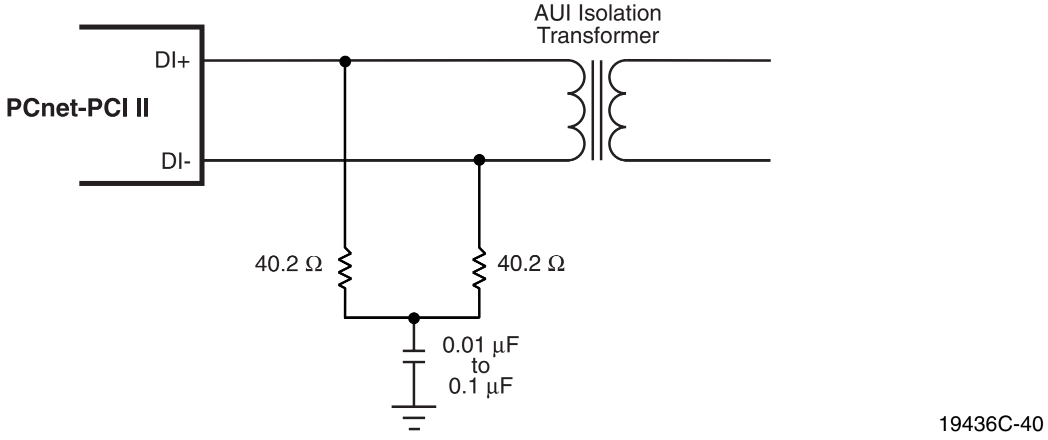


Figure 37. AUI Differential Input Termination

Collision Detection

A MAU detects the collision condition on the network and generates a 10 MHz 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 ICLSN line HIGH. The condition continues for approximately 1.5 bit times after the last LOW-to-HIGH transition on CI±.

Twisted-pair Transceiver

This section describes operation of the Twisted Pair Transceiver (T-MAU) when operating in 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 section "Full-Duplex Operation''.

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 ISO 8802-3 (IEEE/ANSI 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 ISO 8802-3 (IEEE/ANSI 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 and crosstalk noise 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 defined to account for a 1 dB insertion loss at 10 MHz, which is typical for the type of receive filters/transformers employed.

Normal 10BASE-T compatible receive thresholds are employed when the LRT bit (CSR15, bit 9) is cleared to ZERO. When the LRT bit is set to ONE, the Low Receive Threshold option is invoked, and the sensitivity of the T-MAU receiver is increased. This allows longer line lengths to be employed, exceeding the 100 m target distance of normal 10BASE-T (assuming typical 24 AWG cable). The 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, it is recommended that when using the Low Receive Threshold option that the service should be installed on 4-pair cable only.

Multipair 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 the 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 T-MAU to go into a Link Fail state. In the Link Fail state, data trans- mission, data reception, data loopback and the collision detection functions are disabled, and remain disabled until valid data or more than five consecutive link pulses appear on the RXD± pair. During Link Fail, the Link Status signal is inactive. When the link is identified as functional, the Link Status signal is asserted. The $\overline{\text{LNKST}}$ pin displays the Link Status signal by default.

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. If T-MAU is selected using the PORTSEL bits in CSR15, the T-MAU will be forced into the Link Fail state when moving from AUI to T-MAU selection.

Transmission attempts during Link Fail state will produce no network activity and will produce LCAR and CERR error indications.

In order to interoperate with systems which do not implement Link Test, this function can be disabled by setting the DLNKTST bit in CSR15. 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 frames 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 H_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 frames 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 is made due to 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 received signal with a positive amplitude greater than 585 mV (LRT = 1) 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 received signals with a negative amplitude greater than 585 mV with a pulse width of 60–200 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 frames with valid ETD of identical polarity are detected. When “armed”, the receiver is capable of changing the initial or previous polarity configuration based on the ETD polarity.

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

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 Bus Configuration Registers (BCR4 to BCR7).

Twisted Pair Interface Status

When the T-MAU is in Link Pass state, three signals (XMT, RCV and COL) indicate whether the T-MAU is transmitting, receiving, or in a collision state with both functions active simultaneously. These signals are internal signals that can be programmed to appear on any of the LED output pins. Programming is done by writing to BCR4 to BCR7.

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

Collision Detection Function

Activity on both twisted pair signals RXD± and TXD± at the same time constitutes a collision, thereby causing the internal COL signal to be activated. COL will remain active until one of the two colliding signals changes from active to idle. However, transmission attempt in Link Fail state results in LCAR and CERR indication. COL stays active for 2 bit times at the end of a collision.

Signal Quality Error Test Function

The Signal Quality Error (SQE) test function (also called Heartbeat) is disabled when the 10BASE-T port is selected.

Jabber Function

The Jabber function prevents the twisted pair transmit function of the T-MAUTXD± from being active for an excessive period of time (20 ms to 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 is asserted. Once the transmit data stream is removed, the T-MAU waits an "unjab" time of 250 ms to 750 ms before it deasserts COL and re-enables the transmit circuitry.

Power Down

The T-MAU circuitry can be made to go into a power savings mode. The T-MAU will go into the power down mode when H_RESET is active, when coma mode is active, or when the T-MAU is not selected. Refer to the section "Power Savings Modes" for descriptions 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 driven LOW, and the internal T-MAU status signals ($\overline{\text{LNKST}}$, RCVPOL, XMT, RCV and COL) signals are inactive.

After coming out of the power down mode, the T-MAU will remain in the reset state for an additional 10 µs. Immediately after the reset condition is removed, 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 RD± pair.

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

10BASE-T Interface Connection

The figure below shows the proper 10BASE-T network interface design. Refer to Appendix A for a list of compatible 10BASE-T filter/transformer modules.

Note that the recommended resistor values and filter and transformer modules are the same as those used by the IMR+ (Am79C981).

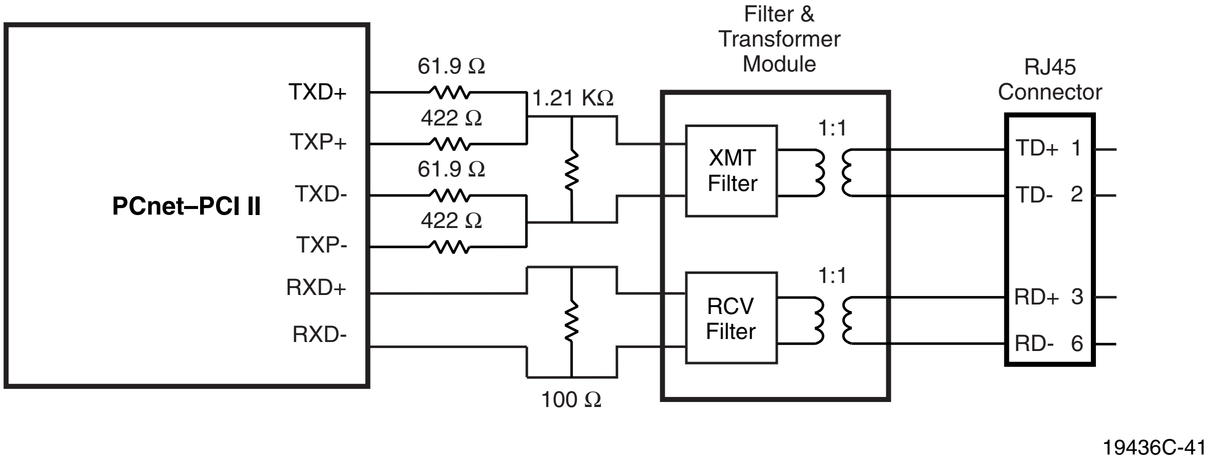


Figure 38. 10BASE-T Interface Connection

Full-Duplex Operation

The PCnet-PCI II controller supports full-duplex operation on all three network interfaces: AUI and 10BASE-T. 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. Full-duplex operation is enabled by the FDEN and AUIFD bits located in BCR9.The PCnet-PCI II controller does not support IEEE standard auto-negotiation between half-duplex and full-duplex operation. When operating in full-duplex mode, the following changes to the device operation are made:

Bus Interface/Buffer Management Unit changes:

- The first 64 bytes of every transmit frame are not pre- served in the transmit FIFO during transmission of the first 512 bits as described in the section "Transmit Exception Conditions". 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.
- Successful reception of the first 64 bytes of every receive frame is not a requirement for receive DMA to begin as described in the section "Receive Exception Condition". Instead, receive DMA will be requested as soon as either the Receive FIFO Watermark (CSR80, bits 13–12) is reached or a complete valid receive frame is detected, 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:

- Changes to the Transmit Deferral mechanism:
  - — Transmission is not deferred while receive is active.
  - — 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.
- When the AUI 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 Link Fail state.)
- The 4.0 μs carrier sense blinding period after a transmission during which the SQE test normally occurs is disabled.
- When the AUI port is active, the SQE Test error reporting (CERR) is disabled. (CERR is still reported when the 10BASE-T port is active if a packet is transmitted while in Link Fail state.)
- The collision indication input to the MAC engine is ignored.

T-MAU changes:

- The internal transmit to receive feedback path which is used to indicate carrier sense during normal transmission in half-duplex mode is disabled.
- The collision detect circuit is disabled.
- The SQE test function is disabled.

Full-Duplex Link Status LED Support

The PCnet-PCI II controller provides a bit in each of the LED Status registers (BCR4, BCR5, BCR6, BCR7) to display the Full-Duplex Link Status. If the FDLSE bit (bit 8) is set, a value of ONE will be sent to the associated LEDOUT bit when the T-MAU is in the Full-Duplex Link Pass state.

External Address Detection Interface

The External Address Detection Interface (EADI) is provided to allow external address filtering. It is selected by setting the EADISEL bit in BCR2 to ONE. This feature is typically utilized for terminal servers, bridges and/or router products. The EADI interface can be used in conjunction with external logic to capture the packet destination address from the serial bit stream as it arrives at the PCnet-PCI II controller, compare the captured address with a table of stored addresses or identifiers, and then determine whether or not the PCnet-PCI II controller should accept the packet.

The EADI interface outputs are delivered directly from the NRZ decoded data and clock recovered by the Manchester decoder. This allows the external address detection to be performed in parallel with frame reception and address comparison in the MAC Station Address Detection (SAD) block of the PCnet-PCI II controller.

SRDCLK is provided to allow clocking of the receive bit stream into the external address detection logic. Note that when the 10BASE-T port is selected, transitions on SRDCLK will only occur during receive activity. When the AUI port is selected, transitions on SRDCLK will occur during both transmit and receive activity. Once a received frame commences and data and clock are available from the decoder, the EADI logic will monitor the alternating (1,0) preamble pattern until the two ONEs of the Start Frame Delimiter (SFD, 10101011 bit pattern) are detected, at which point the SFBD output will be driven HIGH.

The SFBD signal will initially be LOW. The assertion of SFBD is a signal to the external address detection logic that the SFD has been detected and that subsequent SRDCLK cycles will deliver packet data to the external logic. Therefore, when SFBD is asserted, the external address matching logic should begin de-serialization of the SRD data and send the resulting destination address to a Content Addressable Memory (CAM) or other address detection device. In order to reduce the amount of logic external to the PCnet-PCI II controller for multiple address decoding systems, the SFBD signal will toggle at each new byte boundary within the packet, subsequent to the SFD. This eliminates the need for externally supplying byte framing logic.

SRD is the decoded NRZ data from the network. This signal can be used for external address detection. Note that when the 10BASE-T port is selected, transitions on SRD will only occur during receive activity. When the AUI port is selected, transitions on SRD will occur during both transmit and receive activity.

The EAR pin should be driven LOW by the external address comparison logic to reject a frame.

If an address match is detected by comparison with either the Physical Address or Logical Address Filter registers contained within the PCnet-PCI II controller or the frame is of the type “Broadcast”, then the frame will be accepted regardless of the condition of EAR. When the EADISEL bit of BCR2 is set to ONE and the PCnet-PCI II controller is programmed to promiscuous mode (PROM bit of the Mode Register is set to ONE), then all incoming frames will be accepted, regardless of any activity on the EAR pin.

Internal address match is disabled when PROM (CSR15, bit 15) is cleared to ZERO, DRCVBC (CSR15, bit 14) and DRCVPA (CSR15, bit 13) are set to ONE and the Logical Address Filter registers (CSR8 to CSR11) are programmed to all ZEROs.

When the EADISEL bit of BCR2 is set to ONE and internal address match is disabled, then all incoming frames will be accepted by the PCnet-PCI II controller, unless the EAR pin becomes active during the first 64 bytes of the frame (excluding preamble and SFD). This allows external address lookup logic approximately 58 byte times after the last destination address bit is available to generate the EAR signal, assuming that the PC-net-PCI II controller is not configured to accept runt packets. The EADI logic only samples EAR from 2 bit times after SFD until 512 bit times (64 bytes) after SFD. The frame will be accepted if EAR has not been asserted during this window. If Runt Packet Accept (CSR124, bit 3) is enabled, then the EAR signal must be generated prior to the receive message completion, if frame rejection is to be guaranteed. Runt packet sizes 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. EAR must have a pulse width of at least 110 ns.

Note that when the PCnet-PCI II controller is operating in full-duplex mode or runt packet accept is turned on (CSR124, bit 3) the Receive FIFO Watermark (CSR80, bits 13–12) must be programmed to 64 (01b) or 128 (10b) to allow the full window of 512 bit times after SFD for the assertion of EAR. If the watermark was programmed to 16 (00b), receive FIFO DMA could start before EAR is asserted to reject the frame.

The EADI outputs continue to provide data throughout the reception of a frame. This allows the external logic to capture frame header information to determine protocol type, inter-networking information, and other useful data.

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 are set). This configuration is useful as a semi-power-down mode in that the PCnet-PCI II controller will not perform any power-consuming DMA operations. However, external circuitry can still respond to control frames on the network to facilitate remote node control.

The table below summarizes the operation of the EADI interface:

Table 10. EADI Operations

PROM	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 110 ns during the window from 2 bits after SFD to 512 bits after SFD	PCnet-PCI II controller internal physical address and logical address filter matches and broadcast frames

Expansion ROM Interface

The Expansion ROM is an 8-bit ROM connected to the PCnet-PCI II controller Expansion ROM Data bus (ERD). It can be of up to 64 Kbytes in size. The Expansion ROM Address bus (ERA) is 8 bits wide. An external latch is required to store the upper 8 bits of the 16-bit address to the ROM. All ERA outputs are forced to a constant level to conserve power while no access to the Expansion ROM is performed.

EROE is asserted during the Expansion ROM read operation. This signal can be used to control the OE input of the ROM. EROE can be left unconnected and the OE input of the ROM can be tied to ground to always enable the ROM data outputs. The CE input of the ROM can either be tied to ground or it can also be connected to EROE.

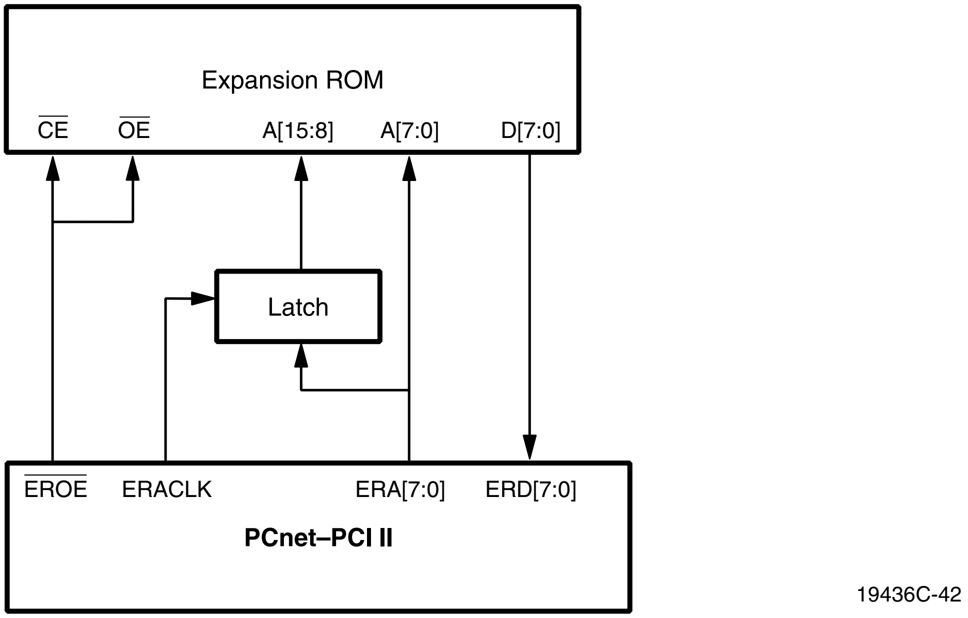


Figure 39. Expansion ROM Interface

The signal ERACLK is provided to strobe the upper 8 bits of the address into an external latch. The timing relation of ERACLK to ERA is such that both ‘373 (transparent latch) and ‘374 (D flip-flop) types of address latch can be used.

The PCnet-PCI II controller will always read four bytes for every host Expansion ROM read access. The interface to the Expansion ROM runs synchronous to the PCI bus interface clock. The PCnet-PCI II controller will start the read operation to the Expansion ROM by driving the upper 8-bits of the Expansion ROM address on ERA[7:0]. This happens in the same clock cycle that the device claims the transfer by asserting $\overline{\text{DEVSEL}}$. One clock later, $\overline{\text{EROE}}$ is asserted and ERACLK goes high to allow latching of the upper address bits externally. The upper portion of the Expansion ROM address will be the same for all four byte read cycles. ERACLK is asserted for one clock. ERA[7:0] are driven with the upper 8-bits of the Expansion ROM address for one more clock cycle after ERACLK goes low. Next, the PCnet-PCI II controller starts driving the lower 8 bits of the Expansion ROM address on ERA[7:0].

The time the PCnet-PCI II controller waits for data to be valid is programmable. ROMTMG (BCR18, bits 15–12) defines the time from when the PCnet-PCI II controller drives ERA[7:0] with the lower 8-bits of the Expansion ROM address to when the PCnet-PCI II controller latches in the data on the ERD[7:0] inputs. The register value specifies the time in number of clock cycles. When ROMTMG is set to Nine (the default value), ERD[7:0] is sampled with the next rising edge of CLK nine clock cycles after ERA[7:0] was driven with a new address value. The clock edge that is used to sample the data is also the clock edge that generates the next Expansion ROM address. Only the first three bytes of Expansion ROM data are stored in holding registers. The fourth byte is passed directly from the ERD[7:0] inputs to the AD[31:24] outputs. One clock cycle after the last data byte is available, PCnet-PCI II controller asserts $\overline{\text{TRDY}}$. Two clock cycles after the data is transferred on the PCI bus, $\overline{\text{EROE}}$ is deasserted.

The access time for the Expansion ROM device (t_{ACC}) can be calculated by subtracting the clock to output delay for the ERA[7:0] outputs (t_{VAL}(ERA)) and the input to clock setup time for the ERD[7:0] inputs (t_{SU}(ERD)) from the time defined by ROMTMG:

$t_{ACC} \leq$ ROMTMG* clock period – t_{VAL}(ERA) – t_{SU}(ERD)

For an adapter card application, the value used for clock period should be 30 ns to guarantee correct interface timing at the maximum clock frequency of 33 MHz.

The timing diagram below assumes the default programming of ROMTMG (1001b = 9 CLK). After reading the first byte, the PCnet-PCI II controller reads in three more bytes by incrementing the lower portion of the ROM address. After the last byte is strobed in, TRDY will be asserted on clock 44. When the host tries to perform a burst read of the Expansion ROM, the PCnet-PCI II will disconnect the access at the second data phase.

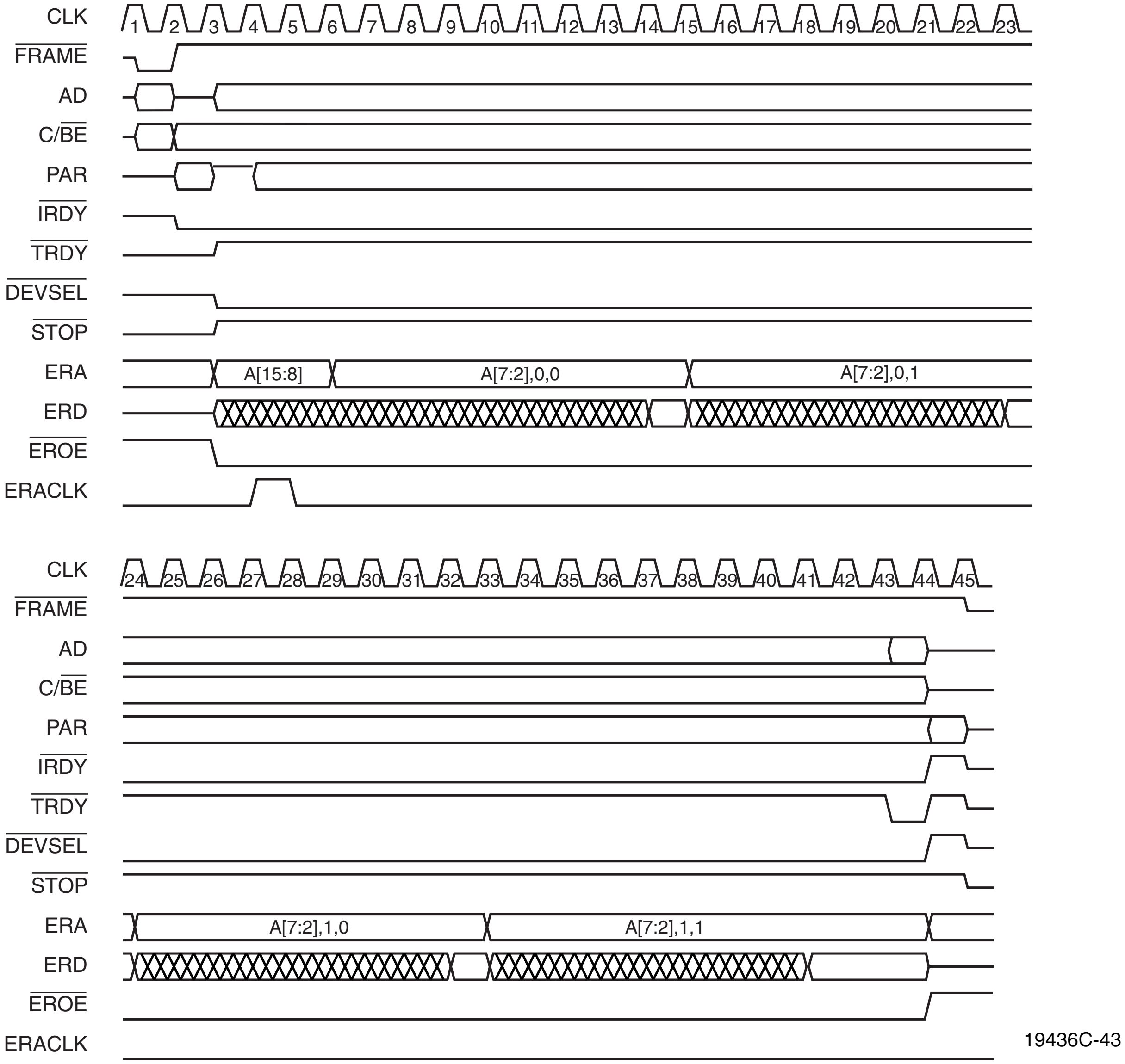


Figure 40. Expansion ROM Bus Read Sequence

The host must program the Expansion ROM Base Address register in the PCI configuration space before the first access to the Expansion ROM. The PCnet-PCI II controller will not react to any access to the Expansion ROM until both MEMEN (PCI Command register, bit 1) and ROMEN (PCI Expansion ROMBase Address register, bit 0) are set to ONE. After the Expansion ROM is enabled, the PCnet-PCI II controller will claim all memory read accesses with an address between ROMBASE and ROMBASE + 64K – 4 (ROMBASE, PCI Expansion ROM Base Address register, bits 31–11). The address output to the Expansion ROM is the offset from the address on the PCI bus to ROMBASE. The PCnet-PCI II controller aliases all accesses to the Expansion ROM of the command types “Memory Read Multiple” and “Memory Read Line” to the basic Memory Read command.

Since setting MEMEN also enables memory mapped access to the I/O resources, attention must be given to the PCI Memory Mapped I/O Base Address register, before enabling access to the Expansion ROM. The host must set the PCI Memory Mapped I/O Base Address register to a value that prevents the PCnet-PCI II controller from claiming any memory cycles not intended for it.

During the boot procedure the system will try to find an Expansion ROM. A PCI system assumes that an Expansion ROM is present when it reads the ROM signature 55h (byte 0) and AAh (byte 1). A design without Expansion ROM can guarantee that the Expansion ROM detection fails by connecting two adjacent ERD pins together and tying them high or low.

EEPROM Microwire Interface

The PCnet-PCI II controller contains a built-in capability for reading and writing to an external serial EEPROM. This built-in capability consists of an interface for direct connection to a Microwire compatible EEPROM, an automatic EEPROM read feature, and a user-programmable register that allows direct access to the Microwire interface pins.

Automatic EEPROM Read Operation

Shortly after the deassertion of the $\overline{\text{RST}}$ pin, the PCnet-PCI II controller will read the contents of the EEPROM that is attached to the Microwire interface. Because of this automatic read capability of the PCnet-PCI II controller, an EEPROM can be used to program many of the features of the PCnet-PCI II controller at power-up, allowing system-dependent configuration information to be stored in the hardware, instead of inside the device driver.

If an EEPROM exists on the Microwire interface, the PCnet-PCI II controller will read the EEPROM contents at the end of the H_RESET operation. The EEPROM contents will be serially shifted into a temporary register and then sent to various register locations on board the PCnet-PCI II controller. Access to the PCnet-PCI II controller configuration space, the Expansion ROM or any I/O resource is not possible during the EEPROM read operation. The PCnet-PCI II controller will terminate any access attempt with the assertion of $\overline{\text{DEVSEL}}$ and $\overline{\text{STOP}}$ while $\overline{\text{TRDY}}$ is not asserted, signaling to the initiator to disconnect and retry the access at a later time.

A checksum verification is performed on the data that is read from the EEPROM. If the checksum verification passes, PVALID (BCR19, bit 15) will be set to ONE. If the checksum verification of the EEPROM data fails, PVALID will be cleared to ZERO and the PCnet-PCI II controller will force all EEPROM-programmable BCR registers back to their H_RESET default values. The content of the Address PROM locations (offsets 0h–Fh from the I/O or memory mapped I/O base address), however, will not be cleared. The 8 bit checksum for the entire 36 bytes of the EEPROM should be FFh.

If no EEPROM is present at the time of the automatic read operation, the PCnet-PCI II controller will recognize this condition and will abort the automatic read operation and clear both the PREAD and PVALID bits in BCR19. All EEPROM-programmable BCR registers will be assigned their default values after H_RESET. The content of the Address PROM locations (offsets 0h–Fh from the I/O or memory mapped I/O base address) will be undefined.

If the user wishes to modify any of the configuration bits that are contained in the EEPROM, then the seven command, data and status bits of BCR19 can be used to write to the EEPROM. After writing to the EEPROM, the host should set the PREAD bit of BCR19. This action forces a PCnet-PCI II controller re-read of the EEPROM so that the new EEPROM contents will be loaded into the EEPROM-programmable registers on board the PCnet-PCI II controller. (The EEPROM-programmable registers may also be reprogrammed directly, but only information that is stored in the EEPROM will be preserved at system power-down.) When the PREAD bit of BCR19 is set, it will cause the PCnet-PCI II controller to ignore further accesses to the PCnet-PCI II controller configuration space, the Expansion ROM, or any I/O resource until the completion of the EEPROM read operation. The PCnet-PCI II controller will terminate these access attempts with the assertion of $\overline{\text{DEVSEL}}$ and $\overline{\text{STOP}}$ while $\overline{\text{TRDY}}$ is not asserted, signaling to the initiator to disconnect and retry the access at a later time.

EEPROM Auto-Detection

The PCnet-PCI II controller uses the EESK/$\overline{\text{LED1}}$/SFBD pin to determine if an EEPROM is present in the system. At the rising edge of CLK during the last clock during which $\overline{\text{RST}}$ is asserted, the PCnet-PCI II controller will sample the value of the EESK/$\overline{\text{LED1}}$/SFBD pin. If the sampled value is a ONE, then the PCnet-PCI II controller assumes that an EEPROM is present, and the EEPROM read operation begins shortly after the $\overline{\text{RST}}$ pin is deasserted. If the sampled value of EESK/$\overline{\text{LED1}}$/SFBD is a ZERO, the PCnet-PCI II controller assumes that an external pulldown device is holding the EESK/$\overline{\text{LED1}}$/SFBD pin low indicating that there is no EEPROM in the system. Note that if the designer creates a system that contains an LED circuit on the EESK/$\overline{\text{LED1}}$/SFBD pin but has no EEPROM present, then the EEPROM auto-detection function will incorrectly conclude that an EEPROM is present in the system. However, this will not pose a problem for the PCnet-PCI II controller, since the checksum verification will fail.

Direct Access to the Microwire Interface

The user may directly access the Microwire port through the EEPROM register, BCR19. This register contains bits that can be used to control the Microwire interface pins. By performing an appropriate sequence of accesses to BCR19, the user can effectively write to and read from the EEPROM. This feature may be used by a system configuration utility to program hardware configuration information into the EEPROM.

EEPROM-programmable Registers

The following registers contain configuration information that will be programmed automatically during the EEPROM read operation:

■ I/O offsets 0h–Fh	Address PROM locations
■ BCR2	Miscellaneous Configuration register
■ BCR4	Link Status LED register
■ BCR5	LED1 Status register
■ BCR6	LED2 Status register
■ BCR7	LED3 Status register
■ BCR9	Full-Duplex Control register
■ BCR18	Burst and Bus Control register
■ BCR22	PCI Latency register

If PREAD (BCR19, bit 14) and PVALID (BCR19, bit 15) are cleared to ZERO, then the EEPROM read has experienced a failure and the contents of the EEPROM programmable BCR register will be set to default H_RESET values. The content of the Address PROM locations, however, will not be cleared.

Note that accesses to the Address PROM I/O locations do not directly access the Address EEPROM itself. Instead, these accesses are routed to a set of shadow registers on board the PCnet-PCI II controller that are loaded with a copy of the EEPROM contents during the automatic read operation that immediately follows the H_RESET operation.

EEPROM MAP

The automatic EEPROM read operation will access 18 words (i.e. 36 bytes) of the EEPROM. The format of the EEPROM contents is shown below, beginning with the byte that resides at the lowest EEPROM address:

Table 11. EEPROM Content

Word Address	Byte Addr.	Most Significant Byte	Byte Addr.	Least Significant Byte
00h (Lowest EEPROM address)	01h	Second byte of the ISO 8802-3 (IEEE/ANSI 802.3) station physical address for this node	00h	First byte of the ISO 8802-3 (IEEE/ANSI 802.3) station physical address for this node, where first byte refers to the first byte to appear on the 802.3 medium
01h	03h	Fourth byte of the node address	02h	Third byte of the node address
02h	05h	Sixth byte of the node address	04h	Fifth byte of the node address
03h	07h	Reserved Location: must be 00h	06h	Reserved location must be 00h
04h	09h	Hardware ID: must be 11h if compatibility to AMD drivers is desired	08h	Reserved location must be 00h
05h	0Bh	User programmable space	0Ah	User programmable space
06h	0Dh	MSByte of two-byte checksum, which is the sum of bytes 00h–0Bh and bytes 0Eh and 0Fh	0Ch	LSByte of two-byte checksum, which is the sum of bytes 00h–0Bh and bytes 0Eh and 0Fh
07h	0Fh	Must be ASCII W (57h) if compatibility to AMD driver software is desired	0Eh	Must be ASCII W (57h) if compatibility to AMD driver software is desired
08h	11h	BCR4[15:8] (Link Status LED)	10h	BCR4[7:0] (Link Status LED)
09h	13h	BCR5[15:8] (LED1 Status)	12h	CR5[7:0] (LED1 Status)
0Ah	15h	BCR18[15:8] (Burst and Bus Control)	14h	BCR18[7:0] (Burst and Bus Control)
0Bh	17h	BCR2[15:8] (Miscellaneous Configuration)	16h	BCR2[7:0] (Miscellaneous Configuration)
0Ch	19h	BCR6[15:8] (LED2 Status)	18h	BCR6[7:0] (LED2 Status)
0Dh	1Bh	BCR7[15:8] (LED3 Status)	1Ah	BCR7[7:0] (LED3 Status)
0Eh	1Dh	BCR9[15:8] (Full-Duplex Control)	1Ch	BCR9[7:0] (Full-Duplex Control)
0Fh	1Fh	Checksum adjust byte for the first 36 bytes of the EEPROM contents, checksum of the first 36 bytes of the EEPROM should total to FFh	1Eh	Reserved location must be 00h
10h	21h	BCR22[15:8] (PCI Latency)	20h	BCR22[7:0] (PCI Latency)
11h	23h	Reserved location must be 00h	22h	Reserved location must be 00h

Note that the first bit out of any word location in the EEPROM is treated as the MSB of the register that is being programmed. For example, the first bit out of EEPROM word location 08h will be written into BCR4, bit 15, the second bit out of EEPROM word location 08h will be written into BCR4, bit 14, etc.

There are two checksum locations within the EEPROM. The first checksum will be used by AMD driver software to verify that the ISO 8802-3 (IEEE/ANSI 802.3) station address has not been corrupted. The value of bytes 0Ch and 0Dh should match the sum of bytes 00h through 0Bh and 0Eh and 0Fh. The second checksum location — byte 1 Fh — is not a checksum total, but is, instead, a checksum adjustment. The value of this byte should be such that the total checksum for the entire 36 bytes of EEPROM data equals the value FFh. The checksum adjust byte is needed by the PCnet-PCI II controller in order to verify that the EEPROM content has not been corrupted.

LED Support

The PCnet-PCI II controller can support up to four LEDs. LED outputs LNKST, LED1 and LED2 allow for direct connection of an LED and its supporting pullup device. LED output LED3 may require an additional buffer between the PCnet-PCI II controller output pin and the LED and its supporting pullup device.

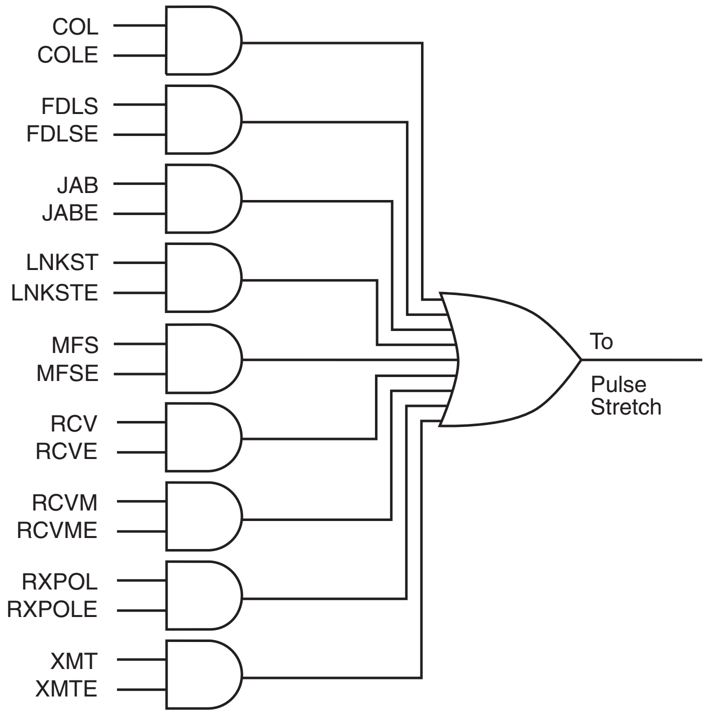
Because the LED3 output is multiplexed with other PCnet-PCI II controller functions, it may not always be possible to connect an LED circuit directly to the LED3 pin. In applications that want to use the pin to drive an LED and also have an EEPROM, it might be necessary to buffer the LED3 circuit from the EEPROM connection. When an LED circuit is directly connected to the EEDO/LED3/SRD pin, then it is not possible for most Microwire EEPROM devices to sink enough IOL to maintain a valid low level on the EEDO input to the PCnet-PCI II controller. In applications where an EEPROM is not needed, the LED3 pin may be directly connected to an LED circuit. The PCnet-PCI II controller LED3 pin driver will be able to sink enough current to properly drive the LED circuit.

Each LED can be programmed through a BCR register to indicate one or more of the following network status or activities: Collision Status, Full-Duplex Link Status, Half-Duplex Link Status, Jabber Status, Magic Packet Status, Receive Match, Receive Polarity, Receive Status and Transmit Status. The LED pins can be configured to operate in either open-drain mode (active low) or in totem-pole mode (active high). The output can be stretched to allow the human eye to recognize even short events that last only several microseconds. After H_RESET, the four LED outputs are configured in the following manner:

Table 12. LED Default Configuration

LED Output	Indication	Driver Mode	Pulse Stretch
LNKST	Link Status	Open Drain – Active Low	Enabled
LED1	Receive Status	Open Drain – Active Low	Enabled
LED2	Receive Polarity	Open Drain – Active Low	Enabled
LED3	Transmit Status	Open Drain – Active Low	Enabled

For each LED register, each of the status signals is ANDed with its enable signal, and these signals are all ORed together to form a combined status signal. Each LED pins combined status signal can be programmed to run to a pulse stretcher, which consists of a 3-bit shift register clocked at 38 Hz (26 ms). The data input of each shift register is normally at logic 0. The OR gate output for each LED register asynchronously sets all three bits of its shift register when the output becomes asserted. The inverted output of each shift register is used to control an LED pin. Thus the pulse stretcher provides 2–3 clocks of stretched LED output, or 52 ms to 78 ms.



19436C-44

Figure 41. LED Control Logic

Power Savings Modes

The PCnet-PCI II controller supports two hardware power savings modes. Both are entered by driving the SLEEP pin LOW.

The power down mode that yields the most power savings is called coma mode. In coma mode, the entire device is shut down. All inputs are ignored except the SLEEP pin itself. Coma mode is enabled when AWAKE (BCR2, bit 2) is at its default value of ZERO and SLEEP is asserted.

The second power saving mode is called snooze mode. In snooze mode, enabled by setting AWAKE to ONE and driving the SLEEP pin LOW, the T-MAU receive circuitry will remain active even while the SLEEP pin is driven LOW. The LNKST output is the only one of the LED pins that continues to function. All other sections of the device are shut down. The LNKSTE bit must be set in BCR4 to enable indication of a good 10BASE-T link if there are link beat pulses or valid frames present. This LNKST pin can be used to drive an LED and/or external hardware that directly controls the SLEEP pin of the PCnet-PCI II controller. This configuration effectively wakes the system when there is any activity on the 10BASE-T link. Snooze mode can be used only if the T-MAU is the selected network port. Link beat pulses are not transmitted during snooze mode.

The SLEEP pin must not be asserted while the PCnet-PCI II controller is requesting the bus or while a slave or bus master cycle is in progress. A recommended method is to set the PCnet-PCI II controller into ***suspend*** mode by setting the SPND bit in CSR5 to ONE prior to asserting the SLEEP pin. Another recommended method is to ***stop*** the device by setting the STOP bit in CSR0 to ONE prior to asserting the SLEEP pin.

Before the sleep mode is invoked, the PCnet-PCI II controller will perform an internal S_RESET. This S_RESET operation will not affect the values of the BCR registers or the PCI configuration space. S_RESET terminates all network activity abruptly. The host can use the suspend mode (SPND, CSR5, bit 0) to terminate all network activity in an orderly sequence before issuing an S_RESET.

When coming out of the sleep mode, the PCnet-PCI II controller can be programmed to generate an interrupt and inform the driver about the wake-up. The PCnet-PCI II controller will set SLPINT (CSR5, bit 9), when coming out of the sleep mode. INTA will be asserted, when the enable bit SLPINTE (CSR5, bit 8) is set to ONE. Note that the assertion of INTA due to SLPINT is not dependent on the main interrupt enable bit INEA (CSR0, bit 6), which will be cleared by the reset going into the sleep mode.

The $\overline{\text{SLEEP}}$ pin should not be asserted during power supply ramp-up. If it is desired that $\overline{\text{SLEEP}}$ be asserted at power up time, then the system must delay the assertion of $\overline{\text{SLEEP}}$ until three clock cycles after the completion of a hardware reset 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 not tested. The T-MAU drivers TXD±, TXP± and receiver RXD± are also not tested.

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

Boundary Scan Circuit

The boundary scan test circuit requires four 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 boundary scan circuit remains active during Sleep mode.

TAP Finite State Machine

The TAP engine is a 16-state finite state machine (FSM), driven by the Test Clock (TCK) and the Test Mode Select (TMS) pins. An independent power-on reset circuit is provided to ensure the FSM is in the TEST_LOGIC_RESET state at power-up. The FSM is also reset when TMS and TDI are high for five TCK periods.

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 following table for a summary of supported instructions.

Table 13. IEEE 1149.1 Supported Instruction Summary

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

Instruction Register and Decoding Logic

After the TAP FSM is 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

Each Boundary Scan Register (BSR) cell has two stages. A flip-flop and a latch are used for the Serial Shift Stage and the Parallel Output Stage, respectively.

There are four possible operation modes in the BSR cell:

Table 14. Boundry Scan Register Mode Of Operation

1	Capture
2	Shift
3	Update
4	System Function

Other Data Registers

1. Bypass Register (1 bit)
2. Device ID register (32 bits)

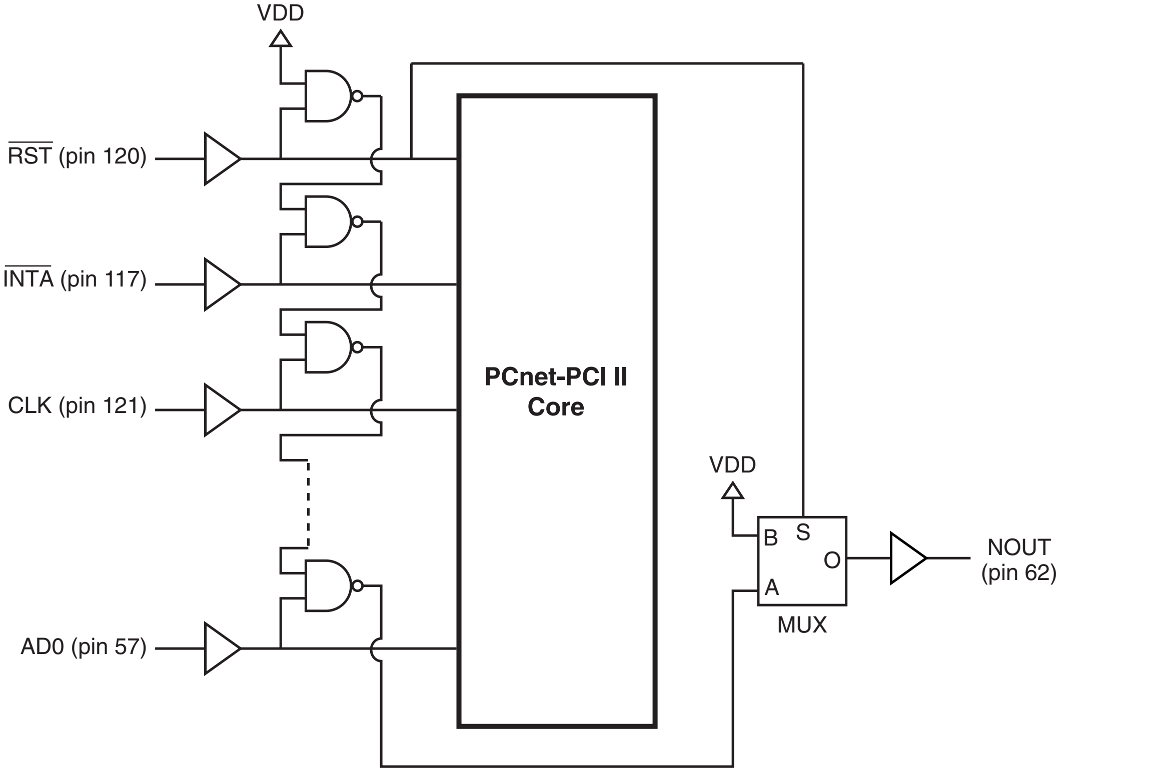
Table 15. Device ID Register

Bits 31–28	Version Bits 27–12 Part Number (0010 0100 0011 XXXX)
Bits 11–1	Manufacturer ID. The 11 bit manufacturer ID cod for AMD is 00000000001 in accordance with JEDEC publication 106-A.
Bit 0	Always a logic 1

Note that the content of the Device ID register is the same as the content of CSR88.

NAND Tree Testing

The PCnet-PCI II controller provides a NAND tree test mode to allow checking connectivity to the device on a printed circuit board. The NAND tree is built on all PCI bus signals. NAND tree testing is enabled by asserting $\overline{\text{RST}}$. All PCI bus signals will become inputs on the assertion of $\overline{\text{RST}}$. The result of the NAND tree test can be observed on the NOUT pin.



19436C-45

Figure 42. NAND Tree Circuitry

Pin 120 ($\overline{\text{RST}}$) is the first input to the NAND tree. Pin 117 ($\overline{\text{INTA}}$) is the second input to the NAND tree, followed by pin 121 (CLK). All other PCI bus signals follow, counter clockwise, with pin 57 (AD0) being the last. Pins labeled NC and all power supply pins are not part of the NAND tree. The table below shows the complete list of pins connected to the NAND tree.

Table 16. NAND Tree Pin Sequence

NAND Tree Input No.	Pin No.	Name	NAND Tree Input No.	Pin No.	Name	NAND Tree Input No.	Pin No.	Name
1	120	$\overline{\text{RST}}$	18	15	AD21	35	36	AD15
2	117	$\overline{\text{INTA}}$	19	16	AD20	36	38	AD14
3	121	CLK	20	18	AD19	37	39	AD13
4	123	$\overline{\text{GNT}}$	21	19	AD18	38	40	AD12
5	126	$\overline{\text{REQ}}$	22	21	AD17	39	41	AD11
6	128	AD31	23	22	AD16	40	42	AD10
7	129	AD30	24	23	C/$\overline{\text{BE}}$2	41	44	AD9
8	131	AD29	25	24	$\overline{\text{FRAME}}$	42	45	AD8
9	132	AD28	26	25	$\overline{\text{IRDY}}$	43	47	C/$\overline{\text{BE}}$0
10	2	AD27	27	26	$\overline{\text{TRDY}}$	44	48	AD7
11	3	AD26	28	27	$\overline{\text{DEVSEL}}$	45	49	AD6
12	5	AD25	29	28	$\overline{\text{STOP}}$	46	51	AD5
13	6	AD24	30	29	$\overline{\text{LOCK}}$	47	52	AD4
14	7	C/$\overline{\text{BE}}$3	31	31	$\overline{\text{PERR}}$	48	53	AD3
15	10	IDSEL	32	32	$\overline{\text{SERR}}$	49	54	AD2
16	12	AD23	33	34	PAR	50	56	AD1
17	13	AD22	34	35	C/$\overline{\text{BE}}$1	51	57	AD0

$\overline{\text{RST}}$ must be asserted low to start a NAND tree test sequence. Initially, all NAND tree inputs except $\overline{\text{RST}}$ should be driven high. This will result in a high output at the NOUT pin. If the NAND tree inputs are driven from high to low in the same order as they are connected to build the NAND tree, NOUT will toggle every time an additional input is driven low. NOUT will change to low, when INTA is driven low and all other NAND tree inputs stay high. NOUT will toggle back to high, when CLK is additionally driven low. The square wave will continue until all NAND tree inputs are driven low. NOUT will be high, when all NAND tree inputs are driven low.

Note, that some of the pins connected to the NAND tree are outputs in normal mode of operation. They must not be driven from an external source until the PCnet-PCI II controller is configured for NAND tree testing.

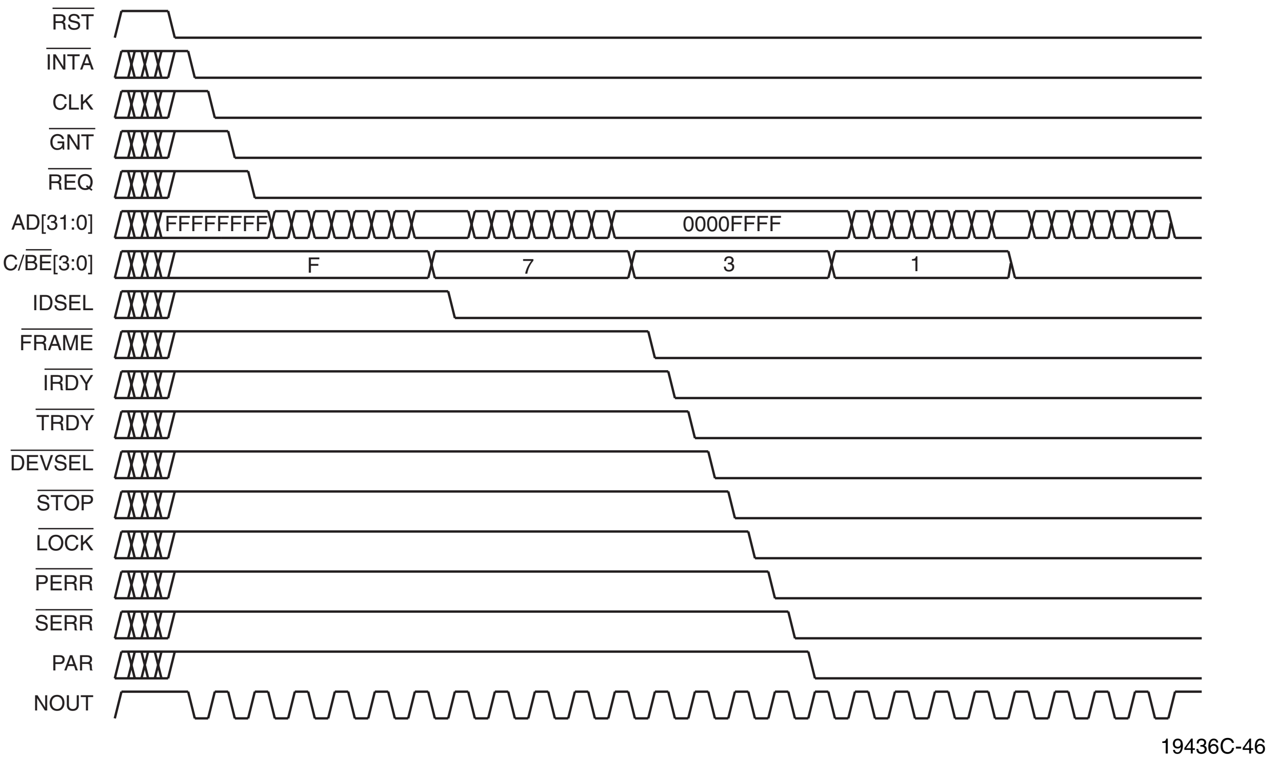


Figure 43. NAND Tree Waveform

Reset

There are three different types of RESET operations that may be performed on the PCnet-PCI II controller device, H_RESET, S_RESET and STOP. These names have been used throughout the document. The following is a description of each type of RESET operation.

H_RESET

Hardware Reset (H_RESET) is a PCnet-PCI II controller reset operation that has been created by the proper assertion of the $\overline{\text{RST}}$ pin of the PCnet-PCI II controller device. When the minimum pulse width timing as specified in the $\overline{\text{RST}}$ pin description has been satisfied, then an internal reset operation will be performed.

H_RESET will program most of the CSR and BCR registers to their default value. Note that there are several CSR and BCR registers that are undefined after H_RESET. See the sections on the individual registers for details. H_RESET will clear all registers in the PCI configuration space. H_RESET will cause the microcode program to jump to its reset state. Following the end of the H_RESET operation, the PCnet-PCI II controller will attempt to read the EEPROM device through the EEPROM Microwire interface. H_RESET resets the T-MAU into the Link Fail state.

S_RESET

Software Reset (S_RESET) is a PCnet-PCI II controller reset operation that has been created by a read access to the Reset register which is located at offset 14h in Word I/O mode or offset 18h in DWord I/O mode from the PCnet-PCI II controller I/O or memory mapped I/O base address.

S_RESET will reset all of or some portions of CSR0, 3, 4, 15, 80, 100 and 124 to default values. For the identity of individual CSRs and bit locations that are affected by S_RESET, see the individual CSR register descriptions. S_RESET will not affect any PCI configuration space locations. With the exception of DWIO (BCR18, bit 7) S_RESET will not affect any of the BCR register values. S_RESET will cause the microcode program to jump to its reset state. Following the end of the S_RESET operation, the PCnet-PCI II controller will not attempt to read the EEPROM device. S_RESET does not affect the status of the T-MAU. After S_RESET, the host must perform a full re-initialization of the PCnet-PCI II controller before starting network activity.

S_RESET will clear DWIO (BCR18, bit 7) and the PCnet-PCI II controller will be in 16-bit I/O mode after the reset operation. A DWord write operation to the RDP (I/O offset 10h) must be performed to set the device into 32-bit I/O mode.

S_RESET will cause $\overline{\text{REQ}}$ to deassert immediately. STOP (CSR0, bit 2) or SPND (CSR5, bit 0) can be used to terminate any pending bus mastership request in an orderly sequence.

S_RESET terminates all network activity abruptly. The host can use the suspend mode (SPND, CSR5, bit 0) to terminate all network activity in an orderly sequence before issuing an S_RESET.

STOP

A STOP reset is generated by the assertion of the STOP bit in CSR0. Writing a ONE to the STOP bit of CSR0, when the stop bit currently has a value of ZERO, will initiate a STOP reset. If the STOP bit is already a ONE, then writing a ONE to the STOP bit will not generate a STOP reset.

STOP will reset all or some portions of CSR0, 3, and 4 to default values. For the identity of individual CSRs and bit locations that are affected by STOP, see the individual CSR register descriptions. STOP will not affect any of the BCR and PCI configuration space locations. STOP will cause the microcode program to jump to its reset state. Following the end of the STOP operation, the PCnet-PCI II controller will not attempt to read the EEPROM device. Setting the STOP bit does not affect the T-MAU.

Note that STOP will not cause a deassertion of the $\overline{\text{REQ}}$ signal, if it happens to be active at the time of the write to CSR0. The PCnet-PCI II controller will wait until it gains bus ownership and it will first finish all scheduled bus master accesses before the STOP reset is executed.

STOP terminates all network activity abruptly. The host can use the suspend mode (SPND, CSR5, bit 0) to terminate all network activity in an orderly sequence before setting the STOP bit.

Software Access

PCI Configuration Registers

The PCnet-PCI II controller implements a 256-byte configuration space as defined by the PCI specification revision 2.0. The 64-byte header includes all registers required to identify the PCnet-PCI II controller and its function. Additional registers are used to setup the configuration of the PCnet-PCI II controller in a system. None of the device specific registers located at offsets 40h through FCh are implemented. The layout of the PCnet-PCI II controller PCI configuration space is shown in the table below.

The PCI configuration registers are accessible only by configuration cycles. All multi-byte numeric fields follow little endian byte ordering. All write accesses to Reserved locations have no effect; reads from these locations will return a data value of ZERO.

Table 17. PCI Configuration Space Layout

31 24	**23 16**	**15 8**	**7 0**	**Offset**
Device ID		Vendor ID		00h
Status		Command		04h
Base-Class	Sub-Class	Programming IF	Revision ID	08h
Reserved	Header Type	Latency Timer	Reserved	0Ch
I/O Base Address				10h
Memory Mapped I/O Base Address				14h
Reserved				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Reserved				2Ch
Expansion ROM Base Address				30h
Reserved				34h
Reserved				38h
MAX_LAT	MIN_GNT	Interrupt Pin	Interrupt Line	3Ch
Reserved				40h
Reserved				:
Reserved				FCh

I/O Resources

The PCnet-PCI II controller requires 32 bytes of address space for access to all the various internal registers as well as to some setup information stored in an external serial EEPROM. A software reset port is available, too.

The PCnet-PCI II controller supports mapping the address space to both I/O and memory space. The value in the PCI I/O Base Address register determines the start address of the I/O address space. The register is typically programmed by the PCI configuration utility after system power-up. The PCI configuration utility must also set the IOEN bit in the PCI Command register to enable I/O accesses to the PCnet-PCI II controller. For memory mapped I/O access, the PCI Memory Mapped I/O Base Address register controls the start address of the memory space. The MEMEN bit in the PCI Command register must also be set to enable the mode. Both base address registers can be active at the same time.

The PCnet-PCI II controller supports two modes for accessing the I/O resources. For backwards compatibility with AMD's 16-bit Ethernet controllers, Word I/O is the default mode after power up. The device can be configured to DWord I/O mode by software.

I/O Registers

The PCnet-PCI II controller registers are divided into two groups. The Control and Status Registers (CSR) are used to configure the Ethernet MAC engine and to obtain status information. The Bus Control Registers (BCR) are use to configure the bus interface unit and the LEDs. Both sets of registers are accessed using indirect addressing.

The CSR and BCR share a common Register Address Port (RAP). There are, however, separate data ports. The Register Data Port (RDP) is used to access a CSR. The BCR Data Port (BDP) is used to access a BCR.

In order to access a particular CSR location, the RAP should first be written with the appropriate CSR address. The RDP will then points to the selected CSR. A read of the RDP will yield the selected CSR data. A write to the RDP will write to the selected CSR. In order to access a particular BCR location, the RAP should first be written with the appropriate BCR address. The BDP will then points to the selected BCR. A read of the BDP will yield the selected BCR data. A write to the BDP will write to the selected BCR.

Once the RAP has been written with a value, the RAP value remains unchanged until another RAP write occurs, or until an H_RESET or S_RESET occurs. RAP is cleared to all ZEROs when an H_RESET or S_RESET occurs. RAP is unaffected by setting the STOP bit.

Address PROM Space

The PCnet-PCI II controller allows for connection of a serial EEPROM. The first 16 bytes of the EEPROM will be automatically loaded into the Address PROM (APROM) space after H_RESET. The Address PROM space is a convenient place to store the value of the 48-bit IEEE station address. It can be overwritten by the host computer. Its content has no effect on the operation of the controller. The software must copy the station address from the Address PROM space to the initialization block or to CSR12-14 in order for the receiver to accept unicast frames directed to this station.

The 6 bytes of IEEE station address occupy the first 6 locations of the Address PROM space. The next six bytes are reserved. Bytes 12 and 13 should match the value of the checksum of bytes 1 through 11 and 14 and 15. Bytes 14 and 15 should each be ASCII W (57h). The above requirements must be met in order to be compatible with AMD driver software.

The APROMWE bit (BCR2, bit 8) must be set to ONE to enable write access to the Address PROM space.

Reset Register

A read of the Reset register creates an internal software reset (S_RESET) pulse in the PCnet-PCI II controller. The internal S_RESET pulse that is generated by this access is different from both the assertion of the hardware $\overline{\text{RST}}$ pin (H_RESET) and from the assertion of the software STOP bit. Specifically, S_RESET is the equivalent of the assertion of the $\overline{\text{RST}}$ pin (H_RESET) except that S_RESET has no effect on the BCR or PCI Configuration space locations or on the T-MAU.

The NE2100 LANCE based family of Ethernet cards requires that a write access to the Reset register follows each read access to the Reset register. The PCnet-PCI II controller does not have a similar requirement. The write access is not required but it does not have any effects.

Note that the PCnet-PCI II controller cannot service any slave accesses for a very short time after a read access of the Reset register, because the internal S_RESET operation takes about 1 µs to finish. The PCnet-PCI II controller will terminate all slave accesses with the assertion of $\overline{\text{DEVSEL}}$ and $\overline{\text{STOP}}$ while $\overline{\text{TRDY}}$ is not asserted, signaling to the initiator to disconnect and retry the access at a later time.

Word I/O Mode

After H_RESET, the PCnet-PCI II controller is programmed to operate in Word I/O mode. DWIO (BCR18, bit 7) will be cleared to ZERO. The table below shows how the 32 bytes of address space are used in Word I/O mode.

Table 18. I/O Map In Word I/O Mode (DWIO = 0)

Offset	No. of Bytes	Register
00h -0Fh	16	APROM
10h	2	RDP
12h	2	RAP (shared by RDP and BDP)
14h	2	Reset Register
16h	2	BDP
18h -1Fh	8	Reserved

All I/O resources must be accessed in word quantities and on word addresses. The Address PROM locations can also be read in byte quantities. The only allowed DWord operation is a write access to the RDP, which switches the device to DWord I/O mode. A read access other than listed in the table below will yield undefined data, a write operation may cause unexpected reprogramming of the PCnet-PCI II controller control registers.

Table 19. Legal I/O Accesses in Word I/O Mode (DWIO = 0)

AD[4:0]	$\overline{BE}$[3:0]	Type	Comment
0XX00	1110	RD	Byte Read of APROM Location 0h, 4h, 8h or Ch
0XX01	1101	RD	Byte Read of APROM Location 1h, 5h, 9h or Dh
0XX10	1011	RD	Byte Read of APROM Location 2h, 6h, Ah or Eh
0XX11	0111	RD	Byte Read of APROM Location 3h, 7h, Bh or Fh
0XX00	1100	RD	Word Read of APROM Locations 1h (MSB) and 0h (LSB), 5h and 4h, 8h and 9h or Ch and Dh
0XX10	0011	RD	Word Read of APROM Location 3h (MSB) and 2h (LSB), 7h and 6h, Bh and Ah or Fh or Eh
10000	1100	RD	Word Read of RDP
10010	0011	RD	Word Read of RAP
10100	1100	RD	Word Read of Reset Register
10110	0011	RD	Word Read of BDP
0XX00	1100	WR	Word Write to APROM Locations 1h (MSB) and 0h (LSB), 5h and 4h, 8h and 9h or Ch and Dh
0XX10	0011	WR	Word Write to APROM Locations 3h (MSB) and 2h (LSB), 7h and 6h, Bh and Ah or Fh and Eh
10000	1100	WR	Word Write to RDP
10010	0011	WR	Word Write to RAP
10100	1100	WR	Word Write to Reset Register
10110	0011	WR	Word Write to BDP
10000	0000	WR	DWord Write to RDP, switches Device to DWord I/O Mode

Double Word I/O Mode

The PCnet-PCI II controller can be configured to operate in DWord (32bit) I/O mode. The software can invoke the DWIO mode by performing a DWord write access to the I/O location at offset 10h (RDP). The data of the write access must be such that it does not affect the intended operation of the PCnet-PCI II controller. Setting the device into 32-bit I/O mode is usually the first operation after H_RESET. The RAP register will point to CSR0 at that time. Writing a value of ZERO to CSR0 is a save operation. DWIO (BCR18, bit 7) will be set to ONE as indicating that the PCnet-PCI II controller operates in 32-bit I/O mode.

Note that even though the I/O resource mapping changes when the I/O mode setting changes, the RDP location offset is the same for both modes. Once the DWIO bit has been set to ONE, only H_RESET or

S_RESET can clear it to ZERO. The DWIO mode setting is unaffected by setting the STOP bit.

The table below shows how the 32 bytes of address space are used in DWord I/O mode.

Table 20. I/O Map In DWord I/O Mode (DWIO = 1)

Offset	No. of Bytes	Register
00h – 0Fh	16	APROM
10h	4	RDP
14h	4	RAP (shared by RDP and BDP)
18h	4	Reset Register
1Ch	4	BDP

All I/O resources must be accessed in DWord quantities and on DWord addresses. A read access other than listed in the table below will yield undefined data, a write operation may cause unexpected reprogramming of the PCnet-PCI II controller control registers.

Table 21. Legal I/O Accesses in Double Word I/O Mode (DWIO = 1)

AD[4:0]	$\overline{\text{BE}}$[3:0]	Type	Comment
0XX00	0000	RD	DWord Read of APROM Locations 3h (MSB) to 0h (LSB), 7h to 4h, Bh to 8h or Fh to Ch
10000	0000	RD	DWord Read of RDP
10100	0000	RD	DWord Read of RAP
11000	0000	RD	DWord Read of Reset Register
0XX00	0000	WR	DWord Write to APROM Locations 3h (MSB) to 0h (LSB), 7h to 4h, Bh to 8h or Fh to Ch
10000	0000	WR	DWord Write to RDP
10100	0000	WR	DWord Write to RAP
11000	0000	WR	DWord Write to Reset Register

USER ACCESSIBLE REGISTERS

The PCnet-PCI II controller has three types of user registers: the PCI configuration registers, the Control and Status registers (CSR) and the Bus Control registers (BCR).

The PCnet-PCI II controller implements all PCnet-ISA (Am79C960) registers, all C-LANCE (Am79C90) registers, all ILACC (Am79C900) registers, plus a number of additional registers. The PCnet-PCI II controller CSRs are compatible with both the PCnet-ISA CSRs and all of the C-LANCE CSRs upon power up. Compatibility to the ILACC set of CSRs requires one access to the Software Style register (BCR20, bits 7–0) to be performed. By setting an appropriate value of the Software Style register (BCR20, bits 7–0) the user can select a set of CSRs that are compatible with the ILACC set of CSRs.

The PCI configuration registers can be accessed in any data width. All other registers must be accessed according to the I/O mode that is currently selected. When WIO mode is selected, all other register locations are defined to be 16 bits in width. When DWIO mode is selected, all these register locations are defined to be 32 bits in width, with the upper 16 bits of most register locations marked as reserved locations with undefined values. When performing register write operations in DWIO mode, the upper 16 bits should always be written as ZEROs. When performing register read operations in DWIO mode, the upper 16 bits of I/O resources should always be regarded as having undefined values, except for CSR88.

PCnet-PCI II controller registers can be divided into four groups:

PCI Configuration Registers

Registers that are intended to be initialized by the system initialization procedure (e.g. BIOS device initialization routine) to program the operation of the PCnet-PCI II controller PCI bus interface.

Setup Registers

Registers that are intended to be initialized by the device driver to program the operation of various PCnet-PCI II controller features.

Running Registers

Registers that are intended to be used by the device driver software once the PCnet-PCI II controller is running to access status information and to pass control information.

Test Registers

Registers that are intended to be used only for testing and diagnostic purposes.

Below is a list of the registers that fall into each of the first three categories. Those registers that are not included in either of these lists can be assumed to be intended for diagnostic purposes.

PCI Configuration Registers

The following is a list of those registers that would typically need to be programmed once during the initialization of the PCnet-PCI II controller within a system:

- PCI I/O Base Address or Memory Mapped I/O Base Address register
- PCI Expansion ROM Base Address register
- PCI Interrupt Line register
- PCI Latency Timer register
- PCI Status register
- PCI Command register

Setup Registers

The following is a list of those registers that would typically need to be programmed once during the setup of the PCnet-PCI II controller within a system. The control bits in each of these registers typically do not need to be modified once they have been written. However, there are no restrictions as to how many times these registers may actually be accessed. Note that if the default power up values of any of these registers is acceptable to the application, then such registers need never be accessed at all. Registers marked with “^” may be programmable through the EEPROM read operation, and therefore do not necessarily need to be written to by the system initialization procedure or by the driver software. Registers marked with “*” will be initialized by the initialization block read operation.

CSR1	Initialization Block Address[15:0]
CSR2	Initialization Block Address[31:16]
CSR3	Interrupt Masks and Deferral Control
CSR4	Test and Features Control
CSR5	Extended Control and Interrupt
CSR8*	Logical Address Filter[15:0]
CSR9*	Logical Address Filter[31:16]
CSR10*	Logical Address Filter[47:32]
CSR11*	Logical Address Filter[63:48]
CSR12*	Physical Address[15:0]
CSR13*	Physical Address[31:16]
CSR14*	Physical Address[47:32]
CSR15*	Mode
CSR24*	Base Address of Receive Descriptor Ring Lower
CSR25*	Base Address of Receive Descriptor Ring Upper
CSR30*	Base Address of Transmit Descriptor Ring Lower
CSR31*	Base Address of Transmit Descriptor Ring Upper
CSR47	Polling Interval
CSR76*	Receive Descriptor Ring Length
CSR78*	Transmit Descriptor Ring Length
CSR82	Bus Activity Timer
CSR100	Memory Error Timeout
CSR122	Receiver Packet Alignment Control
BCR2^	Miscellaneous Configuration
BCR4^	Link Status LED
BCR5^	LED1 Status
BCR6^	LED2 Status
BCR7^	LED3 Status
BCR9^	Full-Duplex Control
BCR18^	Bus and Burst Control
BCR20	Software Style

Running Registers

The following is a list of those registers that would typically need to be periodically read and perhaps written during the normal running operation of the PCnet-PCI II controller within a system. Each of these registers contains control bits or status bits or both.

RAP	Register Address Port
CSR0	PCnet-PCI II Controller Status
CSR4	Test and Features Control
CSR5	Extended Control and Interrupt
CSR112	Missed Frame Count
CSR114	Receive Collision Count
PCI Status register	

PCI Configuration Registers

PCI Vendor ID (Offset 00h)

The PCI Vendor ID register is a 16-bit register that identifies the manufacturer of the PCnet-PCI II controller. Advanced Micro Devices, Inc.’s (AMD) Vendor ID is 1022h. Note that this vendor ID is not the same as the Manufacturer ID in CSR88andCSR89.The vendor ID is assigned by the PCI Special Interest Group.

The PCI Vendor ID register is located at offset 00h in the PCI Configuration Space. It is read only.

PCI Device ID Register (Offset 02h)

The PCI Device ID register is a 16-bit register that uniquely identifies the PCnet-PCI II controller within AMD's product line. The PCnet-PCI II controller Device ID is 2000h. Note that this Device ID is not the same as the Part number in CSR88 and CSR89.The Device ID is assigned by Advanced Micro Devices, Inc.

The PCI Device ID register is located at offset 02h in the PCI Configuration Space. It is read only.

PCI Command Register (Offset 04h)

The PCI Command register is a 16-bit register used to control the gross functionality of the PCnet-PCI II controller. It controls the PCnet-PCI II controller's ability to generate and respond to PCI bus cycles. To logically disconnect the PCnet-PCI II controller device from all PCI bus cycles except Configuration cycles, a value of ZERO should be written to this register.

The PCI Command register is located at offset 04h in the PCI Configuration Space. It is read and written by the host.

Bit	Name	Description
15–10	RES	Reserved locations. Read as ZEROs, write operations have no effect.
9	FBTBEN	Fast Back-to-Back Enable. Read as ZERO, write operations have no effect. The PCnet-PCI II controller will not generate Fast Back-to-Back cycles.
8	SERREN	SERR enable. Controls the assertion of the $\overline{\text{SERR}}$ pin. $\overline{\text{SERR}}$ is disabled when SERREN is cleared. $\overline{\text{SERR}}$ will be asserted on detection of an address parity error and if both SERREN and PERREN (bit 6 of this register) are set. SERREN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.
7	ADSTEP	Address/data stepping. Read as ZERO, write operations have no effect. The PCnet-PCI II controller does not use address stepping.
6	PERREN	Parity Error Response enable. Enables the parity error response functions. When PERREN is "0" and the PCnetPCI II controller detects a parity error, it only sets the Detected Parity Error bit in the PCI Status register. When PERREN is "1", the PCnet-PCI II controller asserts $\overline{\text{PERR}}$ on the detection of a data parity error. It also sets the DATAPERR bit (PCI Status register, bit 8), when the data parity error occurred during a master cycle. PERREN also enables reporting address parity errors through the $\overline{\text{SERR}}$ pin and the $\overline{\text{SERR}}$ bit in the PCI Status register. PERREN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.
5	VGASNOOP	VGA palette snoop. Read as ZERO, write operations have no effect.
4	MWIEN	Memory Write and Invalidate Cycle enable. Read as ZERO, write operations have no effect. The PCnet-PCI II controller only generates Memory Write cycles.
3	SCYCEN	Special Cycle enable. Read as ZERO, write operations have no effect. The PCnet-PCI II controller ignores all Special Cycle operations.
2	BMEN	Bus Master enable. Setting BMEN enables the PCnet-PCI II controller to become a bus master on the PCI bus. The host must set BMEN before setting the INIT or STRT bit in CSR0 of the PCnet-PCI II controller. BMEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.
1	MEMEN	Memory Space access enable. The PCnet-PCI II controller will ignore all memory accesses when MEMEN is cleared. The host must set MEMEN before the first memory access to the device. For memory mapped I/O, the host must program the PCI Memory Mapped I/O Base Address register with a valid memory address before setting MEMEN. For accesses to the Expansion ROM, the host must program the

		PCI Expansion ROM Base Address register at offset 30h with a valid memory address before setting MEMEN. The PCnet-PCI II controller will only respond to accesses to the Expansion ROM when both ROMEN (PCI Expansion ROM Base Address register, bit 0) and MEMEN are set to ONE. Since MEMEN also enables the memory mapped access to the PCnet-PCI II controller I/O resources, the PCI Memory Mapped I/O Base Address register must be programmed with an address so that the device does not claim cycles not intended for it. MEMEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.
0	IOEN	I/O Space access enable. The PCnet-PCI II controller will ignore all I/O accesses when IOEN is cleared. The host must set IOEN before the first I/O access to the device. The PCI I/O Base Address register must be programmed with a valid I/O address before setting IOEN. IOEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.

PCI Status Register (Offset 06h)

The PCI Status register is a 16-bit register that contains status information for the PCI bus related events. It is located at offset 06h in the PCI Configuration Space.

Bit	Name	Description
15	PERR	Parity Error. PERR is set when the PCnet-PCI II controller detects a parity error. The PCnet-PCI II controller samples the AD[31:0], C/$\overline{\text{BE}}$[3:0] and the PAR lines for a parity error at the following times: In slave mode, during the address phase of any PCI bus command. In slave mode, for all I/O, memory and configuration write commands that select the PCnet-PCI II controller when data is transferred ($\overline{\text{TRDY}}$ and $\overline{\text{IRDY}}$ are asserted). In master mode, during the data phase of all memory read commands. In master mode, during the data phase of the memory write command, the PCnet-PCI II controller sets the PERR bit if the target reports a data parity error by asserting the $\overline{\text{PERR}}$ signal. PERR is not effected by the state of the Parity Error Response enable bit (PCI Command register, bit 6). PERR is set by the PCnet-PCI II controller and cleared by writing a ONE. Writing a ZERO has no effect. PERR is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
14	SERR	Signaled SERR. SERR is set when the PCnet-PCI II controller detects an address parity error, and both SERREN and PERREN (PCI Command register, bits 8 and 6) are set. SERR is set by the PCnet-PCI II controller and cleared by writing a ONE. Writing a ZERO has no effect. SERR is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
13	RMABORT	Received Master Abort. RMABORT is set when the PCnet-PCI II controller terminates a master cycle with a master abort sequence. RMABORT is set by the PCnet-PCI II controller and cleared by writing a ONE. Writing a ZERO has no effect. RMABORT is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
12	RTABORT	Received Target Abort. RTABORT is set when a target terminates a PCnet-PCI II controller master cycle with a target abort sequence. RTABORT is set by the PCnet-PCI II controller and cleared

by writing a ONE. Writing a ZERO has no effect. RTABORT is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

11 STABORT Send Target Abort. Read as ZERO, write operations have no effect. The PCnet-PCI II controller will never terminate a slave access with a target abort sequence.

STABORT is read only.

10–9 DEVSEL Device Select timing. DEVSEL is set to 01b (medium), which means that the PCnet-PCI II controller will assert $\overline{\text{DEVSEL}}$ two clock periods after $\overline{\text{FRAME}}$ is asserted.

DEVSEL is read only.

8 DATAPERR Data Parity Error detected. DATAPERR is set when the PCnet-PCI II controller is the current bus master and it detects a data parity error and the Parity Error Response enable bit (PCI Command register, bit 6) is set.

During the data phase of all memory read commands, the PCnet-PCI II controller checks for parity error by sampling the AD[31:0] and C/$\overline{\text{BE}}$[3:0] and the PAR lines. During the data phase of all memory write commands, the PCnet-PCI II controller checks the $\overline{\text{PERR}}$ input to detect whether the target has reported a parity error.

DATAPERR is set by the PCnet-PCI II controller and cleared by writing a ONE. Writing a ZERO has no effect. DATAPERR is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

7 FBTBC Fast Back-To-Back Capable. Read as ONE, write operations have no effect. The PCnet-PCI II controller is capable of accepting fast back-to-back transactions with the first transaction addressing a different target.

6–0 RES Reserved locations. Read as ZERO, write operations have no effect.

PCI Revision ID Register (Offset 08h)

The PCI Revision ID register is an 8-bit register that specifies the PCnet-PCI II controller revision number. The value of this register is 1xh, with the lower four bits being silicon-revision dependent.

The PCI Revision ID register is located at offset 08h in the PCI Configuration Space. It is read only.

PCI Programming Interface Register (Offset 09h)

The PCI Programming Interface register is an 8-bit register that identifies the programming interface of PCnet-PCI II controller. PCI does not define any specific register-level programming interfaces for network devices. The value of this register is 00h.

The PCI Programming Interface register is located at offset 09h in the PCI Configuration Space. It is read only.

PCI Sub-Class Register (Offset 0Ah)

The PCI Sub-Class register is an 8-bit register that identifies specifically the function of the PCnet-PCI II controller. The value of this register is 00h which identifies the PCnet-PCI II controller device as an Ethernet controller.

The PCI Sub-Class register is located at offset 0Ah in the PCI Configuration Space. It is read only.

PCI Base-Class Register (Offset 0Bh)

The PCI Base-Class register is an 8-bit register that broadly classifies the function of the PCnet-PCI II controller. The value of this register is 02h which classifies the PCnet-PCI II controller device as a network controller.

The PCI Base-Class register is located at offset 0Bh in the PCI Configuration Space. It is read only.

PCI Latency Timer Register (Offset 0Dh)

The PCI Latency Timer register is an 8-bit register that specifies the minimum guaranteed time the PCnet-PCI II controller will control the bus once it starts its bus mastership period. The time is measured in clock cycles. Every time the PCnet-PCI II controller asserts $\overline{\text{FRAME}}$ at the beginning of a bus mastership period, it will copy the value of the PCI Latency Timer register into a counter and start counting down. The counter will freeze at ZERO.When the system arbiter removes $\overline{\text{GNT}}$ while the counter is non-ZERO, the PCnet-PCI II controller will continue with its data transfers. It will only release the bus when the counter has reached ZERO.

The PCI Latency Timer is only significant in burst transactions, where $\overline{\text{FRAME}}$ stays asserted until the last data phase. In a non-burst transaction, $\overline{\text{FRAME}}$ is only asserted during the address phase. The internal latency counter will be cleared and suspended while $\overline{\text{FRAME}}$ is deasserted.

All 8 bits of the PCI Latency Timer register are programmable. The host should read the PCnet-PCI II controller PCI MIN_GNT and PCI MAX_LAT registers to determine the latency requirements for the device and then initialize the Latency Timer register with an appropriate value.

The PCI Latency Timer register is located at offset 0Dh in the PCI Configuration Space. It is read and written by the host. The PCI Latency Timer register is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.

PCI Header Type Register (Offset 0Eh)

The PCI Header Type register is an 8-bit register that describes the format of the PCI Configuration Space locations 10h to 3Ch and that identifies a device to be single or multi function. The PCI Header Type register is located at address 0Eh in the PCI Configuration Space. It is read only.

Bit	Name	Description
7	FUNCT	Single function/multi function device. Read as ZERO, write operations have no effect. The PCnet-PCI II controller is a single function device.
6–0	LAYOUT	PCI configuration space layout. Read as ZEROs, write operations have no effect. The layout of the PCI configuration space locations 10h to 3Ch is as shown in the table at the beginning of this section.

PCI I/O Base Address Register (Offset 10h)

The PCI I/O Base Address register is a 32-bit register that determines the location of the PCnet-PCI II controller I/O resources in all of I/O space. It is located at offset 10h in the PCI Configuration Space.

31–5	IOBASE	I/O base address most significant 27 bits. These bits are written by the host to specify the location of the PCnet-PCI II controller I/O resources in all of I/O space. IOBASE must be written with a valid address before the PCnet-PCI II controller slave I/O mode is turned on by setting the IOEN bit (PCI Command register, bit 0). When the PCnet-PCI II controller is enabled for I/O mode (IOEN is set), it monitors the PCI bus for a valid I/O command. If the value on AD[31:5] during the address phase of the cycles matches the value of IOBASE, the PCnet-PCI II controller will drive $\overline{\text{DEVSEL}}$ indicating it will respond to the access. IOBASE is read and written by the host. IOBASE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
4–2	IOSIZE	I/O size requirements. Read as ZEROs, write operations have no effect. IOSIZE indicates the size of the I/O space the PCnet-PCI II controller requires. When the host writes a value of FFFF FFFFh to the I/O Base Address register, it will read back a value of ZERO in bits 4–2. That indicates a PCnet-PCI II controller I/O space requirement of 32 bytes.
1	RES	Reserved location. Read as ZERO, write operations have no effect.
0	IOSPACE	I/O space indicator. Read as ONE, write operations have no effect. Indicating that this base address register describes an I/O base address.

PCI Memory Mapped I/O Base Address Register (Offset 14h)

The PCI Memory Mapped I/O Base Address register is a 32-bit register that determines the location of the PCnet-PCI II controller I/O resources in all of memory space. It is located at offset 14h in the PCI Configuration Space.

Bit	Name	Description
31–5	MEMBASE	Memory mapped I/O base address most significant 27 bits. These bits are written by the host to specify the location of the PCnet-PCI II controller I/O resources in all of memory space. MEMBASE must be written with a valid address before the PCnet-PCI II controller slave memory mapped I/O mode is turned on by setting the MEMEN bit (PCI Command register, bit 1).

		When the PCnet-PCI II controller is enabled for memory mapped I/O mode (MEMEN is set), it monitors the PCI bus for a valid memory command. If the value on AD[31:5] during the address phase of the cycles matches the value of MEMBASE, the PCnet-PCI II controller will drive $\overline{\text{DEVSEL}}$ indicating it will respond to the access. MEMBASE is read and written by the host. MEMBASE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
4	MEMSIZE	Memory mapped I/O size requirements. Read as ZEROs, write operations have no effect. MEMSIZE indicates the size of the memory space the PCnet-PCI II controller requires. When the host writes a value of FFFF FFFFh to the Memory Mapped I/O Base Address register, it will read back a value of ZERO in bit 4. That indicates a PCnet-PCI II controller memory space requirement of 32 bytes.
3	PREFETCH	Prefetchable. Read as ZERO, write operations have no effect. Indicates that memory space controlled by this base address register is not prefetchable. Data in the memory mapped I/O space cannot be prefetched. Because one of I/O resources in this address space is a Reset register, the order of the read accesses is important.
2–1	TYPE	Memory type indicator. Read as ZEROs, write operations have no effect. Indicates that this base address register is 32 bits wide and mapping can be done anywhere in the 32-bit memory space.
0	MEMSPACE	Memory space indicator. Read as ZERO, write operations have no effect. Indicates that this base address register describes a memory base address.

PCI Expansion ROM Base Address Register (Offset 30h)

The PCI Expansion ROM Base Address register is a 32-bit register that defines the base address, size and address alignment of an Expansion ROM. It is located at offset 30h in the PCI Configuration Space.

Bit	Name	Description
31–16	ROMBASE	Expansion ROM base address most significant 16 bits. These bits are written by the host to specify the location of the Expansion ROM in all of memory space. ROMBASE must be written with a valid address before the PCnet-PCI II controller Expansion ROM access is enabled by setting ROMEN (PCI Expansion ROM Base Address register, bit 0) and MEMEN (PCI Command register, bit 1). Since the 16 most significant bits of the base address are programmable, the host can map the Expansion ROM on any 64K boundary. When the PCnet-PCI II controller is enabled for Expansion ROM access (ROMEN and MEMEN are set to ONE), it monitors the PCI bus for a valid memory command. If the value on AD[31:2] during the address phase of the cycle falls between ROMBASE and ROMBASE + 64K – 4, the PCnet-PCI II controller will drive $\overline{\text{DEVSEL}}$ indicating it will respond to the access. ROMBASE is read and written by the host. ROMBASE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
15–11	ROMSIZE	ROM size. Read as ZEROs, write operation have no effect. ROMSIZE indicates the maximum size of the Expansion ROM the PCnet-PCI II controller can support. The host can determine the Expansion ROM size by writing FFFFF800h to the Expansion ROM Base Address register. It will read back a value of ZERO in bits 15–11, indicating an Expansion ROM size of 64K. Note that ROMSIZE only specifies the maximum size of Expansion ROM the PCnet-PCI II

		controller supports. A smaller ROM can be used, too. The actual size of the code in the Expansion ROM is always determined by reading the Expansion ROM header.
10–1	RES	Reserved location. Read as ZEROs, write operations have no effect.
0	ROMEN	Expansion ROM enable. Written by the host to enable access to the Expansion ROM. The PCnet-PCI II controller will only respond to accesses to the Expansion ROM when both ROMEN and MEMEN (PCI Command register, bit 1) are set to ONE. ROMEN is read and written by the host. ROMEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.

PCI Interrupt Line Register (Offset 3Ch)

The PCI Interrupt Line register is an 8-bit register that is used to communicate the routing of the interrupt. This register is written by the POST software as it initializes the PCnet-PCI II controller in the system. The register is read by the network driver to determine the interrupt channel which the POST software has assigned to the PCnet-PCI II controller. The PCI Interrupt Line register is not modified by the PCnet-PCI II controller. It has no effect on the operation of the device.

The PCI Interrupt Line register is located at offset 3Ch in the PCI Configuration Space. It is read and written by the host. It is cleared by H_RESET and is not affected S_RESET or by setting the STOP bit.

PCI Interrupt Pin Register (Offset 3Dh)

This PCI Interrupt Pin register is an 8-bit register that indicates the interrupt pin that the PCnet-PCI II controller is using. The value for the PCnet-PCI II controller Interrupt Pin register is 01h, which corresponds to $\overline{\text{INTA}}$.

The PCI Interrupt Pin register is located at offset 3Dh in the PCI Configuration Space. It is read only.

PCI MIN_GNT Register (Offset 3Eh)

The PCI MIN_GNT register is an 8-bit register that specifies the minimum length of a burst period that the PCnet-PCI II controller needs to keep up with the network activity. The length of the burst period is calculated assuming a clock rate of 33 MHz. The register value specifies the time in units of 1/4 ms. The PCI MIN_GNT register is an alias of BCR22, bits 7–0. The default value for MIN_GNT is 06h, which corresponds to a minimum grant of 1.5 µs. One and a half µs is the time it takes the PCnet-PCI II controller to read/write 64 bytes. (16 DWord transfers in burst mode with one extra wait state per data phase inserted by the target.)

Note that the default is only a typical value. This calculation also does not take into account any descriptor accesses.

The host should use the value in this register to determine the setting of the PCI Latency Timer register.

The PCIMIN_GNT register is located at offset 3Eh in the PCI Configuration Space. It is read only.

PCI MAX_LAT Register (Offset 3Fh)

The PCI MAX_LAT register is an 8-bit register that specifies the maximum arbitration latency the PCnet-PCI II controller can sustain without causing problems to the network activity. The register value specifies the time in units of 1/4 microseconds. The MAX_LAT register is an alias of BCR22, bits 15–8. The default value for MAX_LAT is FFh, which corresponds to a maximum latency of 63.75 µs. The actual maximum latency the PCnet-PCI II controller can handle is 153.6 µs, which is also the value for the bus time-out (see CSR100).

The host should use the value in this register to determine the setting of the PCI Latency Timer register.

The PCIMAX_LAT register is located at offset 3Fh in the PCI Configuration Space. It is read only.

RAP Register

The RAP (Register Address Pointer) register is used to gain access to CSR and BCR registers on board the PCnet-PCI II controller. The RAP contains the address of a CSR or BCR.

As an example of RAP use, consider a read access to CSR4. In order to access this register, it is necessary to first load the value 0004h into the RAP by performing a write access to the RAP offset of 12h (12h when WIO mode has been selected, 14h when DWIO mode has been selected). Then a second access is performed, this time to the RDP offset of 10h (for either WIO or DWIO mode). The RDP access is a read access, and since RAP has just been loaded with the value of 0004h, the RDP read will yield the contents of CSR4. A read of the BDP at this time (offset of 16h when WIO mode has been selected, 1Ch when DWIO mode has been selected) will yield the contents of BCR4, since the RAP is used as the pointer into both BDP and RDP space.

RAP: Register Address Port

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	RES	Reserved locations. Read and written as ZEROs.
7–0	RAP	Register Address Port. The value of these 8 bits determines which CSR or BCR will be accessed when an I/O access to the RDP or BDP port, respectively, is performed.

A write access to undefined CSR or BCR locations may cause unexpected reprogramming of the PCnet-PCI II controller control registers. A read access will yield undefined values.

Read/Write accessible always. RAP is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

Control and Status Registers

The CSR space is accessible by performing accesses to the RDP (Register Data Port). The particular CSR that is read or written during an RDP access will depend upon the current setting of the RAP. RAP serves as a pointer into the CSR space.

CSR0: PCnet-PCI II Controller Status Register

Bit	Name	Description
		Certain bits in CSR0 indicate the cause of an interrupt. The register is designed so that these indicator bits are cleared by writing ONEs to those bit locations. This means that the software can read CSR0 and write back the value just read to clear the interrupt condition.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	ERR	Error is set by the ORing of BABL, CERR, MISS, and MERR. ERR remains set as long as any of the error flags are true. Read accessible always. ERR is read only. Write operations are ignored.
14	BABL	Babble is a transmitter time-out error. BABL is set by the PCnet-PCI II controller when the transmitter has been on the channel longer than the time required to send the maximum length frame. BABL will be set if 1519 bytes or greater are transmitted. When BABL is set, $\overline{INTA}$ is asserted if IENA is ONE and the mask bit BABLM (CSR3, bit 14) is ZERO. BABL assertion will set the ERR bit, regardless of the settings of IENA and BABLM. Read/Write accessible always. BABL is cleared by the host by writing a ONE. Writing a ZERO has no effect. BABL is cleared by H_RESET, S_RESET or by setting the STOP bit.
13	CERR	Collision Error is set by the PCnet-PCI II controller when the device operates in half-duplex mode and the collision inputs to the AUI port failed to activate within 20 network bit times after the chip terminated transmission (SQE Test). This feature is a transceiver test feature. CERR reporting is disabled when the AUI interface is active and the PCnet-PCI II controller operates in full-duplex mode. In 10BASE-T mode, for both half-duplex and full-duplex operation, CERR will be set after a transmission if the T-MAU is in Link Fail state. CERR assertion will not result in an interrupt being generated. CERR assertion will set the ERR bit. Read/Write accessible always. CERR is cleared by the host by writing a ONE. Writing a ZERO has no effect. CERR is cleared by H_RESET, S_RESET or by setting the STOP bit.
12	MISS	Missed Frame is set by the PCnet-PCI II controller when it looses an incoming receive frame because a receive descriptor was not available. This bit is the only immediate indication that receive data has been lost since there is no current receive descriptor. The Missed Frame Counter (CSR112) also increments each time a receive frame is missed.

When MISS is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit MISSM (CSR3, bit 12) is ZERO. MISS assertion will set the ERR bit, regardless of the settings of IENA and MISSM.

Read/Write accessible always. MISS is cleared by the host by writing a ONE. Writing a ZERO has no effect. MISS is cleared by H_RESET, S_RESET or by setting the STOP bit.

11 MERR Memory Error is set by the PCnet-PCI II controller when it requests the use of the system interface bus by asserting $\overline{\text{REQ}}$ and $\overline{\text{GNT}}$ has not been asserted after a programmable length of time. The length of time in microseconds before MERR is asserted will depend upon the setting of the Bus Timeout register (CSR100). The default setting of CSR100 will set MERR after 153.6 µs of bus latency.

When MERR is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit MERRM (CSR3, bit 11) is ZERO. MERR assertion will set the ERR bit, regardless of the settings of IENA and MERRM.

Read/Write accessible always. MERR is cleared by the host by writing a ONE. Writing a ZERO has no effect. MERR is cleared by H_RESET, S_RESET or by setting the STOP bit.

10 RINT Receive Interrupt is set by the PCnet-PCI II controller after the last descriptor of a receive frame has been updated by writing a ZERO to the OWN bit. RINT may also be set when the first descriptor of a receive frame has been updated by writing a ZERO to the OWN bit if the LAPPEN bit of CSR3 has been set to ONE.

When RINT is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit RINTM (CSR3, bit 10) is ZERO.

Read/Write accessible always. RINT is cleared by the host by writing a ONE. Writing a ZERO has no effect. RINT is cleared by H_RESET, S_RESET or by setting the STOP bit.

9 TINT Transmit Interrupt is set by the PCnet-PCI II controller after the OWN bit in the last descriptor of a transmit frame has been cleared to indicate the frame has been sent or an error occurred in the transmission. When TINT is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit TINTM (CSR3, bit 9) is ZERO.

TINT will not be set if TINTOKD (CSR122, bit 2) is set to ONE and the transmission was successful.

Read/Write accessible always. TINT is cleared by the host by writing a ONE. Writing a ZERO has no effect. TINT is cleared by H_RESET, S_RESET or by setting the STOP bit.

8 IDON Initialization Done is set by the PCnet-PCI II controller after the initialization sequence has completed. When IDON is set, the PCnet-PCI II controller has read the initialization block from memory.

When IDON is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit IDONM (CSR3, bit 8) is ZERO.

Read/Write accessible always. IDON is cleared by the host by writing a ONE. Writing a ZERO has no effect. IDON is cleared by H_RESET, S_RESET or by setting the STOP bit.

7 INTR Interrupt Flag indicates that one or more following interrupt causing conditions has occurred: BABL, EXDINT, IDON, JAB, MERR, MISS, MFCO, MPINT, RVCC, RINT, SINT, SLPINT, TINT, TXSTRT or UINT and the associated mask or enable bit is programmed to allow the event to cause an interrupt. If IENA is set to ONE and INTR is set, $\overline{\text{INTA}}$ will be active. When INTR is set by SINT or SLPINT, $\overline{\text{INTA}}$ will be active independent of the state of INEA.

Read accessible always. INTR is read only. INTR is cleared by clearing all of the active individual interrupt bits that have not been masked out.

6 IENA Interrupt Enable allows $\overline{\text{INTA}}$ to be active if the Interrupt Flag is set. If IENA is cleared to ZERO, $\overline{\text{INTA}}$ will be disabled regardless of the state of INTR.

Read/Write accessible always. IENA is set by writing a ONE and cleared by writing a ZERO. IENA is cleared by H_RESET, S_RESET or by setting the STOP bit.

5 RXON Receive On indicates that the receive function is enabled. RXON is set to ONE if DRX (CSR15, bit 0) is cleared to ZERO after the START bit is set. If INIT and START are set together, RXON will not be set until after the initialization block has been read in.

Read accessible always. RXON is read only. RXON is cleared by H_RESET, S_RESET or by setting the STOP bit.

4 TXON Transmit On indicates that the transmit function is enabled. TXON is set to ONE if DTX (CSR15, bit 1) is cleared to ZERO after the START bit is set. If INIT and START are set together, TXON will not be set until after the initialization block has been read in.

Read accessible always. TXON is read only. TXON is cleared by H_RESET, S_RESET or by setting the STOP bit.

3 TDMD Transmit Demand, when set, causes the buffer management unit to access the transmit descriptor ring without waiting for the poll-time counter to elapse. If TXON is not enabled, TDMD bit will be cleared and no transmit descriptor ring access will occur.

If the DPOLL bit in CSR4 is set, automatic polling is disabled and TDMD can be used to start a transmission.

Read/Write accessible always. TDMD is set by writing a ONE. Writing a ZERO has no effect. TDMD will be cleared by the buffer management unit when it polls a transmit descriptor. TDMD is cleared by H_RESET, S_RESET or by setting the STOP bit.

2 STOP STOP assertion disables the chip from all DMA and network activity. The chip remains inactive until either STRT or INIT are set. If STOP, STRT and INIT are all set together, STOP will override STRT and INIT.

Read/Write accessible always. STOP is set by writing a ONE, by H_RESET or S_RESET. Writing a ZERO has no effect. STOP is cleared by setting either STRT or INIT.

1 STRT STRT assertion enables the PCnet-PCI II controller to send and receive frames and perform buffer management operations. Setting STRT clears the STOP bit. If STRT and INIT are set together, the PCnet-PCI II controller initialization will be performed first.

Read/Write accessible always. STRT is set by writing a ONE. Writing a ZERO has no effect. STRT is cleared by H_RESET, S_RESET or by setting the STOP bit.

0 INIT INIT assertion enables the PCnet-PCI II controller to begin the initialization procedure which reads the initialization block from memory. Setting INIT clears the STOP bit. If STRT and INIT are set together, the PCnet-PCI II controller initialization will be performed first. INIT is not cleared when the initialization sequence has completed.

Read/Write accessible always. INIT is set by writing a ONE. Writing a ZERO has no effect. INIT is cleared by H_RESET, S_RESET or by setting the STOP bit.

CSR1: Initialization Block Address 0

Bit	Name	Description
		This register is aliased with CSR16.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IADR[15:0]	Lower 16 bits of the address of the initialization block. Bit locations 1 and 0 must both be ZERO to align the initialization block to a DWord boundary. Read/Write accessible only when either the STOP or the SPND bit is set. Unaffected by H_RESET or S_RESET or by setting the STOP bit.

CSR2: Initialization Block Address 1

Bit	Name	Description
		This register is aliased with CSR17.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	IADR[31:24]	If SSIZE32 (BCR20, bit 8) is cleared to ZERO, then the IADR[31:24] bits will be used to generate the upper 8 bits of all bus mastering addresses, as required for a 32-bit address bus. Note that the 16-bit software structures will yield only 24 bits of address for PCnet-PCI II controller bus master accesses. The PCnet-PCI II controller is designed for 32-bit systems which require 32 bits of address. Therefore, whenever SSIZE32 is cleared to ZERO, the IADR[31:24] bits will be appended to the 24-bit initialization address, to each 24-bit descriptor base address and to each beginning 24-bit buffer address in order to form complete 32-bit addresses. The upper 8 bits that exist in the descriptor address registers and the buffer address registers which are stored on board the PCnet-PCI II controller will be overwritten with the IADR[31:24] value, so that CSR accesses to these registers will show the 32 bit address that includes the appended field. If SSIZE32 is set to ONE, then the IADR[31:24] bits will be used strictly as the upper 8 bits of the initialization block address. In this mode, software will provide 32-bit pointer values for all of the shared software structures-i.e. descriptor bases and buffer addresses. Read/Write accessible only when either the STOP or the SPND bit is set. Unaffected by H_RESET, S_RESET or by setting the STOP bit.
7–0	IADR[23:16]	Bits 23 through 16 of the address of the initialization block. Read/Write accessible only when either the STOP or the SPND bit is set. Unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR3: Interrupt Masks and Deferral Control

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	RES	Reserved location. Read and written as ZERO.
14	BABLM	Babble Mask. If BABLM is set, the BABL bit will be masked and unable to set the INTR bit. Read/Write accessible always. BABLM is cleared by H_RESET or S_RESET and is not affected by STOP.
13	RES	Reserved location. Read and written as ZERO.
12	MISSM	Missed Frame Mask. If MISSM is set, the MISS bit will be masked and unable to set the INTR bit. Read/Write accessible always. MISSM is cleared by H_RESET or S_RESET and is not affected by STOP.
11	MERRM	Memory Error Mask. If MERRM is set, the MERR bit will be masked and unable to set the INTR bit. Read/Write accessible always. MERRM is cleared by H_RESET

		or S_RESET and is not affected by STOP.
10	RINTM	Receive Interrupt Mask. If RINTM is set, the RINT bit will be masked and unable to set the INTR bit. Read/Write accessible always. RINTM is cleared by H_RESET or S_RESET and is not affected by STOP.
9	TINTM	Transmit Interrupt Mask. If TINTM is set, the TINT bit will be masked and unable to set the INTR bit. Read/Write accessible always. TINTM is cleared by H_RESET or S_RESET and is not affected by STOP.
8	IDONM	Initialization Done Mask. If IDONM is set, the IDON bit will be masked and unable to set the INTR bit. Read/Write accessible always. IDONM is cleared by H_RESET or S_RESET and is not affected by STOP.
7	RES	Reserved location. Read and written as ZEROs.
6	DXSUFLO	Disable Transmit Stop on Underflow error. When DXSUFLO is cleared to ZERO, the transmitter is turned off when an UFLO error occurs (CSR0, TXON = 0). When DXSUFLO is set to ONE, the PCnet-PCI II controller gracefully recovers from an UFLO error. It scans the transmit descriptor ring until it finds the start of a new frame and starts a new transmission. Read/Write accessible always. DXSUFLO is cleared by H_RESET or S_RESET and is not affected by STOP.
5	LAPPEN	Look Ahead Packet Processing Enable. When set to ONE, the LAPPEN bit will cause the PCnet-PCI II controller to generate an interrupt following the descriptor write operation to the first buffer of a receive frame. This interrupt will be generated in addition to the interrupt that is generated following the descriptor write operation to the last buffer of a receive packet. The interrupt will be signaled through the RINT bit of CSR0. Setting LAPPEN to ONE also enables the PCnet-PCI II controller to read the STP bit of receive descriptors. The PCnet-PCI II controller will use the STP information to determine where it should begin writing a receive packet's data. Note that while in this mode, the PCnet-PCI II controller can write intermediate packet data to buffers whose descriptors do not contain STP bits set to ONE. Following the write to the last descriptor used by a packet, the PCnet-PCI II controller will scan through the next descriptor entries to locate the next STP bit that is set to ONE.The PCnet-PCI II controller will begin writing the next packet's data to the buffer pointed to by that descriptor. Note that because several descriptors may be allocated by the host for each packet, and not all messages may need all of the descriptors that are allocated between descriptors that have STP set to ONE, then some descriptors/buffers may be skipped in the ring. While performing the search for the next STP bit that is set to ONE, the PCnet-PCI II controller will advance through the receive descriptor ring regardless of the state of ownership bits. If any of the entries that are examined during this search indicate PCnet-PCI II controller ownership of the descriptor but also have STP cleared to ZERO, the PCnet-PCI II controller will clear the OWN bit to ZERO in these entries. If a scanned entry indicates host ownership with STP cleared to ZERO, the PCnet-PCI II controller will not alter the entry, but will advance to the next entry. When the STP bit is set to ONE, but the descriptor that contains this setting is not owned by the PCnet-PCI II controller, then the PCnet-PCI II controller will stop

advancing through the ring entries and begin periodic polling of this entry. When the STP bit is set to ONE, and the descriptor that contains this setting is owned by the PCnet-PCI II controller, then the PCnet-PCI II controller will stop advancing through the ring entries, store the descriptor information that it has just read, and wait for the next receive to arrive.

This behavior allows the host software to pre-assign buffer space in such a manner that the header portion of a receive packet will always be written to a particular memory area, and the data portion of a receive packet will always be written to a separate memory area. The interrupt is generated when the header bytes have been written to the header memory area.

Read/Write accessible always. LAPPEN bit is cleared by H_RESET or S_RESET and is not affected by STOP.

See Appendix D for more information on the Look Ahead Packet Processing concept.

4 DXMT2PD Disable Transmit Two Part Deferral (see the section “Medium Allocation” for more details). If DXMT2PD is set, Transmit Two Part Deferral will be disabled.

Read/Write accessible always. DXMT2PD is cleared by H_RESET or S_RESET and is not affected by STOP.

3 EMBA Enable Modified Back-off Algorithm (see the section “Collision Handling” for more details). If EMBA is set, a modified back-off algorithm is implemented.

Read/Write accessible always. EMBA is cleared by H_RESET or S_RESET and is not affected by STOP.

2 BSWP Byte Swap. This bit is used to choose between big and little endian modes of operation. When BSWP is set to ONE, big endian mode is selected. When BSWP is cleared to ZERO, little endian mode is selected.

When big endian mode is selected, the PCnet-PCI II controller will swap the order of bytes on the AD bus during a data phase on accesses to the FIFOs only: AD[31:24] is byte 0, AD[23:16] is byte 1, AD[15:8] is byte 2 and AD[7:0] is byte 3.

When little endian mode is selected, the order of bytes on the AD bus during a data phase is: AD[31:24] is byte 3, AD[23:16] is byte 2, AD[15:8] is byte 1 and AD[7:0] is byte 0.

Byte swap only affects data transfers that involve the FIFOs. Initialization block transfers, descriptor transfers, RDP, RAP, BDP and PCI configuration space accesses, Address PROM transfers, and Expansion ROM accesses are not affected by the setting of the BSWP bit.

Note that the byte ordering of the PCI bus is defined to be little endian. BSWP should not be set to ONE when the PCnet-PCI II controller is used in a PCI bus application.

Read/Write accessible always. BSWP is cleared by H_RESET or S_RESET and is not affected by STOP.

1 RES Reserved location. The default value of this bit is a ZERO. Writing a ONE to this bit has no effect on device function. If a ONE is written to this bit, then a ONE will be read back. Existing drivers may write a ONE to this bit for compatibility, but new drivers should write a ZERO to this bit and should treat the read value as undefined.

0 RES Reserved location. The default value of this bit is a ZERO. Writing a ONE to this bit has no effect on device function. If a ONE is written to this bit, then a ONE will be read back. Existing drivers may write a ONE to this bit for compatibility, but new drivers should write a ZERO to this bit and should treat the read value as undefined.

CSR4: Test and Features Control

Bit Name Description

Certain bits in CSR4 indicate the cause of an interrupt. The register is designed so that these indicator bits are cleared by writing ONEs to those bit locations. This means that the software can read CSR4 and write back the value just read to clear the interrupt condition.

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15 EN124 Enable CSR124 access. Setting EN124 to ONE allows the user to write to bits in CSR124 which enable Runt Packet Accept mode (RPA, bit 3). Once these bits are accessed EN124 must be cleared back to ZERO.

Read/Write accessible always. EN124 is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

In order to set EN124, it must be written with a ONE during the first write access to CSR4 after H_RESET or S_RESET. Once a ZERO is written to this bit position, EN124 cannot be set until after the PCnet-PCI II controller is reset by H_RESET or S_RESET.

14 DMAPLUS When DMAPLUS is set to ONE, the DMA Burst Transfer Counter in CSR80 is disabled. If DMAPLUS is cleared to ZERO, the counter is enabled.

DMAPLUS should be set to ONE when the PCnet-PCI II controller is used in a PCI bus application.

Read/Write accessible always. DMAPLUS is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

13 TIMER Enable Bus Activity Timer. If TIMER is set to ONE, the Bus Activity Timer (CSR82) is enabled. If TIMER is cleared, the Bus Activity Timer is disabled.

TIMER should stay at its default value of ZERO when the PCnet-PCI II controller is used in a PCI bus application.

Read/Write accessible always. TIMER is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

12 DPOLL Disable Transmit Polling. If DPOLL is set, the Buffer Management Unit will disable transmit polling. If DPOLL is cleared, automatic transmit polling is enabled. If DPOLL is set, the TDMD bit in CSR0 must be set in order to initiate a manual poll of a transmit descriptor. Transmit descriptor polling will not take place if TXON is cleared.

Read/Write accessible always. DPOLL is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

11 APAD_XMT Auto Pad Transmit. When set, APAD_XMT enables the automatic padding feature. Transmit frames will be padded to extend them to 64 bytes including FCS. The FCS is calculated for the entire frame including pad, and appended after the pad. APAD_XMT will override the programming of the DXMTFCS bit (CSR15, bit 3) and of the ADD_FCS/NO_FCS bit (TMD1, bit 29).

Read/Write accessible always. APAD_XMT is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

10 ASTRP_RCV Auto Strip Receive. When set, ASTRP_RCV enables the automatic pad stripping feature. The pad and FCS fields will be stripped from receive frames and not placed in the FIFO.

Read/Write accessible always. ASTRP_RCV is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

9 MFCO Missed Frame Counter Overflow is set by the PCnet-PCI II controller when the Missed Frame Counter (CSR112) wraps around.

When MFCO is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit MFCOM is ZERO.

Read/Write accessible always. MFCO is cleared by the host by writing a ONE. Writing a ZERO has no effect. MFCO is cleared by H_RESET, S_RESET or by setting the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI II controller will never set the value of this bit to ONE.

8 MFCOM Missed Frame Counter Overflow Mask. If MFCOM is set, the MFCO bit will be masked and unable to set the INTR bit.

Read/Write accessible always. MFCOM is set to ONE by H_RESET or S_RESET and is unaffected by setting the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI II controller will clear the value of this bit to ZERO.

7 UINTCMD User Interrupt Command. UINTCMD can be used by the host to generate an interrupt unrelated to any network activity. When UINTCMD is set, $\overline{\text{INTA}}$ is asserted if IENA is set to ONE. UINTCMD will be cleared internally after the PCnet-PCI II controller has set UINT to ONE.

Read/Write accessible always. UINTCMD is cleared by H_RESET or S_RESET or by setting the STOP bit.

6 UINT User Interrupt. UINT is set by the PCnet-PCI II controller after the host has issued a user interrupt command by setting UINTCMD (CSR4, bit 7) to ONE.

Read/Write accessible always. UINT is cleared by the host by writing a ONE. Writing a ZERO has no effect. UINT is cleared by H_RESET or S_RESET or by setting the STOP bit.

5 RCVCCO Receive Collision Counter Overflow is set by the PCnet-PCI II controller when the Receive Collision Counter (CSR114) wraps around.

When RCVCCO is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit RCVCCOM is ZERO.

Read/Write accessible always. RCVCCO is cleared by the host by writing a ONE. Writing a ZERO has no effect. RCVCCO is cleared by H_RESET, S_RESET or by setting the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI II controller will never set the value of this bit to ONE.

4 RCVCCOM Receive Collision Counter Overflow Mask. If RCVCCOM is set, the RCVCCO bit will be masked and unable to set the INTR bit.

Read/Write accessible always. RCVCCOM is set to ONE by H_RESET or S_RESET and is unaffected by setting the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI II controller will clear the value of this bit to ZERO.

3 TXSTRT Transmit Start status is set by the PCnet-PCI II controller whenever it begins transmission of a frame.

When TXSTRT is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit TXSTRTM is ZERO.

Read/Write accessible always. TXSTRT is cleared by the host by writing a ONE. Writing a ZERO has no effect. TXSTRT is cleared by H_RESET, S_RESET or by setting the STOP bit.

2 TXSTRTM Transmit Start Mask. If TXSTRTM is set, the TXSTRT bit

will be masked and unable to set the INTR bit.

Read/Write accessible always. TXSTRTM is set to ONE by H_RESET or S_RESET and is unaffected by setting the STOP bit.

1 JAB Jabber Error is set by the PCnet-PCI II controller when the T-MAU exceeds the allowed transmission time limit. Jabber can only be asserted in 10BASE-T mode.

When JAB is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit JABM is ZERO.

Read/Write accessible always. JAB is cleared by the host by writing a ONE. Writing a ZERO has no effect. JAB is cleared by H_RESET, S_RESET or by setting the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI II controller will never set the value of this bit to ONE.

0 JABM Jabber Error Mask. If JABM is set, the JAB bit will be masked and unable to set the INTR bit.

Read/Write accessible always. JABM is set to ONE by H_RESET or S_RESET and is unaffected by setting the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI II controller will clear the value of this bit to ZERO.

CSR5: Extended Control and Interrupt

Bit Name Description

Certain bits in CSR5 indicate the cause of an interrupt. The register is designed so that these indicator bits are cleared by writing ONEs to those bit locations. This means that the software can read CSR5 and write back the value just read to clear the interrupt condition.

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15 TOKINTD Transmit OK Interrupt Disable. If TOKINTD is set to ONE, the TINT bit in CSR0 will not be set when a transmission was successful. Only a transmit error will set the TINT bit.

TOKINTD has no effect when LTINTEN (CSR5, bit 14) is set to ONE. A transmit descriptor with LTINT set to ONE will always cause TINT to be set to ONE, independent of the success of the transmission.

Read/Write accessible always. TOKINTD is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

14 LTINTEN Last Transmit Interrupt Enable. When set to ONE, the LTINTEN bit will cause the PCnet-PCI II controller to read bit 28 of TMD1 as LTINT. The setting LTINT will determine if TINT will be set at the end of the transmission.

Read/Write accessible always. LTINTEN is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

13–12 RES Reserved locations. Written as ZEROs and read as undefined.

11 SINT System Interrupt is set by the PCnet-PCI II controller when it detects a system error during a bus master transfer on the PCI bus. System errors are data parity error, master abort or a target abort. The setting of SINT due to a data parity error is not dependent on the setting of PERREN (PCI Command register, bit 6).

When SINT is set, $\overline{\text{INTA}}$ is asserted if the enable bit SINTE is ONE. Note that the assertion of an interrupt due to SINT is not dependent on the state of the INEA bit, since INEA is cleared by the STOP reset generated by the system error.

Read/Write accessible always. SINT is cleared by the host by writing a ONE. Writing a ZERO has no effect. The state of SINT is not affected by clearing any of the PCI Status register bits that get set when a data parity error (DATAPERR, bit 8), master abort (RMABORT, bit 13) or target abort (RTABORT, bit 12) occurs. SINT is cleared by H_RESET or S_RESET and is not affected by setting the STOP bit.

10 SINTE System Interrupt Enable. If SINTE is set, the SINT bit will be able to set the INTR bit.

Read/Write accessible always. SINTE is cleared to ZERO by H_RESET or S_RESET and is not affected by setting the STOP bit.

9 SLPINT Sleep Interrupt is set by the PCnet-PCI II controller when it comes out of sleep mode.

When SLPINT is set, $\overline{\text{INTA}}$ is asserted if the enable bit SLPINTE is ONE. Note that the assertion of an interrupt due to SLPINT is not dependent on the state of the INEA bit, since INEA is cleared by S_RESET when entering the sleep mode.

Read/Write accessible always. SLPINT is cleared by the host by writing a ONE. Writing a ZERO has no effect. SLPINT is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

8 SLPINTE Sleep Interrupt Enable. If SLPINTE is set, the SLPINT bit will be able to set the INTR bit.

Read/Write accessible always. SLPINTE is cleared to ZERO by H_RESET and is not affected by S_RESET or by setting the STOP bit.

7 EXDINT Excessive Deferral Interrupt is set by the PCnet-PCI II controller when the transmitter has experienced Excessive Deferral on a transmit frame, where Excessive Deferral is defined in ISO 8802-3 (IEEE/ANSI 802.3).

When EXDINT is set, $\overline{\text{INTA}}$ is asserted if the enable bit EXDINTE is ONE.

Read/Write accessible always. EXDINT is cleared by the host by writing a ONE. Writing a ZERO has no effect. EXDINT is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

6 EXDINTE Excessive Deferral Interrupt Enable. If EXDINTE is set, the EXDINT bit will be able to set the INTR bit.

Read/Write accessible always. EXDINTE is cleared to ZERO by H_RESET and is not affected by S_RESET or by setting the STOP bit.

5 MPPLBA Magic Packet Physical Logical Broadcast Accept. If MPPLBA is cleared to ZERO, the PCnet-PCI II controller will only detect a magic packet if the destination address of the packet matches the content of the physical address register (PADR). If MPPLBA is set to ONE, the destination address of the magic packet can be unicast, multicast or broadcast. Note that the setting of MPPLBA only effects the address detection of the magic packet. The magic packet data sequence must be in all cases the same, i.e., a 16-times repetition of the the physical address (PADR[47:0]).

Read/Write accessible always. MPPLBA is cleared to ZERO by H_RESET or S_RESET and is not affected by setting the STOP bit.

4 MPINT Magic Packet Interrupt is set by the PCnet-PCI II controller when the device is in magic packet mode and it receives a magic packet.

When MPINT is set, $\overline{\text{INTA}}$ is asserted if IENA (CSR0, bit 6) and the enable bit MPINTE are set to ONE.

Read/Write accessible always. MPINT is cleared by the host by writing a ONE. Writing a ZERO has no effect. MPINT is cleared by H_RESET, S_RESET or by setting the STOP bit.

3 MPINTE Magic Packet Interrupt Enable. If MPINTE is set, the MPINT bit will be able to set the INTR bit.

Read/Write accessible always. MPINTE is cleared to ZERO by H_RESET or S_RESET and is not affected by setting the STOP bit.

2 MPEN Magic Packet Enable. MPEN allows activation of the magic packet mode by software. The PCnet-PCI II controller will enter the magic packet mode when both MPEN and MPMODE are set to ONE.

Read/Write accessible always. MPEN is cleared to ZERO by H_RESET or S_RESET and is not affected by setting the STOP bit.

1 MPMODE Magic Packet Mode. Setting MPMODE to ONE will redefine the $\overline{\text{SLEEP}}$ pin to be a magic packet enable pin. The PCnet-PCI II controller will enter the magic packet mode when MPMODE is set to one and either $\overline{\text{SLEEP}}$ is asserted or MPEN is set to ONE.

Read/Write accessible always. MPMODE is cleared to ZERO by H_RESET or S_RESET and is not affected by setting the STOP bit.

0 SPND Suspend. Setting SPND to ONE will cause the PCnet-PCI II controller to start entering the suspend mode. The host must poll SPND until it reads back ONE to determine that the PCnet-PCI II controller has entered the suspend mode. Setting SPND to ZERO will get the PCnet-PCI II controller out of suspend mode. SPND can only be set to ONE if STOP (CSR0, bit 2) is cleared to ZERO. H_RESET, S_RESET or setting the STOP bit will get the PCnet-PCI II controller out of suspend mode.

When the host requests the PCnet-PCI II controller to enter the suspend mode, the device 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 and BCR registers are accessible. As long as the PCnet-PCI II controller is not reset while in suspend mode (by H_RESET, S_RESET or by setting the STOP bit), no re-initialization of the device is required after the device comes out of suspend mode. The PCnet-PCI II controller will continue at the transmit and receive descriptor ring locations where it had left off.

Read/Write accessible always. SPND is cleared by H_RESET, S_RESET or by setting the STOP bit.

CSR6: RX/TX Descriptor Table Length

Bit Name Description

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15–12 TLEN Contains a copy of the transmit encoded ring length (TLEN) field read from the initialization block during PCnet-PCI II controller initialization. This field is written during the PCnet-PCI II controller initialization routine.

Read accessible only when either the STOP or the SPND bit is set. Write operations have no effect and should not be performed. TLEN is only defined after initialization. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

11–8 RLEN Contains a copy of the receive encoded ring length (RLEN) read from the initialization block during PCnet-PCI II controller initialization. This field is written during the PCnet-PCI II controller initialization routine.

Bit	Name	Description
		Read accessible only when either the STOP or the SPND bit is set. Write operations have no effect and should not be performed. RLEN is only defined after initialization. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.
7–0	RES	Reserved locations. Read as ZEROs. Write operations are ignored.

CSR8: Logical Address Filter 0

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LADRF[15:0]	Logical Address Filter, LADRF[15:0]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR9: Logical Address Filter 1

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LADRF[31:16]	Logical Address Filter, LADRF[31:16]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR10: Logical Address Filter 2

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LADRF[47:32]	Logical Address Filter, LADRF[47:32]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR11: Logical Address Filter 3

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LADRF[63:48]	Logical Address Filter, LADRF[63:48]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR12: Physical Address Register 0

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PADR[15:0]	Physical Address Register, PADR[15:0]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected

by H_RESET, S_RESET or by setting the STOP bit.

CSR13: Physical Address Register 1

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PADR[31:16]	Physical Address Register, PADR[31:16]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR14: Physical Address Register 2

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PADR[47:32]	Physical Address Register, PADR[47:32]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR15: Mode

Bit	Name	Description
		This register's fields are loaded during the PCnet-PCI II controller initialization routine with the corresponding initialization block values. The host can also write directly to this register.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	PROM	Promiscuous Mode. When PROM is set to ONE, all incoming receive frames are accepted. Read/Write accessible only when either the STOP or the SPND bit is set.
14	DRCVBC	Disable Receive Broadcast. When set, this bit disables the PCnet-PCI II controller from receiving broadcast messages. DRCVBC has no effect when PROM is set to ONE. Read/Write accessible only when either the STOP or the SPND bit is set. DRCVBC is cleared by H_RESET or S_RESET and not affected by STOP.
13	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the PCnet-PCI II controller will be disabled. Frames addressed to the node's individual physical address will not be recognized. DRCVPA has no effect when PROM is set to ONE. Read/Write accessible only when either the STOP or the SPND bit is set.
12	DLNKTST	Disable Link Status. When DLNKTST is set to ONE, monitoring of Link Pulses is disabled. When DLNKTST is cleared to ZERO, monitoring of Link Pulses is enabled. This bit only has meaning when the 10BASE-T network interface is selected. Read/Write accessible only when either the STOP or the SPND bit is set.
11	DAPC	Disable Automatic Polarity Correction. When DAPC is set to ONE, the 10BASE-T receive polarity reversal algorithm is disabled. When DAPC is cleared to ZERO, the polarity reversal algorithm is enabled. This bit only has meaning when the 10BASE-T network interface is selected. Read/Write accessible only when either the STOP or the SPND bit is set.
10	MENDECL	MENDEC Loopback Mode. See the description of the LOOP bit in CSR15.

Read/Write accessible only when either the STOP or the SPND bit is set.

9 LRT Low Receive Threshold (T-MAU Mode only)

TSEL Transmit Mode Select (AUI Mode only)

LRT Low Receive Threshold. When LRT is set to ONE, the internal twisted pair receive thresholds are reduced by 4.5 dB below the standard 10BASE-T value (approximately 3/5) and the unsquelch threshold for the RXD circuit will be 180 mV–312 mV peak.

Table 22. Network Port Configuration

PORTSEL[1:0]	ASEL(BCR2[1])	Link Status (of 10BASE-T)	Network Port
0X	1	Fail	AUI
0X	1	Pass	10BASE-T
00	0	X	AUI
01	0	X	10BASE-T
10	X	X	Reserved
11	X	X	Reserved

When LRT is cleared to ZERO, the unsquelch threshold for the RXD circuit will be the standard 10BASE-T value of 300 mV–520 mV peak.

In either case, the RXD circuit post squelch threshold will be one half of the unsquelch threshold.

This bit only has meaning when the 10BASE-T network interface is selected.

Read/Write accessible only when either the STOP or the SPND bit is set. Cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

TSEL Transmit Mode Select. TSEL controls the levels at which the AUI drivers rest when the AUI transmit port is idle. When TSEL is cleared to ZERO, DO+ and DO–yield zero differential to operate transformer coupled loads (Ethernet 2 and 802.3). When TSEL is set to ONE, the DO+ idles at a higher value with respect to DO–, yielding a logical HIGH state (Ethernet 1).

This bit only has meaning when the AUI network interface is selected.

Read/Write accessible only when either the STOP or the SPND bit is set. Cleared by H_RESET or S_RESET.

8–7PORTSEL[1:0] Port Select bits allow for software controlled selection of the network medium.

The only legal values for this field are 00 and 01.

PORTSEL setting of AUI and 10BASE-T are ignored when the ASEL bit of BCR2 (bit 1) has been set to ONE.

Read/Write accessible only when either the STOP or the SPND bit is set. Cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

6 INTL Internal Loopback. See the description of LOOP (CSR15, bit 2).

Read/Write accessible only when either the STOP or the SPND bit is set.

5 DRTY Disable Retry. When DRTY is set to ONE, PCnet-PCI II controller will attempt only one transmission. In this mode, the device will not protect the first 64 bytes of frame data in the transmit FIFO from being overwritten, because automatic retransmission will not be necessary. When DRTY is cleared to ZERO, the PCnet-PCI

II controller will attempt 16 transmissions before signaling a retry error.

Read/Write accessible only when either the STOP or the SPND bit is set.

4 FCOLL Force Collision. This bit allows the collision logic to be tested. The PCnet-PCI II controller must be in internal loopback for FCOLL to be valid. If FCOLL is set to ONE, a collision will be forced during loopback transmission attempts, which will result in a Retry Error. If FCOLL is cleared to ZERO, the Force Collision logic will be disabled. FCOLL is defined after the initialization block is read.

Table 23. Loopback Configuration

LOOP	INTL	MENDECL	Loopback Mode
0	X	X	Non-loopback
1	0	X	External Loopback
1	1	0	Internal Loopback Include MENDEC
1	1	1	Internal Loopback Exclude MENDEC

Read/Write accessible only when either the STOP or the SPND bit is set.

3 DXMTFCS Disable Transmit CRC (FCS). When DXMTFCS is cleared to ZERO, the transmitter will generate and append an FCS to the transmitted frame. When DXMTFCS is set to ONE, no FCS is generated or sent with the transmitted frame. DXMTFCS is overridden when ADD_FCS is set in TMD1.

If DXMTFCS is set and ADD_FCS is clear for a particular frame, no FCS will be generated. The value of ADD_FCS is valid only when STP is set in TMD1. If ADD_FCS is set for a particular frame, the state of DXMTFCS is ignored and a FCS will be appended on that frame by the transmit circuitry. See also the ADD_FCS bit in TMD1.

This bit is called DTCR in the C-LANCE (Am79C90).

Read/Write accessible only when either the STOP or the SPND bit is set.

2 LOOP Loopback Enable allows PCnet-PCI II controller to operate in full-duplex mode for test purposes. The setting of the full-duplex control bits in BCR9 have no effect when the device operates in loopback mode. When LOOP is set to ONE, loopback is enabled. In combination with INTL and MENDECL, various loopback modes are defined in the Loopback Configuration table.

Read/Write accessible only when either the STOP or the SPND bit is set. LOOP is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

1 DTX Disable Transmit. When DTX is set to ONE, the PCnet-PCI II controller will not access the transmit descriptor ring and therefore no transmissions are attempted. When DTX is cleared to ZERO, TXON (CSR0, bit 4) is set to ONE after STRT (CSR0, bit 1) has been set to ONE.

Read/Write accessible only when either the STOP or the SPND bit is set.

0 DRX Disable Receiver. When DRX is set to ONE, the PCnet-PCI II controller will not access the receive descriptor ring and therefore all receive frame data are ignored. When DRX is cleared to ZERO, RXON (CSR0, bit 5) is set to ONE after STRT (CSR0, bit 1) has been set to ONE.

Read/Write accessible only when either the STOP or the SPND bit is set.

CSR16: Initialization Block Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IADRL	This register is an alias of CSR1. Read/Write accessible only when either the STOP or the SPND bit is set.

CSR17: Initialization Block Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IADRH	This register is an alias of CSR2. Read/Write accessible only when either the STOP or the SPND bit is set.

CSR18: Current Receive Buffer Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRBAL	Contains the lower 16 bits of the current receive buffer address at which the PCnet-PCI II controller will store incoming frame data. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR19: Current Receive Buffer Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRBAU	Contains the upper 16 bits of the current receive buffer address at which the PCnet-PCI II controller will store incoming frame data. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR20: Current Transmit Buffer Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXBAL	Contains the lower 16 bits of the current transmit buffer address from which the PCnet-PCI II controller is transmitting. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR21: Current Transmit Buffer Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXBAU	Contains the upper 16 bits of the current transmit buffer address from which the PCnet-PCI II controller is transmitting. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR22: Next Receive Buffer Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRBAL	Contains the lower 16 bits of the next receive buffer address to which the PCnet-PCI II controller will store incoming frame data. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR23: Next Receive Buffer Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRBAU	Contains the upper 16 bits of the next receive buffer address to

which the PCnet-PCI II controller will store incoming frame data.

Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR24: Base Address of Receive Descriptor Ring Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	BADRL	Contains the lower 16 bits of the base address of the receive descriptor ring. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR25: Base Address of Receive Descriptor Ring Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	BADRU	Contains the upper 16 bits of the base address of the receive descriptor ring. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR26: Next Receive Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRDAL	Contains the lower 16 bits of the next receive descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR27: Next Receive Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRDAU	Contains the upper 16 bits of the next receive descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR28: Current Receive Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRDAL	Contains the lower 16 bits of the current receive descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR29: Current Receive Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRDAU	Contains the upper 16 bits of the current receive descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR30: Base Address of Transmit Descriptor Ring Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	BADXL	Contains the lower 16 bits of the base address of the transmit descriptor ring. Read/Write accessible only when either the STOP or the SPND bit

is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR31: Base Address of Transmit Descriptor Ring Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	BADXU	Contains the upper 16 bits of the base address of the transmit descriptor ring. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR32: Next Transmit Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXDAL	Contains the lower 16 bits of the next transmit descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR33: Next Transmit Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXDAU	Contains the upper 16 bits of the next transmit descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR34: Current Transmit Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXDAL	Contains the lower 16 bits of the current transmit descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR35: Current Transmit Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXDAU	Contains the upper 16 bits of the current transmit descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR36: Next Next Receive Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NNRDAL	Contains the lower 16 bits of the next next receive descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR37: Next Next Receive Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NNRDAU	Contains the upper 16 bits of the next next receive descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR38: Next Next Transmit Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NNXDAL	Contains the lower 16 bits of the next next transmit descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR39: Next Next Transmit Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NNXDAU	Contains the upper 16 bits of the next next transmit descriptor address pointer. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR40: Current Receive Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZEROs.
11–0	CRBC	Current Receive Byte Count.This field is a copy of the BCNT field of RMD1 of the current receive descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR41: Current Receive Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRST	Current Receive Status. This field is a copy of bits 31–16 of RMD1 of the current receive descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR42: Current Transmit Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZEROs.
11–0	CXBC	Current Transmit Byte Count. This field is a copy of the BCNT field of TMD1 of the current transmit descriptor.

CSR43: Current Transmit Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXST	Current Transmit Status. This field is a copy of bits 31–16 of TMD1 of the current transmit descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR44: Next Receive Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZEROs.
11–0	NRBC	Next Receive Byte Count. This field is a copy of the BCNT field of RMD1 of the next receive descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR45: Next Receive Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRST	Next Receive Status. This field is a copy of bits 31–16 of RMD1 of the next receive descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR46: Poll Time Counter

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	POLL	Poll Time Counter. This counter is incremented by the PCnetPCI II controller microcode and is used to trigger the descriptor ring polling operation of the PC net-PCI II controller. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR47: Polling Interval

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	POLLINT	Polling Interval. This register contains the time that the PCnet-PCI II controller will wait between successive polling operations. The POLLINT value is expressed as the two's complement of the desired interval, where each bit of POLLINT represents one clock period. POLLINT[3:0] are ignored. The sign of the two's complement POLLINT value is implied to be a one, so POLLINT[15] does not represent the sign bit, but is the MSB of the number. The default value of this register is 0000h. This corresponds to a polling interval of 65,536 clock periods (1.966 ms when CLK = 33 MHz). The POLLINT value of 0000h is created during the microcode initialization routine, and therefore might not be seen when reading CSR47 after H_RESET or S_RESET. If the user desires to program a value for POLLINT other than the default, the correct procedure is to first set only INIT in CSR0. When the initialization sequence is complete, the user must set STOP (CSR0, bit 2) or SPND (CSR5, bit 0). Then the user may write to CSR47 and then set STRT in CSR0. In this way, the default value of 0000h in CSR47 will be overwritten with the desired user value. If the user does not use the standard initialization procedure (standard implies use of an initialization block in memory and setting the INIT bit of CSR0), but instead chooses to write directly to each of the registers that are involved in the INIT operation, it is imperative that the user also write to CSR47 as part of the alternative initialization sequence. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR58: Software Style

Bit	Name	Description
		This register is an alias of the location BCR20. Accesses to/from this register are equivalent to accesses to BCR20.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–11	RES	Reserved locations. Written as ZEROs and read as undefined.
10	APERREN	Advanced Parity Error Handling Enable. When APERREN is set to ONE, the BPE bits (RMD1 and TMD1, bit 23) are used to indicated parity error in data transfers to the receive and transmit buffers. Note that since the advanced

parity error handling uses an additional bit in the descriptor, SWSTYLE (bits 7–0 of this register) must be set to ONE, TWO or THREE to program the PCnet-PCI II controller to use 32-bit software structures.

APERREN does not affect the reporting of address parity errors or data parity errors that occur when the PCnet-PCI II controller is the target of the transfer.

Read accessible always, write accessible only when either the STOP or the SPND bit is set. APERREN is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

9 CSRPCNET CSR PCnet-ISA configuration. When set, this bit indicates that the PCnet-PCI II controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the PCnet-ISA (Am79C960) device. When cleared, this bit indicates that PCnet-PCI II controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the ILACC (Am79C900) device.

The value of CSRPCNET is determined by the PCnet-PCI II controller according to the setting of the Software Style (SWSTYLE, bits 7–0 of this register).

Read accessible always. CSRPCNET is read only. Write operations will be ignored. H_RESET (since SWSTYLE defaults to ZERO) and is not affected by S_RESET or by setting the STOP bit.

8 SSIZE32 32-Bit Software Size. When set, this bit indicates that the PCnetPCI II controller utilizes 32-bit software structures for the initialization block and the transmit and receive descriptor entries. When cleared, this bit indicates that the PCnet-PCI II controller utilizes 16-bit software structures for the initialization block and the transmit and receive descriptor entries. In this mode the PCnet-PCI II controller is backwards compatible with the Am79C90 C-LANCE and Am79C960 PCnet-ISA.

The value of SSIZE32 is determined by the PCnet-PCI II controller according to the setting of the Software Style (SWSTYLE, bits 7–0 of this register).

Read accessible always. SSIZE32 is read only. Write operations will be ignored. SSIZE32 will be cleared after H_RESET (since SWSTYLE defaults to ZERO) and is not affected by S_RESET or by setting the STOP bit.

If SSIZE32 is cleared to ZERO, then bits IADR[31:24] of CSR2 will be used to generate values for the upper 8 bits of the 32 bit address bus during master accesses initiated by the PCnetPCI II controller. This action is required, since the 16-bit software structures will yield only 24 bits of address for PCnet-PCI II controller bus master accesses.

If SSIZE32 is set to ONE, then the software structures that are common to the PCnet-PCI II controller and the host system will supply a full 32 bits for each address pointer that is needed by the PCnet-PCI II controller for performing master accesses.

The value of the SSIZE32 bit has no effect on the drive of the upper 8 address bits. The upper 8 address pins are always driven, regardless of the state of the SSIZE32 bit.

Note that the setting of the SSIZE32 bit has no effect on the width for I/O accesses. I/O access width is determined by the state of the DWIO bit (BCR18, bit 7).

7–0 SWSTYLE Software Style register. The value in this register determines the style of register and memory resources that shall be used by the PCnet-PCI II controller. The Software Style selection will affect the interpretation of a few bits within the CSR space, the order of the descriptor entries and the width of the descriptors and initialization block entries.

All PCnet-PCI II controller CSR bits and BCR bits and all descriptor, buffer and initialization block entries not cited in the table below are unaffected by the software style selection.

Read/Write accessible only when either the STOP or the SPND bit is set. The SWSTYLE register will contain the value 00h following H_RESET and will be unaffected by S_RESET or by setting the STOP bit.

Table 24. Software Styles

SWSTYLE [7:0]	Style Name	CSRPCNET	SSIZE32	Initialization Block Entries	Descriptor Ring Entries	Altered Bit Interpretations
00h	C-LANCE / PCnet-ISA	1	0	16-bit software structures, non-burst or burst access	16-bit software structures, non-burst access only	All bits in CSR4 are used, TMD1[29] is ADD_FCS
01h	ILACC	0	1	32-bit software structures, non-burst or burst access	32-bit software access structures, non-burst access only	CSR4[9:8], CSR4[5:4] and CSR4[1:0] have no function, TMD1[29] is NO_FCS.
02h	PCnet-PCI II	1	1	32-bt software structures, non-burst or burst access	32-bit software structures, non-burst access only	All bits in CSR4 are used, TMD1[29] is ADD_FCS
03h	PCnet-PCI II controller	1	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst or burst access	All bits in CSR4 are used, TMD1[29] is ADD_FCS
All Other	Reserved	Undefined	Undefined	Undefined	Undefined	Undefined

CSR60: Previous Transmit Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PXDAL	Contains the lower 16 bits of the previous transmit descriptor address pointer. The PCnet-PCI II controller can stack multiple transmit frames. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR61: Previous Transmit Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PXDAU	Contains the upper 16 bits of the previous transmit descriptor address pointer. The PCnet-PCI II controller can stack multiple transmit frames. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR62: Previous Transmit Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations.
11–0	PXBC	Previous Transmit Byte Count. This field is a copy of the BCNT field of TMD1 of the previous transmit descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR63: Previous Transmit Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PXST	Previous Transmit Status. This field is a copy of bits 31–16 of TMD1 of the previous transmit descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR64: Next Transmit Buffer Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXBAL	Contains the lower 16 bits of the next transmit buffer address from which the PCnet-PCI II controller will transmit an outgoing frame. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR65: Next Transmit Buffer Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXBAU	Contains the upper 16 bits of the next transmit buffer address from which the PCnet-PCI II controller will transmit an outgoing frame. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR66: Next Transmit Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZEROs.
11–0	NXBC	Next Transmit Byte Count. This field is a copy of the BCNT field of TMD1 of the next transmit descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR67: Next Transmit Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXST	Next Transmit Status. This field is a copy of bits 31–16 of TMD1 of the next transmit descriptor. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.
7–0	RES	Reserved locations. Read and written as ZEROs. Accessible only when either the STOP or the SPND bit is set.

CSR72: Receive Descriptor Ring Counter

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	RCVRC	Receive Descriptor Ring Counter location. Contains a two's complement binary number used to number the current receive descriptor. This counter interprets the value in CSR76 as pointing to the first descriptor. A counter value of ZERO corresponds to the last descriptor in the ring. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR74: Transmit Descriptor Ring Counter

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.

15–0 XMTRC Transmit Descriptor Ring Counter location. Contains a two's complement binary number used to number the current transmit descriptor. This counter interprets the value in CSR78 as pointing to the first descriptor. A counter value of ZERO corresponds to the last descriptor in the ring.

Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR76: Receive Descriptor Ring Length

Bit Name Description

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15–0 RCVRL Receive Descriptor Ring Length. Contains the two's complement of the receive descriptor ring length. This register is initialized during the PCnet-PCI II controller initialization routine based on the value in the RLEN field of the initialization block. However, the ring length can be programmed to any value from 1 to 65535 by writing directly to this register.

Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR78: Transmit Descriptor Ring Length

Bit Name Description

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15–0 XMTRL Transmit Descriptor Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the PCnetPCI II controller initialization routine based on the value in the TLEN field of the initialization block. However, the ring length can be programmed to any value from 1 to 65535 by writing directly to this register.

Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR80: DMA Transfer Counter and FIFO Watermark Control

Bit Name Description

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15–14 RES Reserved locations. Read as ONEs and written as ZEROs. Accessible only when either the STOP or the SPND bit is set.

13–12 RCVFW[1:0] Receive FIFO Watermark. RCVFW specifies the number of bytes which must be present in the receive FIFO (once the frame has been verified as a non-runt) before receive DMA is requested. If the network interface is operating in half-duplex mode, at least 64 bytes or a complete frame must be received in order for a receive DMA to start. This effectively avoids having to react to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature is enabled or if the network interface is operating in full-duplex mode, receive DMA will be requested as soon as either the Receive FIFO Watermark is reached, or a complete valid receive frame is detected (regardless of length). If the EADI interface is active and the Runt Packet Accept feature is enabled or the network interface is operating in full-duplex mode, RCVFW must not be programmed to 00b to allow enough time to reject the frame.

Table 25. Receive Watermark Programming

RCVFW[1:0]	Bytes Received
00	16
01	64
10	128
11	Reserved

Read/Write accessible only when either the STOP or the SPND bit is set. RCVFW is set to a value of 01b (64 bytes) after H_RESET or S_RESET and is unaffected by setting the STOP bit.

11–10 XMTSP[1:0] Transmit Start Point. As soon as the number of bytes in the transmit FIFO reaches the XMTSP value, the PCnet-PCI II controller starts trying to transmit. When the entire frame is in the FIFO, transmission attempts will start regardless of the value in XMTSP. If the network interface is operating in half-duplex mode, regardless of XMTSP, the FIFO will not internally overwrite its data until at least 64 bytes (or the entire frame if shorter than 64 bytes) have been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be reloaded into the transmit FIFO, and retries will be handled autonomously by the MAC. If the Disable Retry feature is enabled, or if the network is operating in full-duplex mode, the PCnet-PCI II controller can overwrite the beginning of the frame as soon as the data is transmitted, because no collision handling is required in these modes.

Table 26. Transmit Start Point Programming

XMTSP[1:0]	Bytes Written
00	8
01	64
10	128
11	248

Read/Write accessible only when either the STOP or the SPND bit is set. XMTSP is set to a value of 01b (64 bytes) after H_RESET or S_RESET and is unaffected by setting the STOP bit.

9–8 XMTFW[1:0] Transmit FIFO Watermark. XMTFW controls the point at which transmit DMA is requested. Transmit DMA is requested when the number of bytes specified by XMTFW can be written to the transmit FIFO.

Table 27. Transmit Watermark Programming

XMTFW[1:0]	Byte Spaces Available
00	16
01	64
10	128
11	Reserved

Read/Write accessible only when either the STOP or the SPND bit is set. XMTFW is set to a value of 00b (16 bytes) after H_RESET or S_RESET and is unaffected by setting the STOP bit.

7–0 DMATC[7:0] DMA Transfer Counter. If DMAPLUS (CSR4, bit 14) is cleared to ZERO, this counter contains the maximum number of FIFO read or write data phases the PCnet-PCI II controller will perform during a single bus mastership period, if not preempted. The DMA Transfer Counter is not used to limit the number of data phases during initialization block or descriptor transfers. A value of ZERO will be interpreted as one data phase. If DMAPLUS is set to ONE, the DMA Transfer Counter is disabled, and the PCnet-PCI II controller will try to transfer data as long as the transmit FIFO is not full or as long as the receive FIFO is not empty.

When the PCnet-PCI II controller is preempted and the last data phase has finished, DMATC will freeze. It will continue counting down when the PCnet-PCI II controller is granted bus ownership again and continues with the data transfers.

DMATC should not be enabled when the PCnet-PCI II controller is used in a PCI bus application. The PCI Latency Timer should be the only entity governing the time the PCnet-PCI II controller has control over the bus.

Read/Write accessible only when either the STOP or the SPND bit is set. Note that the read operation will yield the value of the run-time copy of the DMA Transfer Counter and not the register that holds the programmed value. Most read operations will yield a value of ZERO, because the run-time counter is only reloaded with the programmed value at the beginning of a new bus mastership period. The DMA Transfer Counter is set to a value of 16 (10h) after H_RESET or S_RESET and is unaffected by setting the STOP bit.

CSR82: Bus Activity Timer

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	DMABAT	Bus Activity Timer. If TIMER (CSR4, bit 13) is set to ONE, this register controls the maximum allowable time that PCnet-PCI II controller will take up on the system bus during FIFO data transfers. The Bus Activity Timer does not limit the time on the system bus during initialization block or descriptor transfers. The DMABAT value is interpreted as an unsigned number with a resolution of 0.1 µs. For instance, a value of 51 µs would be programmed with a value of 510 (1FEh). A value of ZERO (the default value) will result in a single data transfer. DMABAT starts counting down when the PCnet-PCI II controller is granted bus ownership and the bus is idle. When DMABAT has counted down to ZERO, the PCnet-PCI II controller will finish the current data phase before releasing the bus. Note that because DMABAT does not run on the PCI bus interface clock, the actual time the PCnet-PCI II controller takes up the bus might differ by 2 to 3 clock periods from the value programmed to DMABAT. DMABAT should not be enabled when the PCnet-PCI II controller is used in a PCI bus application. The PCI Latency Timer should be the only entity governing the time the PCnet-PCI II controller has control over the bus. Read/Write accessible only when either the STOP or the SPND bit is set. Note that the read operation will yield the value of the run-time copy of the Bus Activity Timer and not the register that holds the programmed value. Most read operations will yield a value of ZERO, because the run-time counter is only reloaded with the programmed value at the beginning of a new bus mastership period. The Bus Activity Timer register is cleared to a value of 0000h after H_RESET or S_RESET and is unaffected by setting the STOP bit.

CSR84: DMA Address Register Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	DMABAL	This register contains the lower 16 bits of the address of system memory for the current DMA cycle. The Bus Interface Unit controls the Address Register by issuing commands to increment the memory address for sequential operations. The DMABAL register is undefined until the first PCnet-PCI II controller DMA operation. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR85: DMA Address Register Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	DMABAU	This register contains the upper 16 bits of the address of system memory for the current DMA cycle. The Bus Interface Unit controls the Address Register by issuing commands to increment the memory address for sequential operations. The DMABAU

register is undefined until the first PCnet-PCI II controller DMA operation.

Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR86: Buffer Byte Counter

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written with ONEs.
11–0	DMABC	DMA Byte Count Register. Contains the two's complement of the remaining size of the current transmit or receive buffer in bytes. This register is incremented by the Bus Interface Unit. The DMABC register is undefined until written. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR88: Chip ID Register Lower

Bit	Name	Description
31 – 28	VER	Version. This 4-bit pattern is silicon revision dependent. Read accessible always. VER is read only. Write operations are ignored.
27 – 12	PARTID	Part number. The 16-bit code for the PCnet-PCI II controller is 0010 0110 0010 0001b (2621h). This register is exactly the same as the Device ID register in the JTAG description. It is, however, different from the ID stored in the Device ID register in the PCI configuration space. Read accessible only when either the STOP or the SPND bit is set. PARTID is read only. Write operations are ignored.
11 – 1	MANFID	Manufacturer ID. The 11-bit manufacturer code for AMD is 00000000001b. This code is per the JEDEC Publication 106-A. Note that this code is not the same as the Vendor ID in the PCI configuration space. Read accessible always. MANFID is read only. Write operations are ignored.
0	ONE	Always a logic ONE. Read accessible always. ONE is read only. Write operations are ignored.

CSR89: Chip ID Register Upper

Bit	Name	Description
31 – 16	RES	Reserved locations. Read as undefined.
15 – 12	VER	Version. This 4-bit pattern is silicon-revision dependent. Read accessible always. VER is read only. Write operations are ignored.
11 – 0	PARTIDU	Upper 12 bits of the PCnet-PCI II controller part number. I.e. 0010 0110 0010b. Read accessible always. PARTIDU is read only. Write operations are ignored.

CSR94: Transmit Time Domain Reflectometry Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–10	RES	Reserved locations. Read and written as ZEROs.
9–0	XMTTDR	Time Domain Reflectometry reflects the state of an internal counter that counts from the start of transmission to the occurrence of loss of carrier. TDR is incremented at a rate of 10 MHz. Read accessible only when either the STOP or the SPND bit is set. Write operations are ignored. XMTTDR is cleared by H_RESET or S_RESET.

CSR100: Bus Timeout

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	MERRTO	This register contains the value of the longest allowable bus latency (interval between assertion of $\overline{\text{REQ}}$ and assertion of $\overline{\text{GNT}}$) that a system may insert into a PCnet-PCI II controller master transfer. If this value of bus latency is exceeded, then MERR (CSR0, bit 11) will be set to ONE, and an interrupt may be generated, depending upon the setting of the MERRM bit (CSR3, bit 11) and the IENA bit (CSR0, bit 6). The value in this register is interpreted as the unsigned number of XTAL1 clock periods divided by two, i.e. the value in this register is given in 0.1 µs increments. For example, the value 0600h (1536 decimal) will cause a MERR to be indicated after 153.6 µs of bus latency. A value of ZERO will allow an infinitely long bus latency, i.e. bus timeout error will never occur. Read/Write accessible only when either the STOP or the SPND bit is set. This register is set to 0600h by H_RESET or S_RESET and is unaffected by setting the STOP bit.

CSR112: Missed Frame Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	MFC	Missed Frame Count. Indicates the number of missed frames. MFC will roll over to a count of ZERO from the value 65535. The MFCO bit (CSR4, bit 8) will be set each time that this occurs. The PCnet-PCI II controller will not count missed frames while the device is in suspend mode (SPND = 1, CSR5, bit 0). Read accessible always. MFC is read only, write operations are ignored. MFC is cleared by H_RESET or S_RESET or by setting the STOP bit.

CSR114: Receive Collision Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	RCC	Receive Collision Count. Indicates the total number of collisions on the network encountered by the receiver since the last reset of the counter. RCC will roll over to a count of ZERO from the value 65535. The RCVCCO bit of CSR4 (bit 5) will be set each time that this occurs. The PCnet-PCI II controller will continue counting collisions on the network while the device is in suspend mode (SPND = 1, CSR5, bit 0) Read accessible always. RCC is read only, write operations are ignored. RCC is cleared by H_RESET or S_RESET or by setting the STOP bit.

CSR122: Advanced Feature Control

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–2	RES	Reserved locations. Written as ZEROs and read as undefined.
0	RCVALGN	Receive Frame Align. When set, this bit forces the data field of ISO 8802-3 (IEEE/ANSI 802.3) frames to align to DWord address boundaries. It is important to note that this feature will only function correctly if all receive buffer boundaries are DWord aligned and all receive buffers have 0 MOD 4 lengths. In order to accomplish the data alignment, the PCnet-PCI II controller simply inserts two bytes of random data at the beginning of the receive frame (i.e. before the ISO 8802-3 (IEEE/ANSI 802.3) destination address field). The MCNT field reported to the receive descriptor

will not include the extra two bytes.

Read/Write accessible always. RCVALGN is cleared by H_RESET or S_RESET and is not affected by STOP.

CSR124: Test Register 1

Bit	Name	Description
		This register is used to place the PCnet-PCI II controller into various test modes. Only Runt Packet Accept is user accessible test modes. All other test modes are for AMD internal use only.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–5	RES	Reserved locations. Written as ZEROs and read as undefined.
4	RES	Reserved locations. Written as ZEROs and read as undefined.
3	RPA	Runt Packet Accept. This bit forces the PCnet-PCI II controller to accept runt packets (packets shorter than 64 bytes). Read accessible always. Write accessible when EN124 (CSR4, bit 15) is set to ONE. RPA is cleared by H_RESET or S_RESET and is not affected by setting the STOP bit.
2–0	RES	Reserved locations. Written as ZEROs and read as undefined.

Bus Configuration Registers

The Bus Configuration Registers (BCRs) are used to program the configuration of the bus interface and other special features of the PCnet-PCI II controller that are not related to the IEEE 8802-3 MAC functions. The BCRs are accessed by first setting the appropriate RAP value, and then by performing a slave access to the BDP.

All BCR registers are 16 bits wide in Word I/O mode (DWIO = 0, BCR18, bit 7) and 32 bits wide in DWord I/O mode (DWIO = 1). The upper 16 bits of all BCR registers are undefined when in DWord I/O mode. These bits should be written as ZEROs and should be treated as undefined when read. The default value given for any BCR is the value in the register after H_RESET. Some of these values may be changed shortly after H_RESET when the contents of the external EEPROM is automatically read in. With the exception of DWIO (BCR18, bit 7) BCR register values are not affected by S_RESET. None of the BCR register values are affected by the assertion of the STOP bit.

Note that several registers have no default value. BCR0, BCR1, BCR3, BCR8, BCR10–17 and BCR21 are reserved and have undefined values. The content of BCR2 is undefined until is has been first programmed through the EEPROM read operation or a user register write operation.

BCR0, BCR1, BCR16, BCR17 and BCR21 are registers that are used by other devices in the PCnet family. Writing to these registers has no effect on the operation of the PCnet-PCI II controller.

Writes to those registers marked as Reserved will have no effect. Reads from these locations will produce undefined values.

Table 28. BCR Registers

				Programmability	
RAP	MNEMONIC	Default	Name	User	EEPROM
0	MSRDA	0005h	Reserved	No	No
1	MSWRA	0005h	Reserved	No	No
2	MC	0002h	Miscellaneous Configuration	Yes	Yes
3	Reserved	N/A	Reserved	No	No
4	LNKST	00C0h	Link Status LED	Yes	Yes
5	LED1	0084h	LED1 Status	Yes	Yes
6	LED2	0088h	LED2 Status	Yes	Yes
7	LED3	0090h	LED3 Status	Yes	Yes
8	Reserved	N/A	Reserved	No	No
9	FDC	0000h	Full-Duplex Control	Yes	Yes
10–15	Reserved	N/A	Reserved	No	No
16	IOBASEL	N/A	Reserved	No	No
17	IOBASEU	N/A	Reserved	No	No
18	BSBC	9001h	Burst and Bus Control	Yes	Yes
19	EECAS	0002h	EEPROM Control and Status	Yes	No
20	SWS	0000h	Software Style	Yes	No
21	INTCON	N/A	Reserved	No	No
22	PCILAT	FF06h	PCI Latency	Yes	Yes

BCR0: Master Mode Read Active

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	MSRDA	Reserved locations. After H_RESET, the value in this register will be 0005h. The setting of this register has no effect on any PCnet-PCI II controller function. It is only included for software compatibility with other PCnet family devices. Read always. MSRDA is read only Write operations have no effect.

BCR1: Master Mode Write Active

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	MSWRA	Reserved locations. After H_RESET, the value in this register will be 0005h. The setting of this register has no effect on any PCnet-PCI II controller function. It is only included for software compatibility with other PCnet family devices. Read always. MSWRA is read only Write operations have no effect.

BCR2: Miscellaneous Configuration

Note that bits 15–0 in this register are programmable through the external EEPROM. Reserved bits and read-only bits should be programmed to ZERO.

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	RES	Reserved location. Written as ZERO and read as undefined.
14	TMAULOOP	When set, this bit allows external loopback packets to pass on to the network through the T-MAU interface, if the T-MAU interface has been selected. If the T-MAU interface has not been selected, then this bit has no effect. Read/Write accessible always. TMAULOOP is cleared to ZERO by H_RESET and is unaffected

by S_RESET or by setting the STOP bit.

13–9 RES Reserved locations. Written as ZEROs and read as undefined.

8 APROMWE Address PROM Write Enable. The PCnet-PCI II controller contains a shadow RAM on board for storage of the first 16 bytes loaded from the serial EEPROM. Accesses to Address PROM I/O Resources will be directed toward this RAM. When APROMWE is set to ONE, then write access to the shadow RAM will be enabled.

Read/Write accessible always. APROMWE is cleared to ZERO by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

7 INTLEVEL Interrupt Level. This bit allows the interrupt output signals to be programmed for level or edge-sensitive applications.

When INTLEVEL is cleared to ZERO, the $\overline{\text{INTA}}$ pin is configured for level-sensitive applications. In this mode, an interrupt request is signaled by a low level driven on the $\overline{\text{INTA}}$ pin by the PCnet-PCI II controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is tristated by the PCnet-PCI II controller and allowed to be pulled to a high level by an external pullup device. This mode is intended for systems which allow the interrupt signal to be shared by multiple devices.

When INTLEVEL is set to ONE, the $\overline{\text{INTA}}$ pin is configured for edge-sensitive applications. In this mode, an interrupt request is signaled by a high level driven on the $\overline{\text{INTA}}$ pin by the PCnet-PCI II controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is driven to a low level by the PCnet-PCI II controller. This mode is intended for systems that do not allow interrupt channels to be shared by multiple devices.

INTLEVEL should not be set to ONE when the PCnet-PCI II controller is used in a PCI bus application.

Read/Write accessible always. INTLEVEL is cleared to ZERO by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

6 RES Reserved location. Written as ZERO and read as undefined.

5 DXCVRCTL DXCVR Control. When the AUI is the active network port, DXCVRCTL controls the assertion of the DXCVR output. The polarity of the asserted state is controlled by the DXCVRPOL bit (BCR2, bit 4). The DXCVR pin can be used to control a DC-to-DC converter in applications that want to connect a 10BASE2 MAU as well as a standard DB15 AUI connector to the PCnet-PCI II controller AUI. When DXCVRCTL is set to ONE, the DXCVR output will be asserted. This could be used to enable a DC-to-DC converter for 10BASE2 MAUs (assuming the enable input of the DC-to-DC converter is active high and DXCVRPOL is cleared to ZERO). When DXCVRCTL is cleared to ZERO, the DXCVR output will be deasserted. This would power down the DC-to-DC converter. When the 10BASE-T interface is the active network port, the DXCVR output is always deasserted.

Read/Write accessible always. DXCVRCTL is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

4 DXCVRPOL DXCVR Polarity. This bit controls the polarity of the asserted state of the DXCVR output. When DXCVRPOL is cleared to ZERO, the DXCVR output will be HIGH when asserted. When DXCVRPOL is set to ONE, the DXCVR output will be LOW when asserted.

Table 29. DXCVR Output Control

DXCVRCTL	DXCVRPOL	Active Network Port	DXCVR Output
X	0	10BASE-T	Low
X	1	10BASE-T	High
0	0	AUI	Low
1	0	AUI	High
0	1	AUI	High
1	1	AUI	Low

Read/Write accessible always. DXCVRPOL is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

3 EADISEL EADI Select. When set to ONE, this bit enables the three EADI interface pins that are multiplexed with other functions. EESK/$\overline{\text{LED1}}$ becomes SFBD, EEDO/$\overline{\text{LED3}}$ becomes SRD, and $\overline{\text{LED2}}$ becomes SRDCLK.

Read/Write accessible always. EADISEL is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

2 AWAKE This bit selects one of two different sleep modes.

If AWAKE is set to ONE and the $\overline{\text{SLEEP}}$ pin is asserted, the PCnet-PCI II controller goes into snooze mode. If AWAKE is cleared to ZERO and the $\overline{\text{SLEEP}}$ pin is asserted, the PCnet-PCI II controller goes into coma mode. See the section “Power Saving Modes” for more details.

This bit only has meaning when the 10BASE-T network interface is selected. Read/Write accessible always. AWAKE is cleared to ZERO by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

1 ASEL Auto Select. When set, the PCnet-PCI II controller will automatically select the operating media interface port.

If ASEL has been set to ONE, and the 10BASE-T transceiver is in the Link Pass state, the 10BASE-T port will be used. If ASEL has been set to ONE, and the 10BASE-T port is in the Link Fail state, the AUI port will be used. If one of the above conditions changes during transmission, switching between the ports will not occur until the transmission is ended.

When ASEL is set to ONE, Link Beat Pulses will be transmitted on the 10BASE-T port, regardless of the state of Link Status. When ASEL is cleared to ZERO, Link Beat Pulses will only be transmitted on the 10BASE-T port when the PORTSEL bits of the Mode Register (CSR15) have selected 10BASE-T as the active port.

When ASEL is cleared to ZERO, then the selected network port will be determined by the settings of the PORTSEL bits of CSR15.

Read/Write accessible always. ASEL is set to ONE by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

The network port configurations are as follows:

Table 30. Network Port Configuration

PORTSEL[1:0]	ASEL (BCR2[1])	Link Status (of 10BASE-T)	Network Port
0X	1	Fail	AUI
0X	1	Pass	10BASE-T
00	0	X	AUI
01	0	X	10BASE-T
10	X	X	Reserved
11	X	X	Reserved

0 XMAUSEL Reserved location. Read/Write accessible always. This reserved location is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit. Writing a ONE to this bit has no effect on the operation of the PCnet-PCI II controller.

BCR4: Link Status LED (LNKST)

Bit **Name** **Description**

BCR4 determines which function(s) activate the $\overline{\text{LNKST}}$ pin. The pin will indicate the logical OR of the enabled functions. BCR4 defaults to Link Status (LNKST) with pulse stretcher enabled (PSE = 1).

Note that bits 15–0 in this register are programmable through the external EEPROM. Reserved bits and read-only bits should be programmed to ZERO.

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15 LEDOUT This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true.

The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of this register (bits 8 and 6–0).

Read accessible always. This bit is read only. Writes have no effect. LEDOUT is unaffected by H_RESET, S_RESET or by setting the STOP bit.

14 LEDPOL LED Polarity. When this bit has the value ZERO, the LED pin will be asserted LOW whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float whenever the OR of the enabled signals is false. (The LED output will be an open drain output, and the output value will be the inverse of the LEDOUT status bit.)

When this bit has the value ONE, the LED pin will be asserted HIGH whenever the OR of the enabled signals is true, and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false. (The LED output will be a totem pole output, and the output value will be the same polarity as the LEDOUT status bit.)

The setting of this bit will not effect the polarity of the LEDOUT bit for this register.

Read/Write accessible always. LEDPOL is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

13 LEDDIS LED Disable. This bit is used to disable the LED output. When LEDDIS is set to ONE and LEDPOL is cleared to ZERO, the LED output pin will be floating. When LEDDIS is set to ONE and LEDPOL is set to ONE, the LED output pin will be driven LOW. When LEDDIS has the value ZERO, the LED output value will be governed by the LEDOUT and LEDPOL values.

Read/Write accessible always. LEDDIS is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

12–10 RES Reserved locations. Written as ZEROs and read as undefined.

9 MPSE Magic Packet Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when magic packet mode is enabled and a magic packet is detected on the network.

Read/Write accessible always. MPSE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

8 FDLSE Full-duplex Link Status Enable. Indicates the full-duplex Link Test Status. When this bit is set to ONE, a value of ONE is passed to the LEDOUT signal when the PCnet-PCI II controller is functioning in a Link Pass state and full-duplex operation is enabled. When the PCnet-PCI II controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of ZERO is passed to the LEDOUT signal.

When the 10BASE-T port is active, a value of ONE is passed to the LEDOUT signal whenever the Link Test Function detects a Link Pass state and the FDEN (BCR9, bit 0) bit is set. When the AUI port is active, a value of ONE is passed to the LEDOUT signal whenever full-duplex operation on the AUI port is enabled (both FDEN and AUIFD bits in BCR9 are set to ONE).

Read/Write accessible always. FDLSE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

7 PSE Pulse Stretcher Enable. When this bit is set to ONE, the LED illumination time is extended so that brief occurrences of the enabled function will be seen on this LED output. A value of ZERO disables the pulse stretcher.

Read/Write accessible always. PSE is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.

6 LNKSTE Link Status Enable. When this bit is set to ONE, a value of ONE will be passed to the LEDOUT bit in this register when the T-MAU operating in half-duplex mode is in Link Pass state. When the T-MAU operating in half-duplex mode is in Link Fail state, a value of ZERO is passed to the LEDOUT bit.

The function of this bit is masked if the 10BASE-T port is operating in full-duplex mode. This allows a system to have separate LEDs for half-duplex Link Status and for full-duplex Link Status.

Read/Write accessible always. LNKSTE is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.

5 RCVME Receive Match Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the there is receive activity on the network that has passed the address match function for this node. All address matching modes are included: physical, logical filtering, broadcast and promiscuous.

Read/Write accessible always. RCVME is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

4 XMTE Transmit Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is transmit activity on the network.

Read/Write accessible always. XMTE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

3 RXPOLE Receive Polarity Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the polarity of the RXD± pair is not reversed.

Receive polarity indication is valid only if the T-MAU is in link pass state.

Read/Write accessible always. RXPOLE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

2 RCVE Receive Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is receive activity on the network.

Read/Write accessible always. RCVE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

1 JABE Jabber status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the PCnet-PCI II controller is jabbering on the network.

Read/Write accessible always. JABE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

0 COLE Collision status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is collision activity on the network. The activity on the collision inputs to the AUI port within the first 4 µs after every transmission for the purpose of SQE testing will not cause the LEDOUT bit to be set.

Read/Write accessible always. COLE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR5: LED1 Status

Bit	Name	Description
		BCR5 determines which function(s) activate the $\overline{\text{LED1}}$ pin. The pin will indicate the logical OR of the enabled functions. BCR5 defaults to Receive Status (RCV) with pulse stretcher enabled (PSE = 1). Note that bits 15–0 in this register are programmable through the external EEPROM. Reserved bits and read-only bits should be programmed to ZERO.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of this register (bits 8 and 6–0). Read accessible always. This bit is read only. Writes have no effect. LEDOUT is unaffected by H_RESET, S_RESET or by setting the STOP bit.
14	LEDPOL	LED Polarity. When this bit has the value ZERO, the LED pin will be asserted LOW whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float whenever the OR of the enabled signals is false. (The LED output will be an open drain output, and the output value will be the inverse of the LEDOUT status bit.) When this bit has the value ONE, the LED pin will be asserted HIGH whenever the OR of the enabled signals is true, and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false. (The LED output will be a totem pole output, and the output value will be the same polarity as the LEDOUT status bit.) The setting of this bit will not effect the polarity of the LEDOUT bit for this register. Read/Write accessible always. LEDPOL is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS is set to ONE and LEDPOL is cleared to ZERO, the LED output pin will be floating. When LEDDIS is set to ONE and LEDPOL is set to ONE, the LED output pin will be driven LOW. When LEDDIS has the value ZERO, the LED output value will be governed by the LEDOUT and LEDPOL values. Read/Write accessible always. LEDDIS is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
12–10	RES	Reserved locations. Written as ZEROs and read as undefined.
9	MPSE	Magic Packet Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when magic packet mode is enabled and a magic packet is detected on the network. Read/Write accessible always. MPSE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
8	FDLSE	Full-duplex Link Status Enable. Indicates the full-duplex Link Test Status. When this bit is set to ONE, a value of ONE is passed to the LEDOUT signal when the PCnet-PCI II controller is functioning in a Link Pass state and full-duplex operation is enabled. When the PCnet-PCI II controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of ZERO is passed to the LEDOUT signal.

When the 10BASE-T port is active, a value of ONE is passed to the LEDOUT signal whenever the Link Test Function detects a Link Pass state and the FDEN (BCR9, bit 0) bit is set. When the AUI port is active, a value of ONE is passed to the LEDOUT signal whenever full-duplex operation on the AUI port is enabled (both FDEN and AUIFD bits in BCR9 are set to ONE).

Read/Write accessible always. FDLSE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

7 PSE Pulse Stretcher Enable. When this bit is set to ONE, the LED illumination time is extended so that brief occurrences of the enabled function will be seen on this LED output. A value of ZERO disables the pulse stretcher.

Read/Write accessible always. PSE is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.

6 LNKSTE Link Status Enable. When this bit is set to ONE, a value of ONE will be passed to the LEDOUT bit in this register when the T-MAU operating in half-duplex mode is in Link Pass state. When the T-MAU operating in half-duplex mode is in Link Fail state, a value of ZERO is passed to the LEDOUT bit.

The function of this bit is masked if the 10BASE-T port is operating in full-duplex mode. This allows a system to have separate LEDs for half-duplex Link Status and for full-duplex Link Status.

Read/Write accessible always. LNKSTE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

5 RCVME Receive Match Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the there is receive activity on the network that has passed the address match function for this node. All address matching modes are included: physical, logical filtering, broadcast and promiscuous.

Read/Write accessible always. RCVME is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

4 XMTE Transmit Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is transmit activity on the network.

Read/Write accessible always. XMTE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

3 RXPOLE Receive Polarity Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the polarity of the RXD± pair is not reversed.

Receive polarity indication is valid only if the T-MAU is in link pass state.

Read/Write accessible always. RXPOLE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

2 RCVE Receive Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is receive activity on the network.

Read/Write accessible always. RCVE is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.

1 JABE Jabber status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the PCnet-PCI II controller is jabbering on the network.

Read/Write accessible always. JABE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

0 COLE Collision status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is collision activity on the network.

The activity on the collision inputs to the AUI port within the first 4 µs after every transmission for the purpose of SQE testing will not cause the LEDOUT bit to be set.

Read/Write accessible always. COLE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR6: LED2 Status

Bit **Name** **Description**

BCR6 determines which function(s) activate the $\overline{\text{LED2}}$ pin. The pin will indicate the logical OR of the enabled functions. BCR6 defaults to Receive Polarity Status (RXPOL) with pulse stretcher enabled (PSE = 1).

Note that bits 15–0 in this register are programmable through the external EEPROM. Reserved bits and read-only bits should be programmed to ZERO.

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15 LEDOUT This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true.

The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of this register (bits 8 and 6–0).

Read accessible always. This bit is read only. Writes have no effect. LEDOUT is unaffected by H_RESET, S_RESET or by setting the STOP bit.

14 LEDPOL LED Polarity. When this bit has the value ZERO, the LED pin will be asserted LOW whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float whenever the OR of the enabled signals is false. (The LED output will be an open drain output, and the output value will be the inverse of the LEDOUT status bit.)

When this bit has the value ONE, the LED pin will be asserted HIGH whenever the OR of the enabled signals is true, and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false. (The LED output will be a totem pole output, and the output value will be the same polarity as the LEDOUT status bit.)

The setting of this bit will not effect the polarity of the LEDOUT bit for this register.

Read/Write accessible always. LEDPOL is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

13 LEDDIS LED Disable. This bit is used to disable the LED output. When LEDDIS is set to ONE and LEDPOL is cleared to ZERO, the LED output pin will be floating. When LEDDIS is set to ONE and LEDPOL is set to ONE, the LED output pin will be driven LOW. When LEDDIS has the value ZERO, the LED output value will be governed by the LEDOUT and LEDPOL values.

Read/Write accessible always. LEDDIS is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

12–10 RES Reserved locations. Written as ZEROs and read as undefined.

9 MPSE Magic Packet Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when magic packet mode is enabled and a magic packet is detected on the network.

Read/Write accessible always. MPSE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

8 FDLSE Full-duplex Link Status Enable. Indicates the full-duplex Link Test Status. When this bit is set to ONE, a value of ONE is passed to the LEDOUT signal when the PCnet-PCI II controller is functioning in a Link Pass state and full-duplex operation is enabled.

		When the PCnet-PCI II controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of ZERO is passed to the LEDOUT signal. When the 10BASE-T port is active, a value of ONE is passed to the LEDOUT signal whenever the Link Test Function detects a Link Pass state and the FDEN (BCR9, bit 0) bit is set. When the AUI port is active, a value of ONE is passed to the LEDOUT signal whenever full-duplex operation on the AUI port is enabled (both FDEN and AUIFD bits in BCR9 are set to ONE). Read/Write accessible always. FDLSE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
7	PSE	Pulse Stretcher Enable. When this bit is set to ONE, the LED illumination time is extended so that brief occurrences of the enabled function will be seen on this LED output. A value of ZERO disables the pulse stretcher. Read/Write accessible always. PSE is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.
6	LNKSTE	Link Status Enable. When this bit is set to ONE, a value of ONE will be passed to the LEDOUT bit in this register when the T-MAU operating in half-duplex mode is in Link Pass state. When the T-MAU operating in half-duplex mode is in Link Fail state, a value of ZERO is passed to the LEDOUT bit. The function of this bit is masked if the 10BASE-T port is operating in full-duplex mode. This allows a system to have separate LEDs for half-duplex Link Status and for full-duplex Link Status. Read/Write accessible always. LNKSTE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
5	RCVME	Receive Match Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the there is receive activity on the network that has passed the address match function for this node. All address matching modes are included: physical, logical filtering, broadcast and promiscuous. Read/Write accessible always. RCVME is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
4	XMTE	Transmit Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is transmit activity on the network. Read/Write accessible always. XMTE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
3	RXPOLE	Receive Polarity Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the polarity of the RXD± pair is not reversed. Receive polarity indication is valid only if the T-MAU is in link pass state. Read/Write accessible always. RXPOLE is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.
2	RCVE	Receive Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is receive activity on the network. Read/Write accessible always. RCVE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
1	JABE	Jabber status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the PCnet-PCI II controller is jabbering on the network. Read/Write accessible always. JABE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

0 COLE Collision status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is collision activity on the network. The activity on the collision inputs to the AUI port within the first 4 µs after every transmission for the purpose of SQE testing will not cause the LEDOUT bit to be set.

Read/Write accessible always. COLE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR7: LED3 Status

Bit Name Description

BCR7 determines which function(s) activate the LED3 pin. The pin will indicate the logical OR of the enabled functions. BCR7 defaults to Transmit Status (XMT) with pulse stretcher enabled (PSE = 1).

Note that bits 15–0 in this register are programmable through the external EEPROM. Reserved bits and read-only bits should be programmed to ZERO.

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15 LEDOUT This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true.

The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of this register (bits 8 and 6–0).

Read accessible always. This bit is read only. Writes have no effect. LEDOUT is unaffected by H_RESET, S_RESET or by setting the STOP bit.

14 LEDPOL LED Polarity. When this bit has the value ZERO, the LED pin will be asserted LOW whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float whenever the OR of the enabled signals is false. (The LED output will be an open drain output, and the output value will be the inverse of the LEDOUT status bit.)

When this bit has the value ONE, the LED pin will be asserted HIGH whenever the OR of the enabled signals is true, and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false. (The LED output will be a totem pole output, and the output value will be the same polarity as the LEDOUT status bit.)

The setting of this bit will not effect the polarity of the LEDOUT bit for this register.

Read/Write accessible always. LEDPOL is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

13 LEDDIS LED Disable. This bit is used to disable the LED output. When LEDDIS is set to ONE and LEDPOL is cleared to ZERO, the LED output pin will be floating. When LEDDIS is set to ONE and LEDPOL is set to ONE, the LED output pin will be driven LOW. When LEDDIS has the value ZERO, the LED output value will be governed by the LEDOUT and LEDPOL values.

Read/Write accessible always. LEDDIS is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

12–10 RES Reserved locations. Written as ZEROs and read as undefined.

9 MPSE Magic Packet Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when magic packet mode is enabled and a magic packet is detected on the network.

Read/Write accessible always. MPSE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

8 FDLSE Full-duplex Link Status Enable. Indicates the full-duplex Link Test Status. When this bit is set to ONE, a value of ONE is passed

to the LEDOUT signal when the PCnet-PCI II controller is functioning in a Link Pass state and full-duplex operation is enabled. When the PCnet-PCI II controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of ZERO is passed to the LEDOUT signal.

When the 10BASE-T port is active, a value of ONE is passed to the LEDOUT signal whenever the Link Test Function detects a Link Pass state and the FDEN (BCR9, bit 0) bit is set. When the AUI port is active, a value of ONE is passed to the LEDOUT signal whenever full-duplex operation on the AUI port is enabled (both FDEN and AUIFD bits in BCR9 are set to ONE).

Read/Write accessible always. FDLSE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

7 PSE Pulse Stretcher Enable. When this bit is set to ONE, the LED illumination time is extended so that brief occurrences of the enabled function will be seen on this LED output. A value of ZERO disables the pulse stretcher.

Read/Write accessible always. PSE is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.

6 LNKSTE Link Status Enable. When this bit is set to ONE, a value of ONE will be passed to the LEDOUT bit in this register when the T-MAU operating in half-duplex mode is in Link Pass state. When the T-MAU operating in half-duplex mode is in Link Fail state, a value of ZERO is passed to the LEDOUT bit.

The function of this bit is masked if the 10BASE-T port is operating in full-duplex mode. This allows a system to have separate LEDs for half-duplex Link Status and for full-duplex Link Status.

Read/Write accessible always. LNKSTE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

5 RCVME Receive Match Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the there is receive activity on the network that has passed the address match function for this node. All address matching modes are included: physical, logical filtering, broadcast and promiscuous.

Read/Write accessible always. RCVME is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

4 XMTE Transmit Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is transmit activity on the network.

Read/Write accessible always. XMTE is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.

3 RXPOLE Receive Polarity Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the polarity of the RXD± pair is not reversed.

Receive polarity indication is valid only if the T-MAU is in link pass state.

Read/Write accessible always. RXPOLE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

2 RCVE Receive Status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is receive activity on the network.

Read/Write accessible always. RCVE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

1 JABE Jabber status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when the PCnet-PCI

II controller is jabbering on the network.

Read/Write accessible always. JABE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

0 COLE Collision status Enable. When this bit is set to ONE, a value of ONE is passed to the LEDOUT bit in this register when there is collision activity on the network. The activity on the collision inputs to the AUI port within the first 4 µs after every transmission for the purpose of SQE testing will not cause the LEDOUT bit to be set.

Read/Write accessible always. COLE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR9: Full-Duplex Control

Bit Name Description

Note that bits 15–0 in this register are programmable through the external EEPROM. Reserved bits and read-only bits should be programmed to ZERO.

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15–3 RES Reserved locations. Written as ZEROs and read as undefined.

2 FDRPAD Full-Duplex Runt Packet Accept Disable. When FDRPAD is set to ONE and full-duplex mode is enabled, the PCnet-PCI II controller will only receive frames that meet the minimum Ethernet frame length of 64 bytes. Receive DMA will not start until at least 64 bytes or a complete frame have been received. By default, FDRPAD is cleared to ZERO. The PCnet-PCI II controller will accept any length frame and receive DMA will start according to the programming of the receive FIFO watermark. Note that there should not be any runt packets in a full-duplex network, since the main cause for runt packets is a network collision and there are no collisions in a full-duplex network.

Read/Write accessible always. FDRPAD is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

1 AUIFD AUI full-duplex. AUIFD enables full-duplex operation on the AUI port. AUIFD is only meaningful if FDEN (BCR9, bit 0) is set to ONE. If the FDEN bit is ZERO, the AUI port will always operate in half-duplex mode. If FDEN is set to ONE and AUIFD is set to ONE, full-duplex operation on the AUI port is enabled. However, if FDEN is set to ONE but the AUIFD bit is cleared to ZERO, the AUI port will always operate in half-duplex mode.

Read/Write accessible always. AUIFD is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

0 FDEN Full-duplex Enable. FDEN enables full-duplex operation. When FDEN is set to ONE, the PCnet-PCI II controller will operate in full-duplex mode when either the 10BASE-T port is enabled. To enable full-duplex operation on the AUI port, the AUIFD bit (BCR9, bit1) must be set to ONE in addition to setting FDEN to ONE. When the DLNKTST bit (CSR15, bit 12) is set to ONE, the 10BASE-T port will operate in half-duplex mode regardless of the setting of FDEN.

AUIFD (bit 1)	FDEN (bit 0)	Effect on the AUI Port	Effect on the 10BASE-T Port
X	0	Half-Duplex	Half-Duplex
0	1	Half-Duplex	Full-Duplex
1	1	Full-Duplex	Full-Duplex

Read/Write accessible always. FDEN is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR16: I/O Base Address Lower

Bit Name Description

31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–5	IOBASEL	Reserved locations. After H_RESET, the value of these bits will be undefined. The settings of these bits will have no effect on any PCnet-PCI II controller function. It is only included for software compatibility with other PCnet family devices. Read/Write accessible always. IOBASEL is not affected by S_RESET or by setting the STOP bit.
4–0	RES	Reserved locations. Written as ZEROs, read as undefined.

BCR17: I/O Base Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IOBASEU	Reserved locations. After H_RESET, the value in this register will be undefined. The setting of this register has no effect on any PCnet-PCI II controller function. It is only included for software compatibility with other PCnet family devices. Read/Write accessible always. IOBASEU is not affected by S_RESET or by setting the STOP bit.

BCR18: Burst and Bus Control Register

Bit	Name	Description
		Note that bits 15–0 in this register are programmable through the external EEPROM. Reserved bits and read-only bits should be programmed to ZERO.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	ROMTMG	Expansion ROM Timing. The value of ROMTMG is used to tune the timing of the Expansion ROM interface. ROMTMG defines the time from when the PCnet-PCI II controller drives ERA[7:0] with the lower 8-bits of the Expansion ROM address to when the PCnet-PCI II controller latches in the data on the ERD[7:0] inputs. The register value specifies the time in number of clock cycles. A ROMTMG value of ZERO results in the same timing as a ROMTMG value of ONE. The access time for the Expansion ROM device (tACC) can be calculated by subtracting the clock to output delay for the ERA[7:0] outputs (tVAL(ERA)) and the input to clock setup time for the ERD[7:0] inputs (tSU(ERD)) from the time defined by ROMTMG: t_{ACC} £ ROMTMG * clock period – t_{VAL}(ERA) – t_{SU}(ERD) For an adapter card application, the value used for clock period should be 30 ns to guarantee correct interface timing at the maximum clock frequency of 33 MHz. Read accessible always. Write accessible only when either the STOP or the SPND bit is set. ROMTMG is set to the value of 1001b by H_RESET and is not affected by S_RESET or by setting the STOP bit. The default value allows using an Expansion ROM with an access time of 250 ns in a system with a maximum clock frequency of 33 MHz.
11–10	RES	Reserved locations. Written as ZEROs and read as undefined.
9	MEMCMD	Memory Command. This bit determines the command code used for burst read accesses to transmit buffers. When MEMCMD is cleared to ZERO, all burst read accesses to transmit buffers are of the PCI command type Memory Read Line (type 14). When MEMCMD is set to ONE, all burst read accesses to transmit buffers are of the PCI command type Memory Read Multiple (type 12). Read accessible always. Write accessible only when either the STOP or the SPND bit is set. MEMCMD is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

8 EXTREQ Extended Request. This bit controls the deassertion of $\overline{\text{REQ}}$ for a burst transaction. If EXTREQ is cleared to ZERO, $\overline{\text{REQ}}$ is deasserted at the beginning of a burst transaction. (The PCnetPCI II controller never performs more than one burst transaction within a single bus mastership period.) In this mode, the PCnetPCI II controller relies on the PCI Latency Timer to get enough bus bandwidth, in case the system arbiter also removes $\overline{\text{GNT}}$ at the beginning of the burst transaction. If EXTREQ is set to ONE, $\overline{\text{REQ}}$ stays asserted until the next to last data phase of the burst transaction is done. This mode is useful for systems that implement an arbitration scheme without preemption and require that $\overline{\text{REQ}}$ stays asserted throughout the transaction.

EXTREQ should not be set to ONE when the PCnet-PCI II controller is used in a PCI bus application.

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. EXTREQ is cleared by H_RESET and or S_RESET and is not affected by setting the STOP bit.

7 DWIO Double Word I/O. When set, this bit indicates that the PCnet-PCI II controller is programmed for DWord I/O (DWIO) mode. When cleared, this bit indicates that the PCnet-PCI II controller is programmed for Word I/O (WIO) mode. This bit affects the I/O Resource Offset map and it affects the defined width of the PCnet-PCI II controller's I/O resources. See the sections "Word I/O Mode" and "Double Word I/O Mode" for more details.

Read accessible always. DWIO is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

6 BREADE Burst Read Enable. When set, this bit enables burst mode during memory read accesses. The PCnet-PCI II controller can perform burst transfers when reading the initialization block, the descriptor ring entries (when SWSTYLE is set to Three), and the buffer memory. When cleared, this bit prevents the device from bursting during read accesses.

BREADE should be set to ONE when the PCnet-PCI II controller is used in a PC bus application to guarantee maximum performance.

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. BREADE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

5 BWRITE Burst Write Enable. When set, this bit enables burst mode during memory write accesses. The PCnet-PCI II controller can perform burst transfers when writing the descriptor ring entries (when SWSTYLE is set to Three) and the buffer memory. When cleared, this bit prevents the device from bursting during write accesses.

BWRITE should be set to ONE when the PCnet-PCI II controller is used in a PCI bus application to guarantee maximum performance.

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. BWRITE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

4–3 TSTSHDW Reserved locations. Written and read as ZEROs.

2–0 RES Reserved locations. Read accessible always. Write accessible only when either the STOP or the SPND bit is set. After H_RESET, the value in these bits will be 001b. The setting of these bits has no effect on any PCnet-PCI II controller function. LINBC is not affected by S_RESET or by setting the STOP bit.

BCR19: EEPROM Control and Status

Bit Name Description

31–16 RES

Reserved locations. Written as ZEROs and read as undefined.

15 PVALID

EEPROM Valid status bit. A value of ONE in this bit indicates that a PREAD operation has occurred, and that (1) there is an EEPROM connected to the PCnet-PCI II controller Microwire interface pins and (2) the contents read from the EEPROM have passed the checksum verification operation. A value of ZERO in this bit indicates that the checksum for the entire 36 bytes of EEPROM is incorrect or that no EEPROM is connected to the Microwire interface pins.

If PVALID becomes ZERO following an EEPROM read operation (either automatically generated after H_RESET, or requested through PREAD), then all EEPROM-programmable BCR locations will be reset to their H_RESET values. The contents of the Address PROM locations, however, will not be cleared.

If the EEPROM detection fails, then all attempted PREAD commands will terminate early and PVALID will not be set. This applies to the automatic read of the EEPROM after H_RESET as well as to host-initiated PREAD commands.

Read accessible only. PVALID is read only. Write operations have no effect. PVALID is cleared to ZERO during H_RESET and is unaffected by S_RESET or by setting the STOP bit.

14 PREAD

EEPROM Read command bit. When this bit is set to ONE by the host, the PVALID bit (BCR19, bit 15) will immediately be cleared to ZERO and then the PCnet-PCI II controller will perform a read operation of 36 bytes from the EEPROM through the Microwire interface. The EEPROM data that is fetched during the read will be stored in the appropriate internal registers on board the PCnet-PCI II controller. EEPROM contents will be indirectly accessible to the host through read accesses to the Address PROM (offsets 0h through Fh) and through read accesses to the EEPROM-programmable BCRs. Note that read accesses from these locations will not actually access the EEPROM itself, but instead will access the PCnet-PCI II controller's internal copy of the EEPROM contents. Write accesses to these locations may change the PCnet-PCI II controller register contents, but the EEPROM locations will not be affected. EEPROM locations may also be accessed directly by programming bits 4–0 of this register.

At the end of the read operation, the PREAD bit will automatically be cleared to ZERO by the PCnet-PCI II controller and PVALID will be set, provided that an EEPROM existed on the Microwire interface pins and that the checksum for the entire 36 bytes of EEPROM was correct.

Note that when PREAD is set to ONE, the PCnet-PCI II controller will no longer respond to any accesses directed toward it, until the PREAD operation has completed successfully. The PCnet-PCI II controller will terminate these accesses with the assertion of $\overline{\text{DEVSEL}}$ and $\overline{\text{STOP}}$ while $\overline{\text{TRDY}}$ is not asserted, signaling to the initiator to disconnect and retry the access at a later time.

If a PREAD command is given to the PCnet-PCI II controller but no EEPROM is detected at the Microwire interface pins, the PREAD command will terminate early, the PREAD bit will be cleared to ZERO, the PVALID bit will remain ZERO, and all EEPROM-programmable BCR locations will be reset to their H_RESET values. The contents of the Address PROM locations, however, will not be cleared.

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. PREAD is cleared to ZERO during H_RESET and is unaffected by S_RESET or by setting the STOP bit.

13 EEDET EEPROM Detect. This bit indicates the sampled value of the EESK/$\overline{\text{LED1}}$/SFBD pin at the rising edge of CLK during the last clock during which $\overline{\text{RST}}$ is asserted. This value indicates whether or not an EEPROM has been detected at the EEPROM interface. If this bit is a ONE, it indicates that an EEPROM has been detected. If this bit is a ZERO, it indicates that an EEPROM has not been detected.

Read accessible always. EEDET is read only. Write operations have no effect. It is unaffected by S_RESET or by setting the STOP bit.

The following table indicates the possible combinations of EEDET and the existence of an EEPROM and the resulting operations that are possible on the EEPROM Microwire interface:

Table 31. EEDET Setting

EEDET Value (BCR19[3])	EEPROM Connected?	Result if PREAD is Set to ONE	Result of Automatic EEPROM Read Operation Following H_RESET
0	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is cleared to ZERO.	First TWO EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is cleared to ZERO.
0	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.	First TWO EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is cleared to ZERO.
1	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is cleared to ZERO.	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is cleared to ZERO.
1	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.

12–5 RES Reserved locations. Written as ZEROs, read as undefined.

4 EEN EEPROM port enable. When this bit is set to ONE, it causes the values of ECS, ESK, and EDI to be driven onto the EECS, EESK, and EEDI pins, respectively. If EEN is cleared to ZERO and no EEPROM read function is currently active, then EECS will be driven LOW and the EESK and EEDI pins change their function to $\overline{\text{LED1}}$ and $\overline{\text{LNKST}}$ and are controlled by BCR5 and BCR4, respectively.

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. EEN is cleared to ZERO by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

3 RES Reserved location. Written as ZERO and read as undefined.

2 ECS EEPROM Chip Select. This bit is used to control the value of the EECS pin of the Microwire interface when the EEN bit is set to ONE and the PREAD bit is cleared to ZERO. If EEN is set to ONE and PREAD is cleared to ZERO and ECS is set to ONE, then the EECS pin will be forced to a HIGH level at the rising edge of the next clock following bit programming. If EEN is set to ONE and PREAD is cleared to ZERO and ECS is cleared to ZERO, then the EECS pin will be forced to a LOW level at the rising edge

of the next clock following bit programming. ECS has no effect on the output value of the EECS pin unless the PREAD bit is cleared to ZERO and the EEN bit is set to ONE.

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. ECS is cleared to ZERO by H_RESET and is not affected by S_RESET or by setting the STOP bit.

1 ESK EEPROM Serial Clock. This bit and the EDI/EDO bit are used to control host access to the EEPROM. Values programmed to this bit are placed on to the EESK pin at the rising edge of the next clock following bit programming, except when the PREAD bit is set to ONE or the EEN bit is cleared to ZERO. If both the ESK bit and the EDI/EDO bit values are changed during one BCR19 write operation while EEN is set to ONE, then setup and hold times of the EEDI pin value with respect to the EESK signal edge are not guaranteed.

ESK has no effect on the EESK pin unless the PREAD bit is cleared to ZERO and the EEN bit is set to ONE.

Table 32. Microwire Interface Pin Assignment

$\overline{\text{RST}}$ Pin	PREAD or Auto Read in Progress	EEN	EECS	EESK	EEDI
High	X	X	Low	Tri-State	Tri-State
Low	1	X	Active	Active	Active
Low	0	1	BCR19[2]	BCR19[1]	BCR19[0]
Low	0	0	Low	$\overline{\text{LED1}}$	$\overline{\text{LNKST}}$

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. ESK is set to ONE by H_RESET and is not affected by S_RESET or by setting the STOP bit.

0 EDI/EDO EEPROM Data In/EEPROM Data Out. Data that is written to this bit will appear on the EEDI output of the Microwire interface, except when the PREAD bit is set to ONE or the EEN bit is cleared to ZERO. Data that is read from this bit reflects the value of the EEDO input of the Microwire interface.

EDI/EDO has no effect on the EEDI pin unless the PREAD bit is cleared to ZERO and the EEN bit is set to ONE.

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. EDI/EDO is cleared to ZERO by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR20: Software Style

Bit Name Description

This register is an alias of the location CSR58. Accesses to/from this register are equivalent to accesses to CSR58.

31–16 RES Reserved locations. Written as ZEROs and read as undefined.

15–11 RES Reserved locations. Written as ZEROs and read as undefined.

10 APERREN Advanced Parity Error Handling Enable. When APERREN is set to ONE, the BPE bits (RMD1 and TMD1, bit 23) are used to indicated parity error in data transfers to the receive and transmit buffers. Note that since the advanced parity error handling uses an additional bit in the descriptor, SWSTYLE (bits 7–0 of this register) must be set to ONE, TWO or THREE to program the PCnet-PCI II controller to use 32-bit software structures.

APERREN does not affect the reporting of address parity errors or data parity errors that occur when the PCnet-PCI II controller is the target of the transfer.

Read accessible always, write accessible only when either the STOP or the SPND bit is set. APERREN is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

9 CSRPCNET CSR PCnet-ISA configuration. When set, this bit indicates that the PCnet-PCI II controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the PCnet-ISA (Am79C960) device. When cleared, this bit indicates that PCnet-PCI II controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the ILACC (Am79C900) device.

The value of CSRPCNET is determined by the PCnet-PCI II controller according to the setting of the Software Style (SWSTYLE, bits 7–0 of this register).

Read accessible always. CSRPCNET is read only. Write operations will be ignored. CSRPCNET will be set after H_RESET (since SWSTYLE defaults to ZERO) and is not affected by S_RESET or by setting the STOP bit.

8 SSIZE32 32-Bit Software Size. When set, this bit indicates that the PCnet-PCI II controller utilizes 32-bit software structures for the initialization block and the transmit and receive descriptor entries. When cleared, this bit indicates that the PCnet-PCI II controller utilizes 16-bit software structures for the initialization block and the transmit and receive descriptor entries. In this mode the PCnet-PCI II controller is backwards compatible with the Am79C90 C-LANCE and Am79C960 PCnet-ISA.

The value of SSIZE32 is determined by the PCnet-PCI II controller according to the setting of the Software Style (SWSTYLE, bits 7–0 of this register).

Read accessible always. SSIZE32 is read only. Write operations will be ignored. SSIZE32 will be cleared after H_RESET (since SWSTYLE defaults to ZERO) and is not affected by S_RESET or by setting the STOP bit.

If SSIZE32 is cleared to ZERO, then bits IADR[31:24] of CSR2 will be used to generate values for the upper 8 bits of the 32 bit address bus during master accesses initiated by the PCnet-PCI II controller. This action is required, since the 16-bit software structures will yield only 24 bits of address for PCnet-PCI II controller bus master accesses.

If SSIZE32 is set to ONE, then the software structures that are common to the PCnet-PCI II controller and the host system will supply a full 32 bits for each address pointer that is needed by the PCnet-PCI II controller for performing master accesses.

The value of the SSIZE32 bit has no effect on the drive of the upper 8 address bits. The upper 8 address pins are always driven, regardless of the state of the SSIZE32 bit.

Note that the setting of the SSIZE32 bit has no effect on the width for I/O accesses. I/O access width is determined by the state of the DWIO bit (BCR18, bit 7).

7–0 SWSTYLE Software Style register. The value in this register determines the style of register and memory resources that shall be used by the PCnet-PCI II controller. The Software Style selection will affect the interpretation of a few bits within the CSR space, the order of the descriptor entries and the width of the descriptors and initialization block entries.

All PCnet-PCI II controller CSR bits and BCR bits and all descriptor, buffer and initialization block entries not cited in the table below are unaffected by the software style selection.

Table 33. Software Styles

SWSTYLE [7:0]	Style Name	CSRPCNET	SSIZE32	Initialization Block Entries	Descriptor Ring Entries	Altered Bit Interpretations
00h	C-LANCE / PCnet-ISA	1	0	16-bit software structures, non-burst or burst access	16-bit software structures, non-burst access only	All bits in CSR4 are used, TMD1[29] is ADD_FCS
01h	ILACC	0	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only	CSR4[9:8], CSR4[5:4] and CSR4[1:0] have no function, TMD1[29] is NO_FCS.
02h	PCnet-PCI	1	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only	All bits in CSR4 are used, TMD1[29] is ADD_FCS
03h	PCnet-PCI II Controller	1	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst or burst access	All bits in CSR4 are used, TMD1[29] is ADD_FCS
All Other	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

Read/Write accessible only when either the STOP or the SPND bit is set. The SWSTYLE register will contain the value 00h following H_RESET and will be unaffected by S_RESET or by setting the STOP bit.

BCR21: Interrupt Control

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	INTCON	Reserved locations. The setting of this register has no effect on any PCnet-PCI II controller function. It is only included for software compatibility with other PCnet family devices. Read/Write accessible always. INTCON is not affected by S_RESET or by setting the STOP bit.

BCR22: PCI Latency Register

Bit	Name	Description
		Note that bits 15–0 in this register are programmable through the external EEPROM.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	MAX_LAT	Maximum Latency. Specifies the maximum arbitration latency the PCnet-PCI II controller can sustain without causing problems to the network activity. The register value specifies the time in units of 1/4 microseconds. MAX_LAT is aliased to the PCI configuration space register MAX_LAT (offset 3Fh). The host should use the value in this register to determine the setting of the PCI Latency Timer register. Read accessible always. Write accessible only when either the STOP or the SPND bit is set. MAX_LAT is set to the value of FFh by H_RESET which corresponds to a maximum latency of 63.75 microseconds. The actual maximum latency the PCnet-PCI II controller can handle is 153.6 µs which is also the value for the bus time-out (see CSR100). MAX_LAT is not affected by

S_RESET or by setting the STOP bit.

7–0 MIN_GNT Minimum Grant. Specifies the minimum length of a burst period the PCnet-PCI II controller needs to keep up with the network activity. The length of the burst period is calculated assuming a clock rate of 33 MHz. The register value specifies the time in units of 1/4 microseconds. MIN_GNT is aliased to the PCI configuration space register MIN_GNT (offset 3Eh). The host should use the value in this register to determine the setting of the PCI Latency Timer register.

Read accessible always. Write accessible only when either the STOP or the SPND bit is set. MIN_GNT is set to the value of 06h by H_RESET which corresponds to a minimum grant of 1.5 microseconds. 1.5 microseconds is the time it takes to PCnet-PCI II controller to read/write 64 bytes. (16 DWord transfers in burst mode with one extra wait state per data phase inserted by the target.) Note that the default is only a typical value. It also does not take into account any descriptor accesses. MIN_GNT is not affected by S_RESET or by setting the STOP bit.

Initialization Block

When SSIZE32 (BCR20, bit 8) is set to ZERO, the software structures are defined to be 16 bits wide. The base address of the initialization block must be aligned to a DWord boundary, i.e. CSR1, bit 1 and 0 must be cleared to ZERO. When SSIZE32 is set to ZERO, the initialization block looks like this:

Table 34. Initialization Block (SSIZE32 = 0)

Address	Bits 15–13	Bit 12	Bits 11–8	Bits 7–4	Bits 3–0
IADR+00h	MODE 15–00				
IADR+02h	PADR 15–00				
IADR+04h	PADR 31–16				
IADR+06h	PADR 47–32				
IADR+08h	LADRF 15–00				
IADR+0Ah	LADRF 31–16				
IADR+0Ch	LADRF 47–32				
IADR+0Eh	LADRF 63–48				
IADR+10h	RDRA 15–00				
IADR+12h	RLEN	0	RES	RDRA 23–16	
IADR+14h	TDRA 15–00				
IADR+16h	TLEN	0	RES	TDRA 23–16	

Table 35. Initialization Block (SSIZE32 = 1)

Address	Bits 31–28	Bits 27–24	Bits 23–20	Bits 19–16	Bits 15–12	Bits 11–8	Bits 7–4	Bits 3–0
IADR+00h	TLEN	RES	RLEN	RES	MODE			
IADR+04h	PADR 31–00							
IADR+08h	RES				PADR 47–32			
IADR+0Ch	LADR 31–00							
IADR+10h	LADR 63–32							
IADR+14h	RDRA 31–00							
IADR+18h	TDRA 31–00							

Note that the PCnet-PCI II controller performs DWord accesses to read the initialization block. This statement is always true, regardless of the setting of the SSIZE32 bit.

When SSIZE32 (BCR20, bit 8) is set to ONE, the software structures are defined to be 32 bits wide. The base address of the initialization block must be aligned to a DWord boundary, i.e. CSR1, bits 1 and 0 must be cleared to ZERO. When SSIZE32 is set to ONE, the initialization block looks as shown in Table 35.

RLEN AND TLEN

When SSIZE32 (BCR20, bit 8) is set to ZERO, the software structures are defined to be 16 bits wide, and the RLEN and TLEN fields in the initialization block are each 3 bits wide. The values in these fields determine the number of transmit and receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is as follows:

Table 36. R/TLEN Decoding (SSIZE32 = 0)

R/TLEN	No. of DREs
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete.

When SSIZE32 (BCR20, bit 8) is set to ONE, the software structures are defined to be 32 bits wide, and the RLEN and TLEN fields in the initialization block are each 4 bits wide. The values in these fields determine the number of transmit and receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is as follows:

Table 37. R/TLEN Decoding (SSIZE32 = 1)

R/TLEN	No. of DREs
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
11XX	512
1X1X	512

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete.

RDRA and TDRA

TDRA and RDRA indicate where the transmit and receive descriptor rings begin. Each DRE must be located at a 16-byte address boundary when SSIZE32 is set to ONE (BCR20, bit 8). Each DRE must be located at an 8-byte address boundary when SSIZE32 is set to ZERO (BCR20, bit 8).

LADRF

The Logical Address Filter (LADRF) is a 64-bit mask that is used to accept incoming Logical Addresses. If the first bit in the incoming address (as transmitted on the wire) is a ONE, it indicates a logical address. If the first bit is a ZERO, it is a physical address and is compared against the physical address that was loaded through the initialization block.

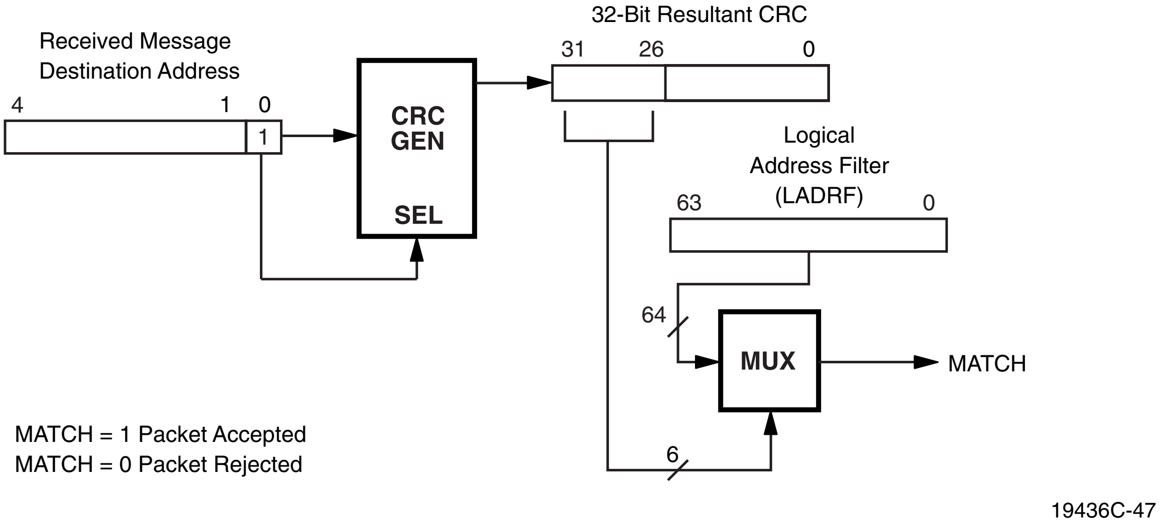


Figure 44. Address Match Logic

A logical address is passed through the CRC generator, producing a 32 bit result. The high order 6 bits of the CRC is used to select one of the 64 bit positions in the Logical Address Filter. If the selected filter bit is set, the address is accepted and the frame is placed into memory.

The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the node's responsibility to determine if the message is actually intended for the node by comparing the destination address of the stored message with a list of acceptable logical addresses.

If the Logical Address Filter is loaded with all ZEROs and promiscuous mode is disabled, all incoming logical addresses except broadcast will be rejected.

PADR

This 48-bit value represents the unique node address assigned by the ISO 8802-3 (IEEE/ANSI 802.3) and used for internal address comparison. PADR[0] is compared with the first bit in the destination address of the incoming frame. It must be ZERO since only the destination address of a unicast frames is compared to PADR. The six hex-digit nomenclature used by the ISO 8802-3 (IEEE/ANSI 802.3) maps to the PCnet-PCI II controller PADR register as follows: the first byte is compared with PADR[7:0], with PADR[0] being the least significant bit of the byte. The second ISO 8802-3 (IEEE/ANSI 802.3) byte is compared with PADR[15:8], again from the least significant bit to the most significant bit, and so on. The sixth byte is compared with PADR[47:40], the least significant bit being PADR[40].

MODE

The mode register field of the initialization block is copied into CSR15 and interpreted according to the description of CSR15.

Receive Descriptors

When SWSTYLE (BCR20, bits 7–0) is set to ZERO, then the software structures are defined to be 16 bits wide, and receive descriptors, (CRDA = Current Receive Descriptor Address), are as shown in Table 38.

When SWSTYLE (BCR 20, bits 7–0) is set to ONE or TWO, then the software structures are defined to be 32 bits wide, and receive descriptors, (CRDA = Current Receive Descriptor Address), are as shown in Table 39.

When SWSTYLE (BCR 20, bits 7–0) is set to THREE, then the software structures are defined to be 32 bits wide, and receive descriptors, (CRDA = Current Receive Descriptor Address), are as shown in Table 40.

Table 38. Receive Descriptor (SWSTYLE = 0)

Address	15	14	13	12	11	10	9	8	7–0
CRDA+00h	RBADR[15:0]								
CRDA+02h	OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	RBADR[23:16]
CRDA+04h	1	1	1	1	BCNT				
CRDA+06h	0	0	0	0	MCNT				

Table 39. Receive Descriptor (SWSTYLE = 1,2)

Address	31	30	29	28	27	26	25	24	23	22	21	20	19–16	15–12	11–0
CRDA+00h	RBADR[31:0]														
CRDA+04h	OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	BPE	PAM	LAFM	BAM	RES	1111	BCNT
CRDA+08h	RCC								RPC					0000	MCNT
CRDA+0Ch	RESERVED														

Table 40. Receive Descriptor (SWSTYLE = 3)

Address	31	30	29	28	27	26	25	24	23	22	21	20	19–16	15–12	11–0
CRDA+00h	RCC								RPC					0000	MCNT
CRDA+04h	OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	BPE	PAM	LAFM	BAM	RES	1111	BCNT
CRDA+08h	RBADR[31:0]														
CRDA+0Ch	RESERVED														

RMD0

Bit Name Description

31–0 RBADR Receive Buffer address. This field contains the address of the receive buffer that is associated with this descriptor.

RMD1

Bit Name Description

31 OWN This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the PCnet-PCI II controller (OWN = 1). The PCnet-PCI II controller clears the OWN bit after filling the buffer that the descriptor points to. The host sets the OWN bit after emptying the buffer. Once the PCnet-PCI II controller or host has relinquished ownership of a buffer, it must not change any field in the descriptor entry.

30 ERR ERR is the OR of FRAM, OFLO, CRC, BUFF or BPE. ERR is set by the PCnet-PCI II controller and cleared by the host.

29 FRAM Framing error indicates that the incoming frame contains a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non-integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is set by the PCnet-PCI II controller and cleared by the host.

28 OFLO Overflow error indicates that the receiver has lost all or part of the incoming frame, due to an inability to move data from the receive FIFO into a memory buffer before the internal FIFO overflowed. OFLO is valid only when ENP is not set. OFLO is set by the PCnet-PCI II controller and cleared by the host.

27 CRC CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is set by the PCnet-PCI II controller and cleared by the host.

26 BUFF Buffer error is set any time the PCnet-PCI II controller does not own the next buffer while data chaining a received frame. This can occur in either of two ways:

1. The OWN bit of the next buffer is ZERO.
2. FIFO overflow occurred before the PCnet-PCI II controller was able to read the OWN bit of the next descriptor.

If a Buffer Error occurs, an Overflow Error may also occur internally in the FIFO, but will not be reported in the descriptor status entry unless both BUFF and OFLO errors occur at the same time. BUFF is set by the PCnet-PCI II controller and cleared by the host.

25 STP Start of Packet indicates that this is the first buffer used by the PCnet-PCI II controller for this frame. If STP and ENP are both set to ONE, the frame fits into a single buffer. Otherwise, the frame is spread over more than one buffer. When LAPPEN (CSR3, bit 5) is cleared to ZERO, STP is set by the PCnet-PCI II controller and cleared by the host. When LAPPEN is set to ONE, STP must be set by the host.

24 ENP End of Packet indicates that this is the last buffer used by the PCnet-PCI II controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the PCnet-PCI II controller and cleared by the host.

23 BPE Bus Parity Error is set by the PCnet-PCI II controller when a parity error occurred on the bus interface during a data transfers to a receive buffer. BPE is valid only when ENP, OFLO or BUFF are set. The PCnet-PCI II controller will only set BPE when the advanced parity error handling is enabled by setting APERREN (BCR20, bit 10) to ONE. BPE is set the PCnet-PCI II controller and cleared by the host.

This bit does not exist, when the PCnet-PCI II controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7–0, SWSTYLE is cleared to ZERO).

22 PAM Physical Address Match is set by the PCnet-PCI II controller when it accepts the received frame due to a match of the frame's destination address with the content of the physical address register. PAM is valid only when ENP is set. PAM is set by the PCnet-PCI II controller and cleared by the host.

This bit does not exist when the PCnet-PCI II controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7–0, SWSTYLE is cleared to ZERO).

21 LAFM Logical Address Filter Match is set by the PCnet-PCI II controller when it accepts the received frame based on the value in the logical address filter register. LAFM is valid only when ENP is set. LAFM is set by the PCnet-PCI II controller and cleared by the host.

Note that if DRCVBC (CSR15, bit 14) is cleared to ZERO, only BAM, but not LAFM will be set when a Broadcast frame is received, even if the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter. If DRCVBC is set to ONE and the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter, LAFM will be set on the reception of a Broadcast frame.

This bit does not exist when the PCnet-PCI II controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7–0, SWSTYLE is cleared to ZERO).

20 BAM Broadcast Address Match is set by the PCnet-PCI II controller when it accepts the received frame because the frame's destination address is of the type "Broadcast". BAM is valid only when ENP is set. BAM is set by the PCnet-PCI II controller and cleared by the host.

This bit does not exist when the PCnet-PCI II controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7–0, SWSTYLE is cleared to ZERO).

19–16 RES Reserved locations. These locations should be read and written as ZEROs.

15–12 ONES These four bits must be written as ONEs. They are written by the host and unchanged by the PCnet-PCI II controller.

11–00 BCNT Buffer Byte Count is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and unchanged by the PCnet-PCI II controller.

RMD2

Bit Name Description

31–24 RCC Receive Collision Count. Indicates the accumulated number of collisions detected on the network since the last packet was received, excluding collisions that occurred during transmissions from this node. The PCnet-PCI II controller implementation of this counter may not be compatible with the ILACC RCC definition. If network statistics are to be monitored, then CSR114 should be used for the purpose of monitoring receive collisions instead of these bits.

23–16 RPC Runt Packet Count. Indicates the accumulated number of runts that were addressed to this node since the last time that a receive packet was successfully received and its corresponding RMD2 ring entry was written to by the PCnet-PCI II controller. In order to be included in the RPC value, a runt must be long enough to meet the minimum requirement of the internal address matching logic. The minimum requirement for a runt to pass the internal address matching mechanism is: 18 bits of valid preamble plus a valid SFD detected, followed by 7 bytes of frame data. This requirement is unvarying, regardless of the address matching mechanisms in force at the time of reception. (I.e. physical, logical, broadcast or promiscuous). The PCnet-PCI II controller implementation of this counter may not be compatible with the ILACC RPC definition.

15–12 ZEROS This field is reserved. PCnet-PCI II controller will write ZEROs to these locations.

11–0 MCNT Message Byte Count is the length in bytes of the received message, expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the PCnet-PCI II controller and cleared by the host.

RMD3

Bit Name Description

31–0 RES Reserved locations.

Transmit Descriptors

When SWSTYLE (BCR20, bits 7–0) is set to ZERO, the software structures are defined to be 16 bits wide, and transmit descriptors, (CXDA = Current Transmit Descriptor Address), are as shown in Table 41.

When SWSTYLE (BCR 20, bits 7–0) is set to ONE or TWO, the software structures are defined to be 32 bits wide, and transmit descriptors, (CXDA = Current Transmit Descriptor Address), are as shown in Table 42.

When SWSTYLE (BCR 20, bits 7–0) is set to THREE, then the software structures are defined to be 32 bits wide, and transmit descriptors, (CXDA = Current Transmit Descriptor Address), are as shown in Table 43.

Table 41. Transmit Descriptor (SWSTYLE = 0)

Address	15	14	13	12	11	10	9	8	7–0
CXDA+00h	TBADR[15:0]								
CXDA+02h	OWN	ERR	ADD_/ NO_ FCS	MORE / LTINT	ONE	DEF	STP	ENP	TBADR[23:16]

Address	15	14	13	12	11	10	9	8	7–0
CXDA+04h	1	1	1	1	BCNT				
CXDA+06h	BUFF	UFLO	EX DEF	LCOL	LCAR	RTRY	TDR		

Table 42. Transmit Descriptor (SWSTYLE = 1,2)

Address	31	30	29	28	27	26	25	24	23	22–16	15–12	11–4	3–0
CXDA+00h	TBADR[31:0]												
CXDA+04h	OWN	ERR	ADD_/ NO_ FCS	MORE / LTINT	ONE	DEF	STP	ENP	BPE	RES	1111	BCNT	
CXDA+08h	BUFF	UFLO	EX DEF	LCOL	LCAR	RTRY	TDR					RES	TRC
CXDA+0Ch	RESERVED												

Table 43. Transmit Descriptor (SWSTYLE = 3)

Address	31	30	29	28	27	26	25	24	23	22–16	15–12	11–4	3–0
CXDA+00h	BUFF	UFLO	EX DEF	LCOL	LCAR	RTRY	TDR					RES	TRC
CXDA+04h	OWN	ERR	ADD_/ NO_ FCS	MORE / LTINT	ONE	DEF	STP	ENP	BPE	RES	1111	BCNT	
CXDA+08h	TBADR[31:0]												
CXDA+0Ch	RESERVED												

TMD0

Bit Name Description

31–0 TBADR Transmit Buffer address. This field contains the address of the transmit buffer that is associated with this descriptor.

TMD1

Bit Name Description

31 OWN This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the PCnet-PCI II controller (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The PCnet-PCI II controller clears the OWN bit after transmitting the contents of the buffer. Both the PCnet-PCI II controller and the host must not alter a descriptor entry after it has relinquished ownership.

30 ERR ERR is the OR of UFLO, LCOL, LCAR, RTRY or BPE. ERR is set by the PCnet-PCI II controller and cleared by the host. This bit is set in the current descriptor when the error occurs, and therefore may be set in any descriptor of a chained buffer transmission.

29 ADD_FCS/NO Bit 29 functions as when NO_FCSSWSTYLE (BCR20, bits 7–0) is set to ONE (ILACC style). Otherwise bit 29 functions as ADD_FCS.

ADD_FCS ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the STP bit is set. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS is cleared to ZERO, FCS generation is controlled by DXMTFCS. When APAD_XMT (CSR4, bit 11) is set to ONE, the setting of ADD_FCS has no effect. ADD_FCS is set by the host, and is not changed by the PCnet-PCI II controller. This is a reserved bit in the C-LANCE

		(Am79C90). This function differs from the corresponding ILACC function.
	NO_FCS	NO_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the ENP bit is set. When NO_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is deactivated. When NO_FCS is cleared to ZERO, FCS generation is controlled by DXMTFCS. When APAD_XMT (CSR4, bit 11) is set to ONE, the setting of NO_FCS has no effect. NO_FCS is set by the host, and is not changed by the PCnet-PCI II controller. This is a reserved bit in the C-LANCE (Am79C90). This function is identical to the corresponding ILACC function.
28	MORE/LTINT	Bit 28 always function as MORE. The value of MORE is written by the PCnet-PCI II controller and is read by the host. When LTINTEN is cleared to ZERO (CSR5, bit 14), the PCnet-PCI II controller will never look at the content of bit 28, write operations by the host have no effect. When LTINTEN is set to ONE bit 28 changes its function to LTINT on host write operations and on PCnet-PCI II controller read operations.
	MORE	MORE indicates that more than one retry was needed to transmit a frame. The value of MORE is written by the PCnet-PCI II controller. This bit has meaning only if the ENP bit is set.
	LTINT	LTINT is used to suppress interrupts after successful transmission on selected frames. When LTINT is cleared to ZERO and ENP is set to ONE, the PCnet-PCI II controller will not set TINT (CSR0, bit 9) after a successful transmission. TINT will only be set when the last descriptor of a frame has both LTINT and ENP set to ONE. When LTINT is cleared to ZERO, it will only cause the suppression of interrupts for successful transmission. TINT will always be set if the transmission has an error. The LTINTEN overrides the function of TOKINTD (CSR5, bit 15).
27	ONE	ONE indicates that exactly one retry was needed to transmit a frame. ONE flag is not valid when LCOL is set. The value of the ONE bit is written by the PCnet-PCI II controller. This bit has meaning only if the ENP bit is set.
26	DEF	Deferred indicates that the PCnet-PCI II controller had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the PCnet-PCI II controller is ready to transmit. DEF is set by the PCnet-PCI II controller and cleared by the host.
25	STP	Start of Packet indicates that this is the first buffer to be used by the PCnet-PCI II controller for this frame. It is used for data chaining buffers. The STP bit must be set in the first buffer of the frame, or the PCnet-PCI II controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is set by the host and is not changed by the PCnet-PCI II controller.
24	ENP	End of Packet indicates that this is the last buffer to be used by the PCnet-PCI II controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the host and is not changed by the PCnet-PCI II controller.
23	BPE	Bus Parity Error is set by the PCnet-PCI II controller when a parity error occurred on the bus interface during a data transfers from the transmit buffer associated with this descriptor. The PCnet-PCI II controller will only set BPE when the advanced parity error handling is enabled by setting APERREN (BCR20, bit 10) to ONE. BPE is set by the PCnet-PCI II controller and cleared by the host.

		This bit does not exist, when the PCnet-PCI II controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7–0, SWSTYLE is cleared to ZERO).
22–16	RES	Reserved locations.
15–12	ONES	These four bits must be written as ONEs. This field is written by the host and unchanged by the PCnet-PCI II controller.
11–00	BCNT	Buffer Byte Count is the usable length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the PCnet-PCI II controller. This field is written by the host and is not changed by the PCnet-PCI II controller. There are no minimum buffer size restrictions.

TMD2

Bit	Name	Description
31	BUFF	Buffer error is set by the PCnet-PCI II controller during transmission when the PCnet-PCI II controller does not find the ENP flag in the current descriptor and does not own the next descriptor. This can occur in either of two ways: 1. The OWN bit of the next descriptor is ZERO. 2. FIFO underflow occurred before the PCnet-PCI II controller obtained the STATUS byte (TMD1[31:24]) of the next descriptor. BUFF is set by the PCnet-PCI II controller and cleared by the host. If a Buffer Error occurs, an Underflow Error will also occur. BUFF is not valid when LCOL or RTRY error is set during transmit data chaining. BUFF is set by the PCnet-PCI II controller and cleared by the host.
30	UFLO	Underflow error indicates that the transmitter has truncated a message because it could not read data from memory fast enough. UFLO indicates that the FIFO has emptied before the end of the frame was reached. When DXSUFLO (CSR3, bit 6) is cleared to ZERO, the transmitter is turned off when an UFLO error occurs (CSR0, TXON = 0). When DXSUFLO is set to ONE, the PCnet-PCI II controller gracefully recovers from an UFLO error. It scans the transmit descriptor ring until it finds the start of a new frame and starts a new transmission. UFLO is set by the PCnet-PCI II controller and cleared by the host.
29	EXDEF	Excessive Deferral. Indicates that the transmitter has experienced Excessive Deferral on this transmit frame, where Excessive Deferral is defined in ISO 8802-3 (IEEE/ANSI 802.3). Excessive Deferral will also set the interrupt bit EXDINT (CSR5, bit 7).
28	LCOL	Late Collision indicates that a collision has occurred after the first slot time of the channel has elapsed. The PCnet-PCI II controller does not retry on late collisions. LCOL is set by the PCnet-PCI II controller and cleared by the host.
27	LCAR	Loss of Carrier is set when the carrier is lost during a PCnet-PCI II controller-initiated transmission when in AUI mode and the device is operating in half-duplex mode. The PCnet-PCI II controller does not retry upon loss of carrier. It will continue to transmit the whole frame until done. LCAR will not be set when the device is operating in full-duplex mode and the AUI port is active. LCAR is not valid in Internal Loopback Mode. LCAR is set by the PCnet-PCI II controller and cleared by the host. In 10BASE-T mode, LCAR will be set when the T-MAU was in Link Fail state during the transmission.

26	RTRY	Retry error indicates that the transmitter has failed after 16 attempts to successfully transmit a message, due to repeated collisions on the medium. If DRTY is set to ONE in the MODE register, RTRY will set after 1 failed transmission attempt. RTRY is set by the PCnet-PCI II controller and cleared by the host.
25–16	TDR	Time Domain Reflectometer reflects the state of an internal PCnet-PCI II controller counter that counts at a 10 MHz rate from the start of a transmission to the occurrence of a collision or loss of carrier. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the PCnet-PCI II controller and is valid only if RTRY is set. Note that 10 MHz gives very low resolution and in general has not been found to be particularly useful. This feature is here primarily to maintain full compatibility with the C-LANCE device (Am79C90).
15–4	RES	Reserved locations.
3–0	TRC	Transmit Retry Count. Indicates the number of transmit retries of the associated packet. The maximum count is 15. However, if a RETRY error occurs, the count will roll over to ZERO. In this case only, the Transmit Retry Count value of ZERO should be interpreted as meaning 16. TRC is written by the PCnet-PCI II controller into the last transmit descriptor of a frame, or when an error terminates a frame. Valid only when OWN is cleared to ZERO.

TMD3

Bit	**Name**	**Description**
31–0	RES	Reserved locations.

REGISTER SUMMARY

PCI Configuration Registers

Offset	Name	Width in Bit	Access Mode	Default Value
00h	PCI Vendor ID	16	RO	1022h
02h	PCI Device ID	16	RO	2000h
04h	PCI Command	16	RW	0000h
06h	PCI Status	16	RW	0280h
08h	PCI Revision ID	8	RO	1xh
09h	PCI Programming IF	8	RO	00h
0Ah	PCI Sub-Class	8	RO	00h
0Bh	PCI Base-Class	8	RO	02h
0Ch	Reserved	8	RO	00h
0Dh	PCI Latency Timer	8	RO	00h
0Eh	PCI Header Type	8	RO	00h
0Fh	Reserved	8	RO	00h
10h	PCI I/O Base Address	32	RW	0000 0001h
14h	PCI Memory Mapped I/O Base Address	32	RW	0000 0000h
18h–2Fh	Reserved	8	RO	00h
30h	PCI Expansion ROM Base Address	8	RO	0000 0000h
34–3Bhh	Reserved	8	RO	00h
3Ch	PCI Interrupt Line	8	RW	00h
3Dh	PCI Interrupt Pin	8	RO	01h
3Eh	PCI MIN_GNT	8	RO	06h
3Fh	PCI MAX_LAT	8	RO	FFh
40h–FFh	Reserved	8	RO	00h

Note: *RO = read only, RW = read/write, x = silicon-revision dependent*

Control and Status Registers

RAP Addr	Symbol	Default Value	Comments	Use
00	CSR0	uuuu 0004	PCnet-PCI II Controller Status Register	R
01	CSR1	uuuu uuuu	Lower IADR]: Maps to Location 16	S
02	CSR2	uuuu uuuu	Upper IADR: Maps to Location 17	S
03	CSR3	uuuu 0000	Interrupt Masks and Deferral Control	S
04	CSR4	uuuu 0115	Test and Features Control	R
05	CSR5	uuuu 0000	Extended Control and Interrupt	R
06	CSR6	uuuu uuuu	RXTX: RX/TX Encoded Ring Lengths	S
07	CSR7	uuuu uuuu	Reserved	
08	CSR8	uuuu uuuu	LADR0: Logical Address Filter –- LADRF[15:0]	S
09	CSR9	uuuu uuuu	LADR1: Logical Address Filter –- LADRF[31:16]	S
10	CSR10	uuuu uuuu	LADR2: Logical Address Filter –- LADRF[47:32]	S
11	CSR11	uuuu uuuu	LADR3: Logical Address Filter –- LADRF[63:48]	S
12	CSR12	uuuu uuuu	PADR0: Physical Address Register –- PADR[15:0][	S
13	CSR13	uuuu uuuu	PADR1: Physical Address Register –- PADR[31:16]	S
14	CSR14	uuuu uuuu	PADR2: Physical Address Register –- PADR[47:32]	S
15	CSR15	see reg. desc.	MODE: Mode Register	S
16	CSR16	uuuu uuuu	IADR[15:0]: Base Address of INIT Block Lower (Copy)	T
17	CSR17	uuuu uuuu	IADR[31:16]: Base Address of INIT Block Upper (Copy)	T
18	CSR18	uuuu uuuu	CRBAL: Current Receive Buffer Address Lower	T
19	CSR22	uuuu uuuu	CRBAU: Current Receive Buffer Address Upper	T
20	CSR20	uuuu uuuu	CXBAL: Current Transmit Buffer Address Lower	T
21	CSR21	uuuu uuuu	CXBAU: Current Transmit Buffer Address Upper	T
22	CSR22	uuuu uuuu	NRBAL: Next Receive Buffer Address Lower	T
23	CSR23	uuuu uuuu	NRBAU: Next Receive Buffer Address Upper	T
24	CSR24	uuuu uuuu	BADRL: Base Address of Receive Ring Lower	S
25	CSR25	uuuu uuuu	BADRU: Base Address of Receive Ring Upper	S
26	CSR26	uuuu uuuu	NRDAL: Next Receive Descriptor Address Lower	T
27	CSR27	uuuu uuuu	NRDAU: Next Receive Descriptor Address Upper	T
28	CSR28	uuuu uuuu	CRDAL: Current Receive Descriptor Address Lower	T
29	CSR29	uuuu uuuu	CRDAU: Current Receive Descriptor Address Upper	T
30	CSR30	uuuu uuuu	BADXL: Base Address of Transmit Descriptor Ring Lower	S
31	CSR31	uuuu uuuu	BADXU: Base Address of Transmit Descriptor Ring Upper	S
32	CSR32	uuuu uuuu	NXDAL: Next XMT Descriptor Address Lower	T
33	CSR33	uuuu uuuu	NXDAU: Next XMT Descriptor Address Upper	T
34	CSR34	uuuu uuuu	CXDAL: Current Transmit Descriptor Address Lower	T
35	CSR35	uuuu uuuu	CXDAU: Current Transmit Descriptor Address Upper	T
36	CSR36	uuuu uuuu	NNRDAL: Next Next Receive Descriptor Address Lower	T
37	CSR37	uuuu uuuu	NNRDAU: Next Next Receive Descriptor Address Upper	T

Note: *u = undefined value, R = Running register, S = Setup register, T = Test register*

Control and Status Registers (continued)

RAP Addr	Symbol	Default Value	Comments	Use
38	CSR38	uuuu uuuu	NNXDAL: Next Next Transmit Descriptor Address Lower	T
39	CSR39	uuuu uuuu	NNXDAU: Next Next Transmit Descriptor Address Upper	T
40	CSR40	uuuu uuuu	CRBC: Current Receive Byte Count	T
41	CSR41	uuuu uuuu	CRST: Current Receive Status	T
42	CSR42	uuuu uuuu	CXBC: Current Transmit Byte Count	T
43	CSR43	uuuu uuuu	CXST: Current Transmit Status	T
44	CSR44	uuuu uuuu	NRBC: Next Receive Byte Count	T
45	CSR45	uuuu uuuu	NRST: Next Receive Status	T
46	CSR46	uuuu uuuu	POLL: Poll Time Counter	T
47	CSR47	uuuu uuuu	PI: Polling Interval	S
48	CSR48	uuuu uuuu	Reserved	
49	CSR49	uuuu uuuu	Reserved	
50	CSR50	uuuu uuuu	Reserved	
51	CSR51	uuuu uuuu	Reserved	
52	CSR52	uuuu uuuu	Reserved	
53	CSR53	uuuu uuuu	Reserved	
54	CSR54	uuuu uuuu	Reserved	
55	CSR55	uuuu uuuu	Reserved	
56	CSR56	uuuu uuuu	Reserved	
57	CSR57	uuuu uuuu	Reserved	
58	CSR58	see reg. desc.	SWS: Software Style	S
59	CSR59	uuuu uuuu	Reserved	
60	CSR60	uuuu uuuu	PXDAL: Previous Transmit Descriptor Address Lower	T
61	CSR61	uuuu uuuu	PXDAU: Previous Transmit Descriptor Address Upper	T
62	CSR62	uuuu uuuu	PXBC: Previous Transmit Byte Count	T
63	CSR63	uuuu uuuu	PXST: Previous Transmit Status	T
64	CSR64	uuuu uuuu	NXBA: Next Transmit Buffer Address Lower	T
65	CSR65	uuuu uuuu	NXBAU: Next Transmit Buffer Address Upper	T
66	CSR66	uuuu uuuu	NXBC: Next Transmit Byte Count	T
67	CSR67	uuuu uuuu	NXST: Next Transmit Status	T
68	CSR68	uuuu uuuu	Reserved	
69	CSR69	uuuu uuuu	Reserved	
70	CSR70	uuuu uuuu	Reserved	
71	CSR71	uuuu uuuu	Reserved	
72	CSR72	uuuu uuuu	RCVRC: Receive Ring Counter	T
73	CSR73	uuuu uuuu	Reserved	
74	CSR74	uuuu uuuu	XMTRC: Transmit Descriptor Ring Counter	T
75	CSR75	uuuu uuuu	Reserved	
76	CSR76	uuuu uuuu	RCVRL: Receive Descriptor Ring Length	S
77	CSR77	uuuu uuuu	Reserved	
78	CSR78	uuuu uuuu	XMTRL: Transmit Descriptor Ring Length	S
79	CSR79	uuuu uuuu	Reserved	
80	CSR80	uuuu 1410	DMATCFW: DMA Transfer Counter and FIFO Watermark	S
81	CSR81	uuuu uuuu	Reserved	
82	CSR82	uuuu 0000	DMABAT: Bus Activity Timer	S

Control and Status Registers (continued)

RAP Addr	Symbol	Default Value	Comments	Use
83	CSR83	uuuu uuuu	Reserved	
84	CSR84	uuuu uuuu	DMABA: Address Register Lower	T
85	CSR85	uuuu uuuu	DMABAU: Address Register Upper	T
86	CSR86	uuuu uuuu	DMABC: Buffer Byte Counter	T
87	CSR87	uuuu uuuu	Reserved	
88	CSR88	0242 1003	Chip ID Register Lower	T
89	CSR89	uuuu 0262	Chip ID Register Upper	T
90	CSR90	uuuu uuuu	Reserved	
91	CSR91	uuuu uuuu	Reserved	
92	CSR92	uuuu uuuu	Reserved	
93	CSR93	uuuu uuuu	Reserved	
94	CSR94	uuuu 0000	XMTTDR: Transmit Time Domain Reflectometry Count	T
95	CSR95	uuuu uuuu	Reserved	
96	CSR96	uuuu uuuu	Reserved	
97	CSR97	uuuu uuuu	Reserved	
98	CSR98	uuuu uuuu	Reserved	
99	CSR99	uuuu uuuu	Reserved	
100	CSR100	uuuu 0200	Bus Time-Out	S
101	CSR101	uuuu uuuu	Reserved	
102	CSR102	uuuu uuuu	Reserved	
103	CSR103	uuuu 0105	Reserved	
104	CSR104	uuuu uuuu	Reserved	
105	CSR105	uuuu uuuu	Reserved	
106	CSR106	uuuu uuuu	Reserved	
107	CSR107	uuuu uuuu	Reserved	
108	CSR108	uuuu uuuu	Reserved	
109	CSR109	uuuu uuuu	Reserved	
110	CSR110	uuuu uuuu	Reserved	
111	CSR111	uuuu uuuu	Reserved	
112	CSR112	uuuu 0000	Missed Frame Count	R
113	CSR113	uuuu uuuu	Reserved	
114	CSR114	uuuu 0000	Receive Collision Count	R
115	CSR115	uuuu uuuu	Reserved	
116	CSR116	uuuu uuuu	Reserved	
117	CSR117	uuuu uuuu	Reserved	
118	CSR118	uuuu uuuu	Reserved	
119	CSR119	uuuu uuuu	Reserved	
120	CSR120	uuuu uuuu	Reserved	
121	CSR121	uuuu uuuu	Reserved	
122	CSR122	uuuu 0000	Receive Frame Alignment Control	S
123	CSR123	uuuu uuuu	Reserved	
124	CSR124	uuuu 0000	Test Register 1	
125	CSR125	uuuu uuuu	Reserved	
126	CSR126	uuuu uuuu	Reserved	
127	CSR127	uuuu uuuu	Reserved	T

BUS CONFIGURATION REGISTERS

BCR	MNEMONIC	Default	Description	Programmability	
				User	EEPROM
0	MSRDA	0005h	Reserved	No	No
1	MSWRA	0005h	Reserved	No	No
2	MC	0002h	Miscellaneous Configuration	Yes	Yes
3	Reserved	N/A	Reserved	No	No
4	LNKST	00C0h	Link Status LED	Yes	No
5	LED1	0084h	LED1 Status	Yes	No
6	LED2	0088h	LED2 Status	Yes	No
7	LED3	0090h	LED3 Status	Yes	No
8	Reserved	N/A	Reserved	No	No
9	FDC	0000h	Full-Duplex Control	Yes	Yes
10–15	Reserved	N/A	Reserved	No	No
16	IOBASEL	N/A	Reserved	Yes	Yes
17	IOBASEU	N/A	Reserved	Yes	Yes
18	BSBC	9001h	Burst Size and Bus Control	Yes	Yes
19	EECAS	0002h	EEPROM Control and Status	Yes	No
20	SWS	0200h	Software Style	Yes	No
21	INTCON	N/A	Reserved	Yes	Yes
22	PCILAT	FF06h	PCI Latency	Yes	Yes

ABSOLUTE MAXIMUM RATINGS

Storage Temperature . . . . –65°C to +150°C

Ambient Temperature Under Bias . . –65°C to +125°C

Supply Voltage to AV_{SS} or V_{SSB}
(AV_{DD}, V_{DD}, V_{DDB}) . . . . 0.3 V to +6.0V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A) . . . . 0°C to + 70°C

Industrial (I) Devices

Ambient Temperature (T_A) . . . . –40°C to + 85°C

Supply Voltages

(AV_{DD}, V_{DD}) . . . . +5V ± 5%
(V_{DDB} for 5 V Signaling) . . . . +5V ± 5%
(V_{DDB} for 3.3 V Signaling). . . . . + 3.3 V ± 10%

All inputs within the range: . . . $AV_{SS} – 0.5\ V \le V_{IN} \le$
. . . . $AV_{DD} + 0.5$ V, or
. . . . $V_{SS} – 0.5\ V \le V_{IN} \le V_{DD} + 0.5$ V, or
. . . . $V_{SSB} – 0.5 < V_{IN} < V_{DDB} + 0.5$V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL and INDUSTRIAL operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
Digital Input Voltage for 5 V Signaling					
V_{IL}	Input LOW Voltage			0.8	V
V_{IH}	Input HIGH Voltage		2.0		V
Digital Output Voltage for 5 V Signaling					
V_{OL}	Output LOW Voltage	I_{OL1} = 3 mA I_{OL2} = 6 mA I_{OL3} = 12 mA (Note 1)		0.45	V
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH} = –2 mA (Note 3)	2.4		V
Digital Input Leakage Current for 5 V Signaling					
I_{IX}	Input Low Leakage Current (Note 4)	V_{IN} = 0 V, $V_{DD} = V_{DDB}$ = 5 V	–10	10	μA
Digital Output Leakage Current for 5 V Signaling					
I_{OZL}	Output Low Leakage Current (Note 5)	V_{OUT} = 0.4V		–10	μA
I_{OZH}	Output High Leakage Current (Note 5)	$V_{OUT} = V_{DD}$, V_{DDB}		10	μA
Digital Input Voltage for 3.3 V Signaling					
V_{IL}	Input LOW Voltage		–0.5	0.325 V_{DDB}	V
V_{IH}	Input HIGH Voltage		0.475 V_{DDB}	V_{DDB} + 0.5	V
Digital Output Voltage for 3.3 V Signaling					
V_{OL}	Output LOW Voltage	I_{OL} = 1.5 mA		0.1 V_{DDB}	V
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH} = –0.5 mA	0.9 V_{DDB}		V
Digital Input Leakage Current for 3.3 V Signaling					
I_{IX}	Input Low Leakage Current	V_{IN} = 0 V, $V_{DD} = V_{DDB}$ = 3.3 V (Note 4)	–10	10	μA
Digital Output Leakage Current for 3.3 V Signaling					
I_{OZL}	Output Low Leakage Current (Note 5)	V_{OUT} = 0.4V		–10	μA
I_{OZH}	Output High Leakage Current (Note 5)	$V_{OUT} = V_{DD}$, V_{DDB}		10	μA

DC CHARACTERISTICS over COMMERCIAL and INDUSTRIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions		Min	Max	Units
Crystal Input Current						
V_{ILX}	XTAL1 Input LOW Voltage Threshold	V_{IN} = External Clock		–0.5	0.8	V
V_{IHX}	XTAL1 Input HIGH Voltage Threshold	V_{IN} = External Clock		VDD – 0.8	VDD + 0.5	V
I_{ILX}	XTAL1 Input LOW Current	V_{IN} = External Clock	Active	–120	0	µA
		V_{IN} = VSS	Sleep	–10	+10	µA
I_{IHX}	XTAL1 Input HIGH Current	V_{IN} = External Clock	Active	0	120	µA
		V_{IN} = VDD	Sleep		400	µA
Power Supply Current						
I_{DD}	Active Power Supply Current	XTAL1 = 20 MHz, CLK = 33 MHz			90	mA
I_{DDCOMA}	Sleep Mode Power Supply Current	SLEEP active AWAKE = 0 (BCR2, bit 2)			200	µA
$I_{DDSNOOZE}$	Auto Wake Mode Power Supply Current	SLEEP active AWAKE = 1 (BCR2, bit 2)			10	mA
$I_{DDMAGIC0}$	Magic Packet Mode Power Supply Current	CLK = 0 MHz (Note 10)			47	mA
$I_{DDMAGIC33}$	Magic Packet Mode Power Supply Current	CLK = 33 MHz (Note 10)			80	mA
Pin Capacitance						
C_{IN}	Input Pin Capacitance	FC = 1 MHz (Note 6)			10	pF
C_{IDSEL}	IDSEL Pin Capacitance	FC = 1 MHz (Note 6)			8	pF
C_O	I/O or Output Pin Capacitance	FC = 1 MHz (Note 6)			10	pF
C_{CLK}	CLK Pin Capacitance	FC = 1 MHz (Note 6)		5	12	pF
Twisted Pair Interface (10BASE-T)						
I_{IRXD}	Input Current at RXD±	$AV_{SS} < V_{IN} < AV_{DD}$		–500	500	µA
R_{RXD}	RXD± Differential Input Resistance			10		KΩ
V_{TIVB}	RXD+, RXD– Open Circuit I_{IN} = 0 mA Input Voltage (Bias)			AV_{DD}–3.0	AV_{DD}–1.5	V
V_{TIDV}	Differential Mode Input Voltage Range (RXD±)	AV_{DD} = 5.0 V		–3.1	3.1	V
V_{TSQ+}	RXD Positive Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤ 10 MHz LRT = 0 (CSR15, bit 9)		300	520	mV
V_{TSQ-}	RXD Negative Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤ 10 MHz LRT = 0 (CSR15, bit 9)		–520	–300	mV
V_{THS+}	RXD Post-Squelch Positive Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤ 10 MHz LRT = 0 (CSR15, bit 9)		150	293	mV
V_{THS-}	RXD Post-Squelch Negative Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤ 10 MHz LRT = 0 (CSR15, bit 9)		–293	–150	mV
V_{LTSQ+}	RXD Positive Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤ 10 MHz LRT = 1 (CSR15, bit 9)		180	312	mV
V_{LTSQ-}	RXD Negative Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤ 10 MHz LRT = 1 (CSR15, bit 9)		–312	–180	mV
V_{LTHS+}	RXD Post-Squelch Positive Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤ 10 MHz LRT = 1 (CSR15, bit 9)		90	176	mV
V_{LTHS-}	RXD Post-Squelch Negative Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤ 10 MHz LRT = 1 (CSR15, bit 9)		–176	–90	mV

DC CHARACTERISTICS over COMMERCIAL and INDUSTRIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
Twisted Pair Interface (10BASE-T) (Continued)					
V_{RXDTH}	RXD Switching Threshold	(Note 6)	–35	35	mV
V_{TXH}	TXD± and TXP± Output HIGH Voltage	AV_{SS}= 0 V	AV_{DD}–0.6	AV_{DD}	V
V_{TXL}	TXD± and TXP± Output LOW Voltage	AV_{DD} = 5 V	AV_{SS}	AV_{SS}+0.6	V
V_{TXI}	TXD± and TXP± Differential Output Voltage Imbalance		–40	40	mV
V_{TXOFF}	TXD± and TXP± Idle Output Voltage			40	mV
R_{TX}	TXD±, TXP± Differential Driver Output Impedance	(Note 6)		80	Ω
Attachment Unit Interface (AUI)					
I_{IAXD}	Input Current at DI+ and DI–	–1V < V_{IN} < AV_{DD} + 0.5 V	–500	+500	µA
I_{IAXC}	Input Current at CI+ and CI–	–1V < V_{IN} < AV_{DD} + 0.5 V	–500	+500	µA
V_{AOD}	Differential Output Voltage |(DO+)–(DO–)|	R_L = 78 Ω	630	1200	mV
V_{AODOFF}	Transmit Differential Output Idle Voltage	R_L = 78 Ω (Note 9)	–40	40	mV
I_{AODOFF}	Transmit Differential Output Idle Current	R_L = 78 Ω (Note 8)	–1	1	mA
V_{CMT}	Transmit Output Common Mode Voltage	R_L = 78 Ω	2.5	AV_{DD}	V
V_{ODI}	DO± Transmit Differential Output Voltage Imbalance	R_L = 78 Ω (Note 7)		25	mV
V_{ATH}	Receive Data Differential Input Threshold		–35	35	mV
V_{ASQ}	DI± and CI± Differential Input Threshold (Squelch)	–275	–160	mV	
V_{IRDVD}	DI± and CI± Differential Mode Input Voltage Range	–1.5	1.5	V	
V_{ICM}	DI± and CI± Input Bias Voltage	I_{IN} = 0 mA	AV_{DD}–3.0	AV_{DD}–1.0	V
V_{OPD}	DO± Undershoot Voltage at ZERO Differential on Transmit Return to ZERO (ETD)	(Note 9)		–100	mV

Notes:

1. *I_{OL1} applies to AD[31:0], C/$\overline{\text{BE}}$[3:0], PAR and $\overline{\text{REQ}}$.*
   I_{OL2} applies to $\overline{\text{DEVSEL}}$, $\overline{\text{FRAME}}$, $\overline{\text{INTA}}$, $\overline{\text{IRDY}}$, $\overline{\text{PERR}}$, $\overline{\text{SERR}}$, $\overline{\text{STOP}}$, $\overline{\text{TRDY}}$, EECS, ERA[7:0], ERACLK, $\overline{\text{EROE}}$, DXCVR/NOUT, ERD7/TXDAT, ERD6/TXEN and TDO.
   I_{OL3} applies to EESK/$\overline{\text{LED1}}$/SFBD, $\overline{\text{LED2}}$/SRDCLK, EEDO/$\overline{\text{LED3}}$/SRD, and EEDI/$\overline{\text{LNKST}}$.
2. *V_{OH} does not apply to open-drain output pins.*
3. *Outputs are CMOS and will be driven to rail if the load is not resistive.*
4. *I_{IX} applies to all input pins except XTAL1.*
5. *I_{OZL} and I_{OZH} apply to all three-state output pins and bi-directional pins.*
6. *Parameter not tested. Value determined by characterization.*
7. *Tested, but to values in excess of limits. Test accuracy not sufficient to allow screening guard bands.*
8. *Correlated to other tested parameters—-not tested directly.*
9. *Test not implemented to data sheet specification.*
10. *The power supply current in Magic Packet mode is linear. For example, at CLK = 20 MHz the maximum Magic Packet mode power supply current would be 67 mA.*

SWITCHING CHARACTERISTICS: Bus Interface (Unless otherwise noted, parametric values are the same between Commercial devices and Industrial devices.)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Clock Timing					
F_{CLK}	CLK Frequency		0	33	MHz
t_{CYC}	CLK Period	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V	30	∝	ns
t_{HIGH}	CLK High Time	@ 2.0 V for V_{DDB} = 5 V @ 0.475 V_{DDB} for V_{DDB} = 3.3 V	12		ns
t_{LOW}	CLK Low Time	@ 0.8 V for V_{DDB} = 5 V @ 0.325 V_{DDB} for V_{DDB} = 3.3 V	12		ns
t_{FALL}	CLK Fall Time	Over 2 V p-p for V_{DDB} = 5 V Over 0.4 V_{DDB} p–p for V_{DDB} = 3.3 V (Note 1)	1	4	V/ns
t_{RISE}	CLK Rise Time	Over 2 V p-p for V_{DDB} = 5 V Over 0.4 V_{DDB} p–p for V_{DDB} = 3.3 V (Note 1)	1	4	V/ns
Output and Float Delay Timing					
t_{VAL}	AD[31:00], C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$, $\overline{\text{PERR}}$, $\overline{\text{SERR}}$ Valid Delay	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V	2	11	ns
t_{VAL} ($\overline{\text{REQ}}$)	$\overline{\text{REQ}}$ Valid Delay	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V	2	12	ns
t_{ON}	AD[31:00], C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$ Active Delay	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V	2	11	ns
t_{OFF}	AD[31:00], C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$ Float Delay	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V		28	ns
Setup and Hold Timing					
t_{SU}	AD[31:00], C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{LOCK}}$, $\overline{\text{DEVSEL}}$, IDSEL Setup Time	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V	7		ns
t_H	AD[31:00], C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{LOCK}}$, $\overline{\text{DEVSEL}}$, IDSEL Hold Time	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V	0		ns
t_{SU} ($\overline{\text{GNT}}$)	$\overline{\text{GNT}}$ Setup Time	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V	10		ns
t_H ($\overline{\text{GNT}}$)	$\overline{\text{GNT}}$ Hold Time	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DDB} for V_{DDB} = 3.3 V	0		ns

SWITCHING CHARACTERISTICS: Bus Interface (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
EEPROM Timing					
f_{EESK} (EESK)	EESK Frequency	@ 1.5 V for V (Note 2)		650	KHz
t_{HIGH} (EESK)	EESK High Time	@ 0.2 V	780		ns
t_{LOW}	EESK Low Time	@ 0.8 V	780		ns
t_{VAL} (EEDI)	EEDI Valid Output Delay from EESK	@ 1.5 V for V (Note 2)	–15	15	ns
t_{VAL} (EESK)	EECS Valid Output Delay from EESK	@ 1.5 V for V (Note 2)	–15	15	ns
t_{LOW} (EECS)	EECS Low Time	@ 1.5 V for V (Note 2)	1550		ns
t_{SU} (EEDO)	EEDO Setup Time to EESK	@ 1.5 V for V (Note 2)	50		ns
t_{H} (EEDO)	EEDO Hold Time from EESK	@ 1.5 V for V (Note 2)	0		ns
Expansion ROM Interface Timing					
t_{VAL} (ERA)	ERA Valid Delay from CLK	@ 1.5 V			ns
t_{VAL} (EROE)	EROE Valid Delay from CLK	@ 1.5 V			ns
t_{VAL} (ERACLK)	ERACLK Valid Delay from CLK	@ 1.5 V			ns
t_{SU} (ERD)	ERD Setup Time to CLK	@ 1.5 V			ns
t_{H} (ERD)	ERD Hold Time to CLK	@ 1.5 V			ns
JTAG (IEEE 1149.1) Test Signal Timing					
t_{J1}	TCK Frequency			10	MHz
t_{J2}	TCK Period		100		
t_{J3}	TCK High Time	@ 2.0 V	45		ns
t_{J4}	TCK Low Time	@ 0.8 V	45		ns
t_{J5}	TCK Rise Time			4	ns
t_{J6}	TCK Fall Time			4	ns
t_{J7}	TDI, TMS Setup Time		8		ns
t_{J8}	TDI, TMS Hold Time		10		ns
t_{9}	TDO Valid Delay		3	30	ns
t_{J9}	TDO Float Delay			50	ns
t_{J11}	All Outputs (Non-Test) Valid Delay		3	25	ns
t_{J12}	All Outputs (Non-Test) Float Delay			36	ns
t_{J13}	All Outputs (Non-Test) Setup Time		8		ns
t_{J4}	All Outputs (Non-Test) Hold Time		7		ns

Note:

1. *Not tested; parameter guaranteed by characterization.*
2. *Parameter value is given for automatic EEPROM read operation. When EEPROM port (BCR19) is used to access the EEPROM, software is responsible for meeting EEPROM timing requirements.*

SWITCHING CHARACTERISTICS: 10BASE-T Interface (Unless otherwise noted, parametric values are the same between Commercial devices and Industrial devices.)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{TETD}	Transmit Start of Idle		250	350	ns
t_{TR}	Transmitter Rise Time	(10% to 90%)		5.5	ns
t_{TF}	Transmitter Fall Time	(90% to 10%)		5.5	ns
t_{TM}	Transmitter Rise and Fall Time Mismatch	($t_{TM} = \lvert t_{TR} - t_{TF} \rvert$)		1	ns
t_{XMTON}	XMT Asserted Delay			100	ns
t_{XMTOFF}	XMT Deasserted Delay		20	62	ms
t_{PERLP}	Idle Signal Period		8	24	ms
t_{PWLP}	Idle Link Pulse Width	(Note 1)	75	120	ns
t_{PWPLP}	Predistortion Idle Link Pulse Width	(Note 1)	45	55	ns
t_{JA}	Transmit Jabber Activation Time		20	150	ms
t_{JR}	Transmit Jabber Reset Time		250	750	ms
t_{JREC}	Transmit Jabber Recovery Time (Minimum time gap between transmitted frames to prevent jabber activation)		1.0		µs
Receiving Timing					
t_{PWNRD}	RXD Pulse Width Not to Turn Off Internal Carrier Sense	$V_{IN} > V_{THS}$ (min)	136		ns
t_{PWROFF}	RXD Pulse Width To Turn Off	$V_{IN} > V_{THS}$ (min)		200	ns
t_{RETD}	Receive Start of Idle		200		ns
t_{RCVON}	RCV Asserted Delay		TRON –50	TRON +100	ns
t_{RCVON}	RCV Deasserted Delay		20	62	ms
Collision Detection and SQE Test					
t_{COLON}	COL Asserted Delay		750	900	ns
t_{COLOFF}	COL Deasserted Delay		20	62	ms

Note:

1. Not tested; parameter guaranteed by characterization.

SWITCHING CHARACTERISTICS: AUI (Unless otherwise noted, parametric values are the same between Commercial devices and Industrial devices.)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
AUI Port					
t_{DOTR}	DO+, DO– Rise Time (10% to 90%)		2.5	5.0	ns
t_{DOTF}	DO+, DO– Fall Time (10% to 90%)		2.5	5.0	ns
t_{DORM}	DO+, DO– Rise and Fall Time Mismatch			1.0	ns
t_{DOETD}	DO± End of Transmission		200	375	ns
t_{PWODI}	DI Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|VASQ\|$ (Note 1)	15	45	ns
t_{PWKDI}	DI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|VASQ\|$ (Note 2)	136	200	ns
t_{PWOCI}	CI Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|VASQ\|$ (Note 3)	10	26	ns
t_{PWKCI}	CI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|VASQ\|$ (Note 4)	90	160	ns
Internal MENDEC Clock Timing					
t_{X1}	XTAL1 Period	V_{IN} = External Clock	49.995	50.001	ns
t_{X1H}	XTAL1 HIGH Pulse Width	V_{IN} = External Clock	20		ns
t_{X1L}	XTAL1 LOW Pulse Width	V_{IN} = External Clock	20		ns
t_{X1R}	XTAL1 Rise Time	V_{IN} = External Clock		5	ns
t_{X1F}	XTAL1 Fall Time	V_{IN} = External Clock		5	ns

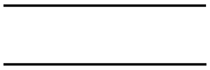
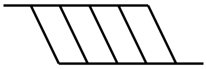
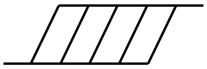
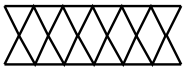
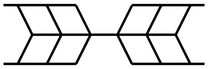
Note:

1. *DI pulses narrower than t_{PWODI} (min) will be rejected; pulses wider than t_{PWODI} (max) will turn internal DI carrier sense on.*
2. *DI pulses narrower than t_{PWKDI} (min) will maintain internal DI carrier sense on; pulses wider than t_{PWKDI} (max) will turn internal DI carrier sense off.*
3. *CI pulses narrower than t_{PWOCI} (min) will be rejected; pulses wider than t_{PWOCI} (max) will turn internal CI carrier sense on.*
4. *CI pulses narrower than t_{PWKCI} (min) will maintain internal CI carrier sense on; pulses wider than t_{PWKCI} (max) will turn internal CI carrier sense off.*

SWITCHING CHARACTERISTICS: EADI (Unless otherwise noted, parametric values are the same between Commercial devices and Industrial devices.)

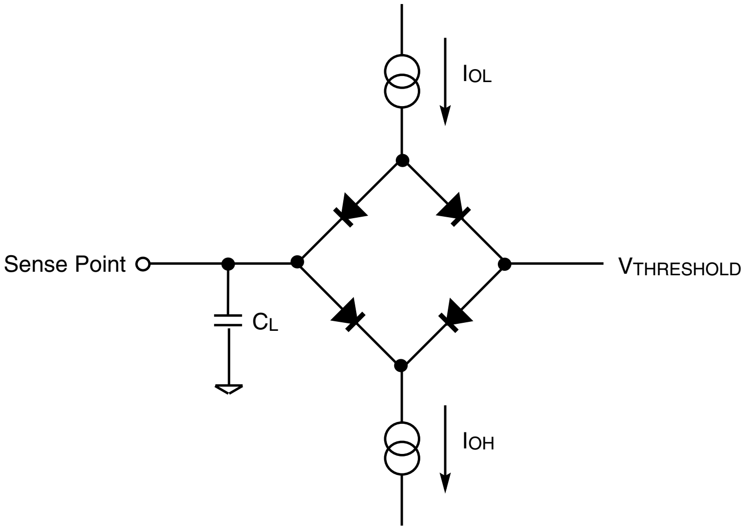
Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
t_{EAD1}	SRD Setup to SRDCLK	@ 1.5 V	40		ns
t_{EAD2}	SRD Hold to SRDCLK	@ 1.5 V	40		ns
t_{EAD3}	SF/BD Change to SRDCLK	@ 1.5 V	–15	+15	ns
t_{EAD4}	$\overline{\text{EAR}}$ Deassertion to SRDCLK (First Rising Edge)	@ 1.5 V	50		ns
t_{EAD5}	$\overline{\text{EAR}}$ Assertion after SFD Event (Frame Rejection)	@ 1.5 V	200	51,090	ns
t_{EAD6}	$\overline{\text{EAR}}$ Assertion	@ 1.5 V	110		ns

A KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance “Off” State

KS000010

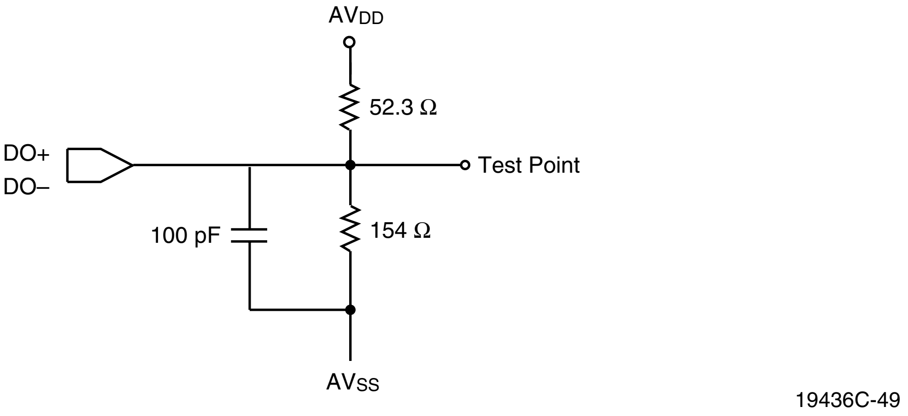
SWITCHING TEST CIRCUITS



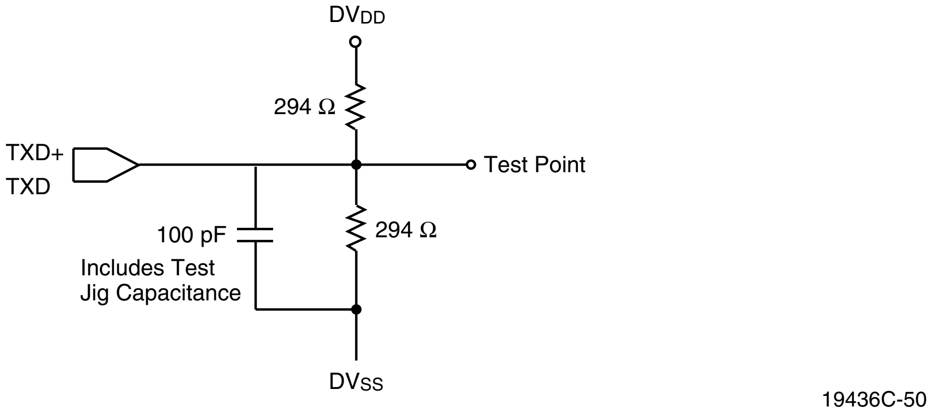
19436C-48

Normal and Tri-State OutPuts

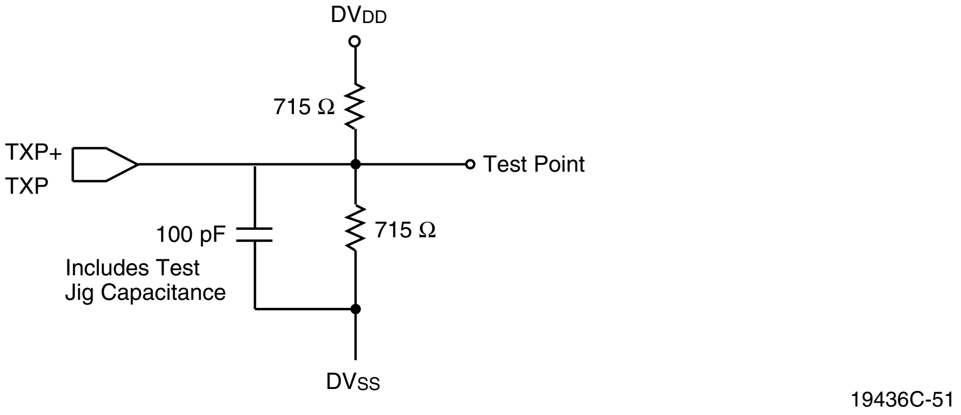
SWITCHING TEST CIRCUITS



AUI DO Switching Test Circuit

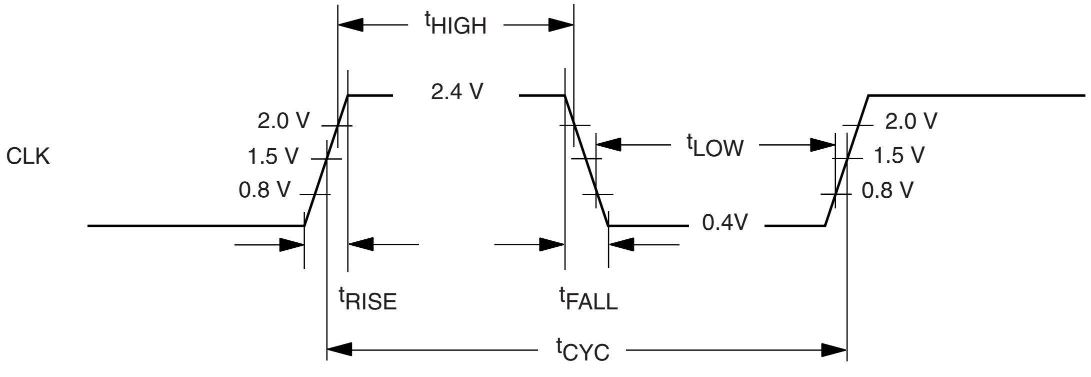


TXD Switching Test Circuit



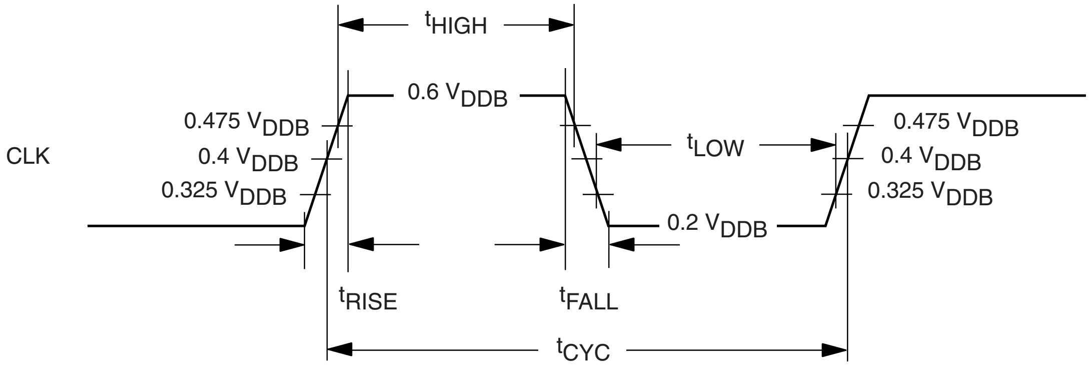
TXP Outputs Test Circuit

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE



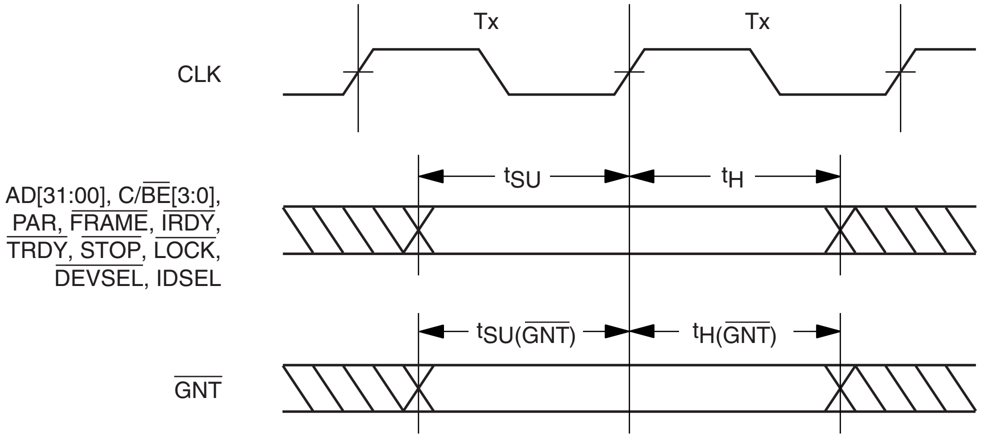
19436C-52

CLK Waveform for 5 V Signaling



19436C-53

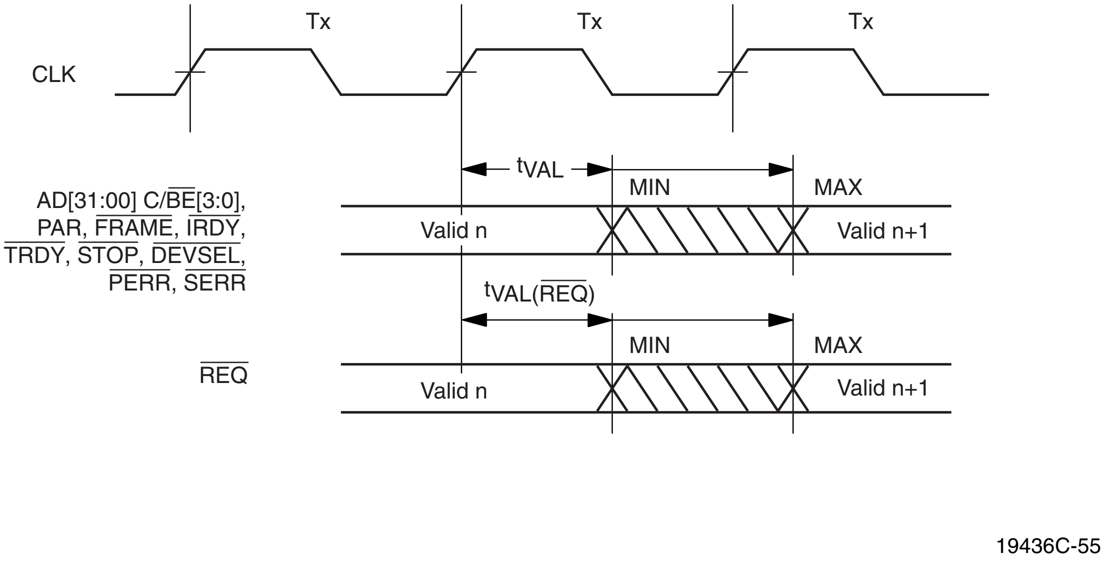
CLK Waveform for 3.3 V Signaling


Tx
Tx
CLK
t_{SU}
t_H
AD[31:00], C/$\overline{BE}$[3:0],
PAR, $\overline{FRAME}$, $\overline{IRDY}$,
$\overline{TRDY}$, $\overline{STOP}$, $\overline{LOCK}$,
$\overline{DEVSEL}$, IDSEL
$t_{SU(\overline{GNT})}$
$t_{H(\overline{GNT})}$
$\overline{GNT}$

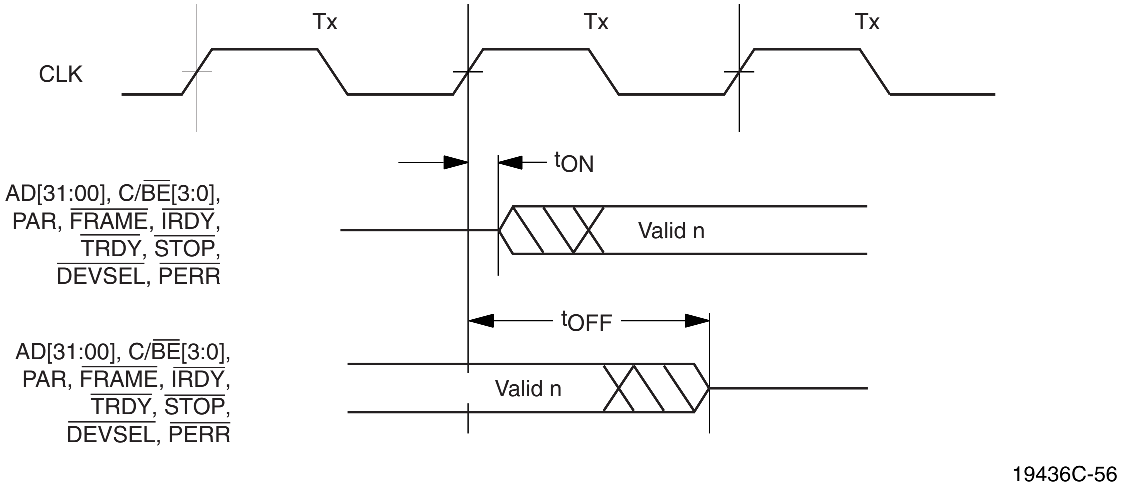
19436C-54

Input Setup and Hold Timing

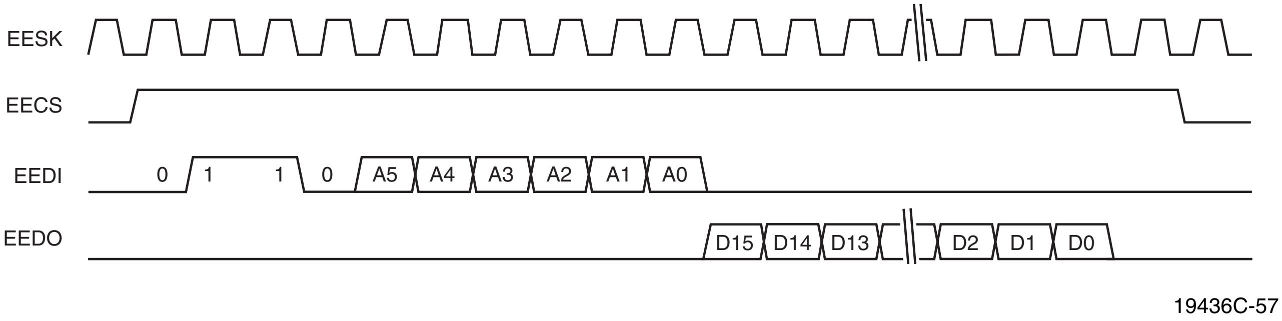
SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE



Output Valid Delay Timing

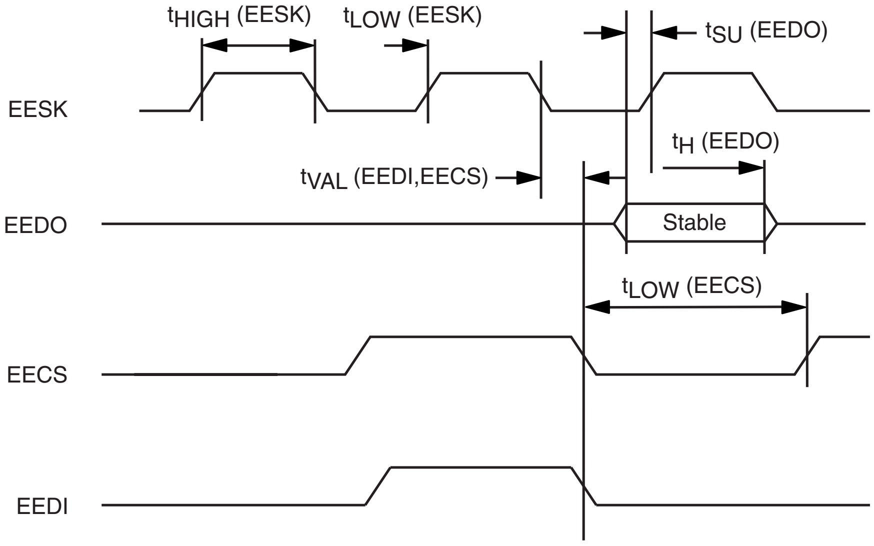

Tx
Tx
Tx
CLK
tON
AD[31:00], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, DEVSEL, PERR
Valid n
tOFF
AD[31:00], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, DEVSEL, PERR
Valid n
19436C-56

Output Tri-state Delay Timing



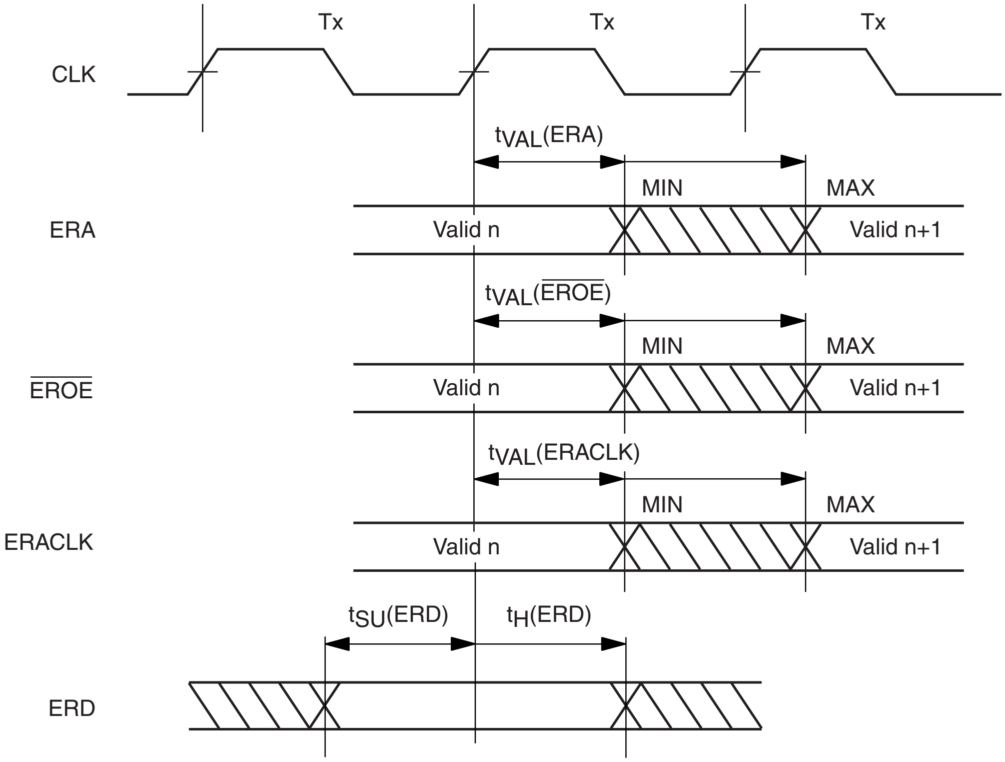
Automatic EEPROM Read Functional Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE



19436C-58

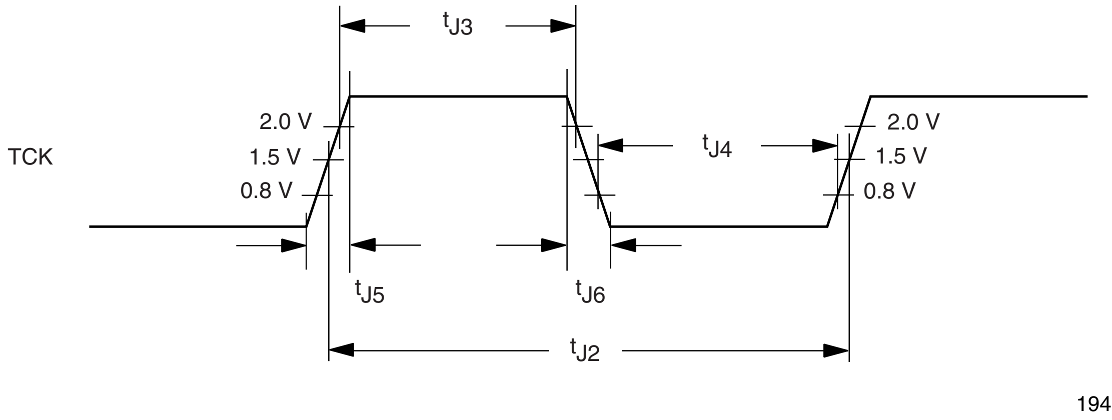
Automatic EEPROM Read Timing



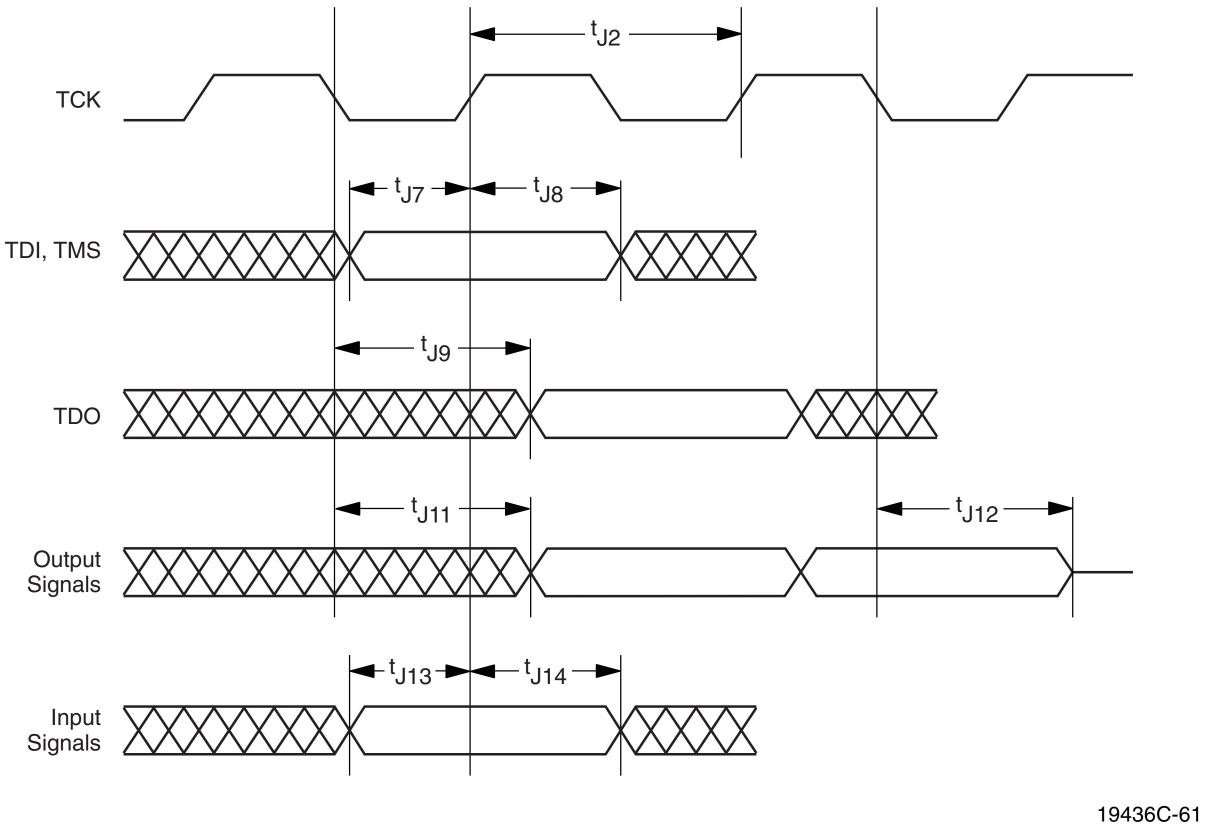
19436C-59

Expansion ROM Read Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE

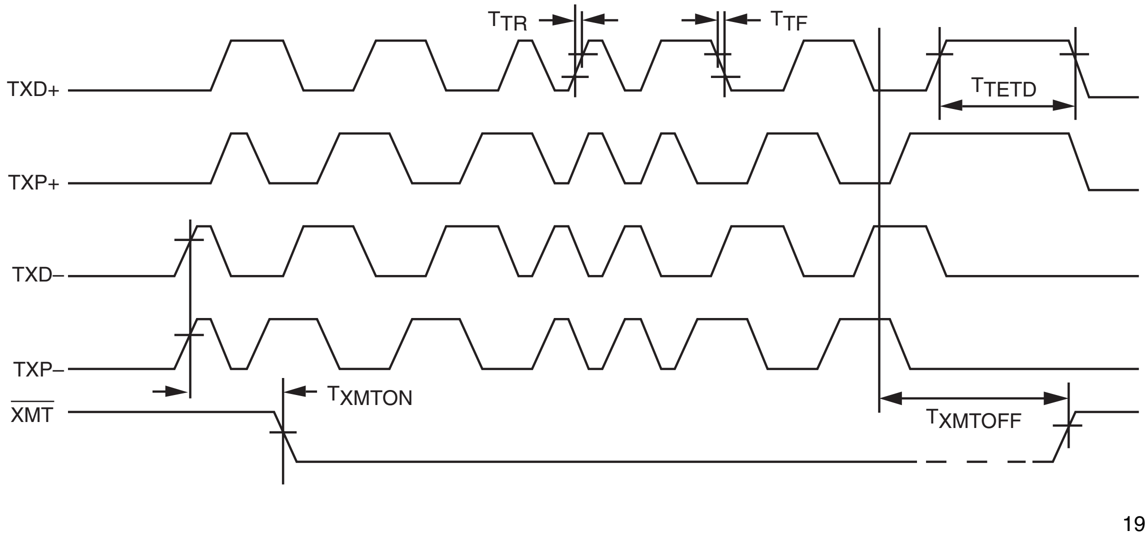


JTAG (IEEE 1149.1) TCK Waveform for 5 V Signaling

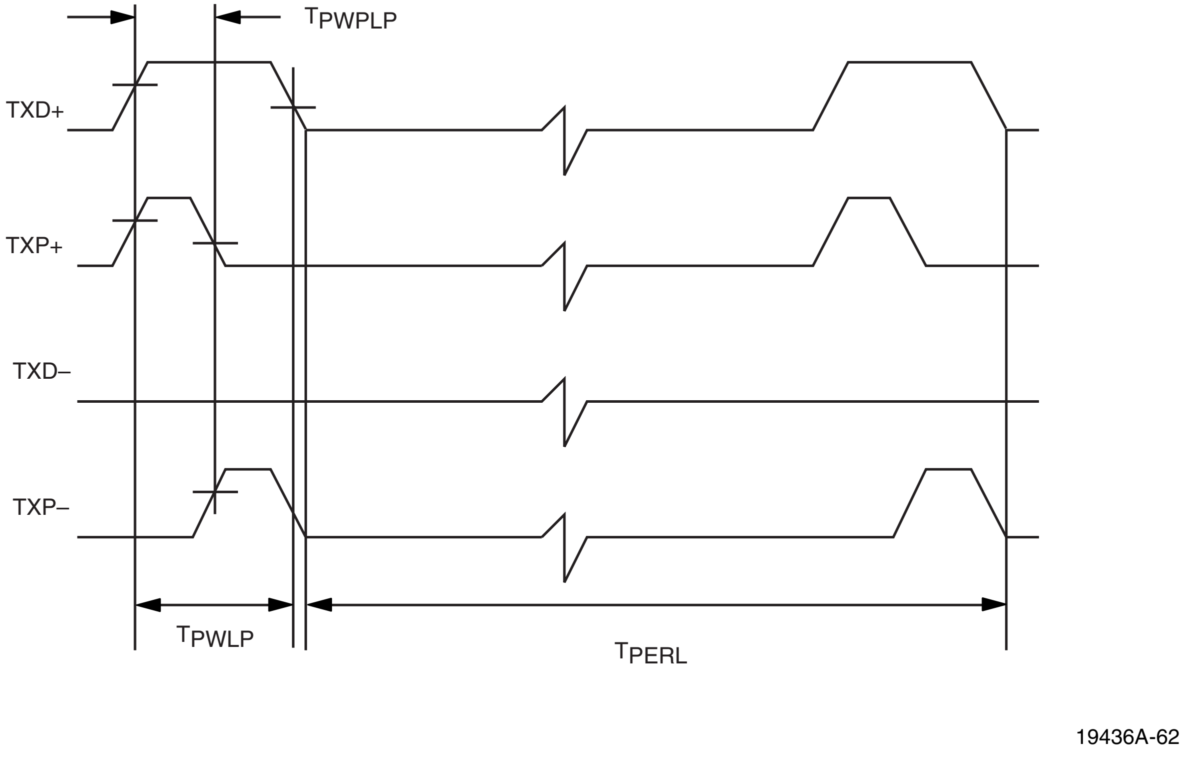


JTAG (IEEE 1149.1) Test Signal Timing

SWITCHING WAVEFORMS: 10BASE-T INTERFACE

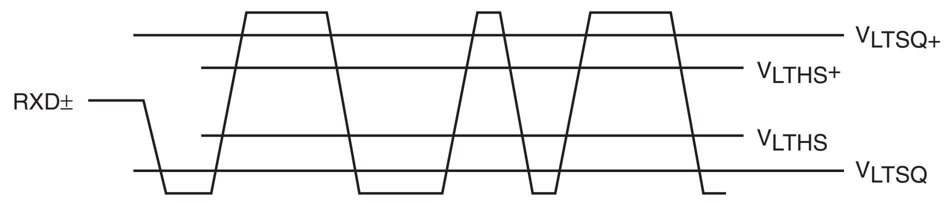


Transmit Timing



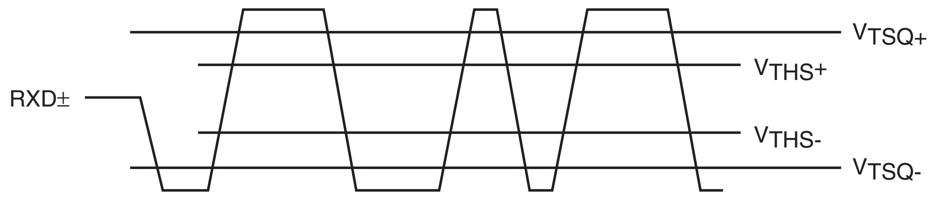
Idle Link Test Pulse

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



19436C-62

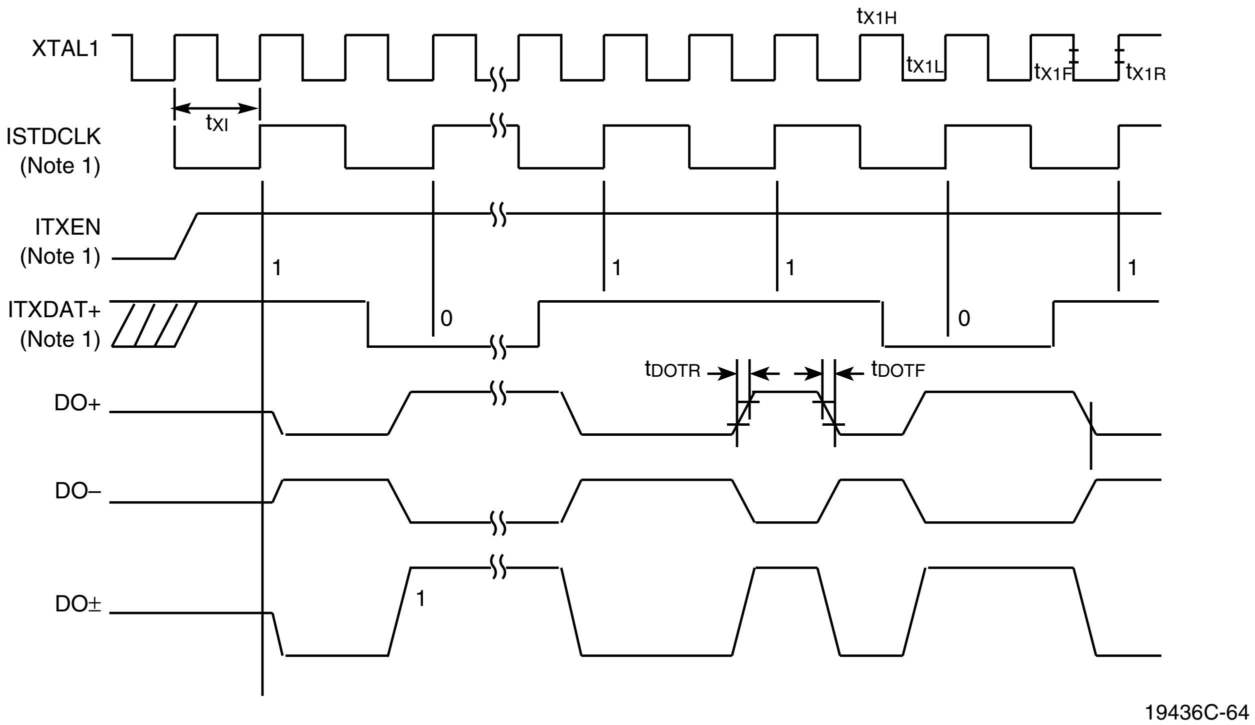
Receive Thresholds (LRT = 1)



19436C-63

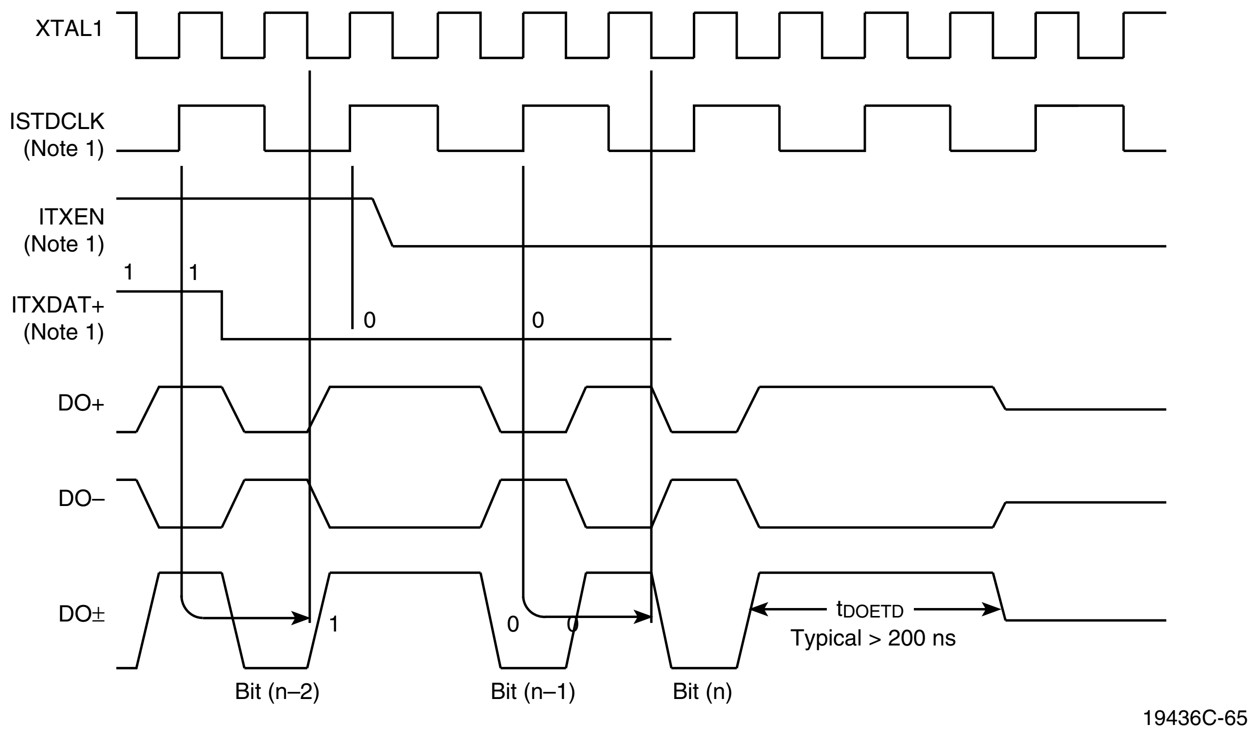
Receive Thresholds (LRT = 0)

SWITCHING WAVEFORMS: AUI



Note 1:
Internal signal and is shown for clarification only.

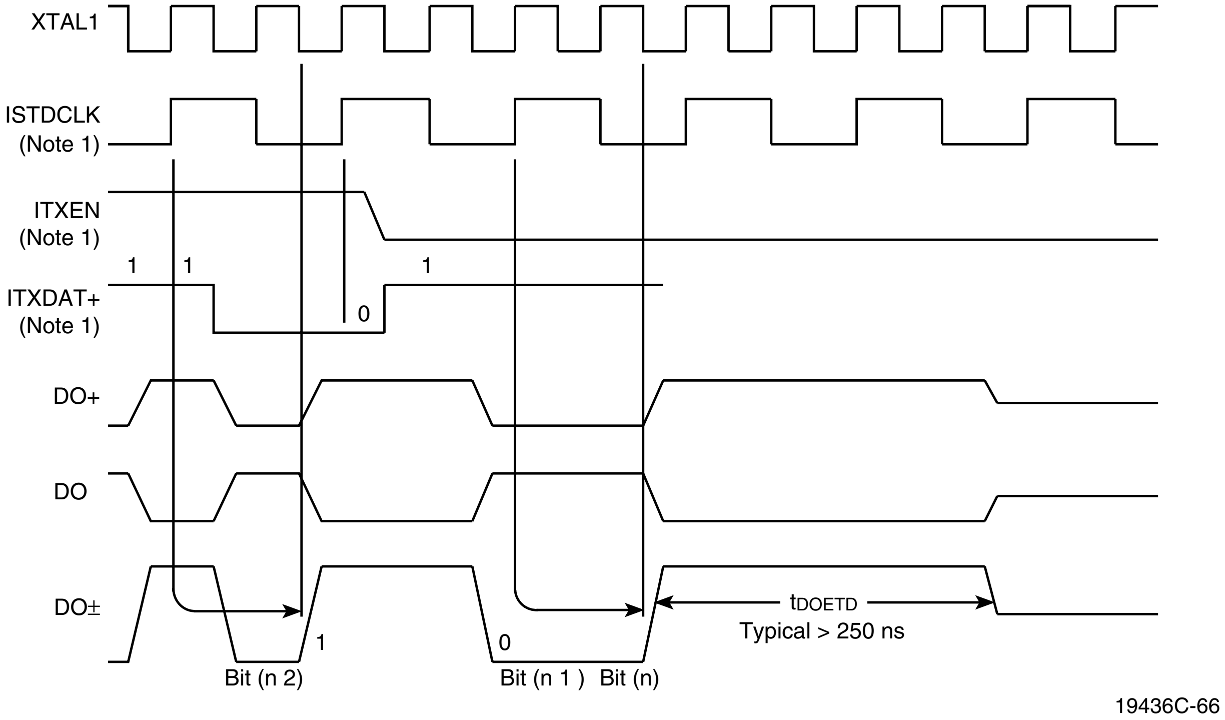
Transmit Timing—Start of Packet



Note 1:
Internal signal and is shown for clarification only.

Transmit Timing—End of Packet (Last Bit = 0)

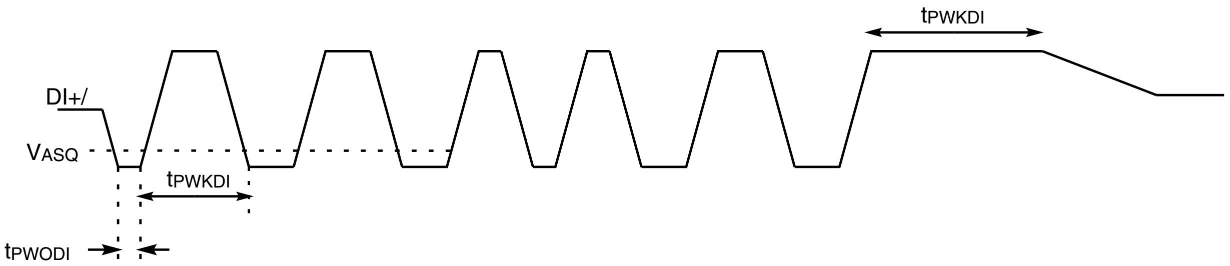
SWITCHING WAVEFORMS: AUI



Note 1:
Internal signal and is shown for clarification only.

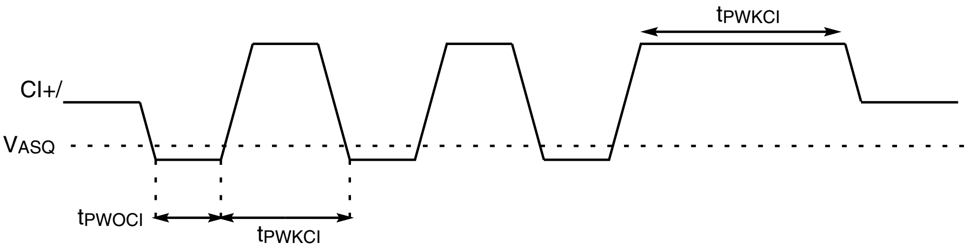
Transmit Timing—End of Packet (Last Bit = 1)

SWITCHING WAVEFORMS: AUI



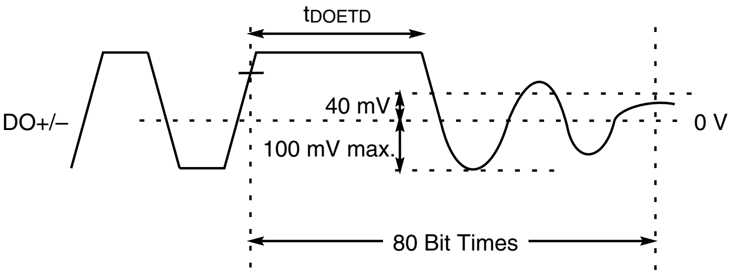
19436C-67

Receive Timing Diagram



19436C-68

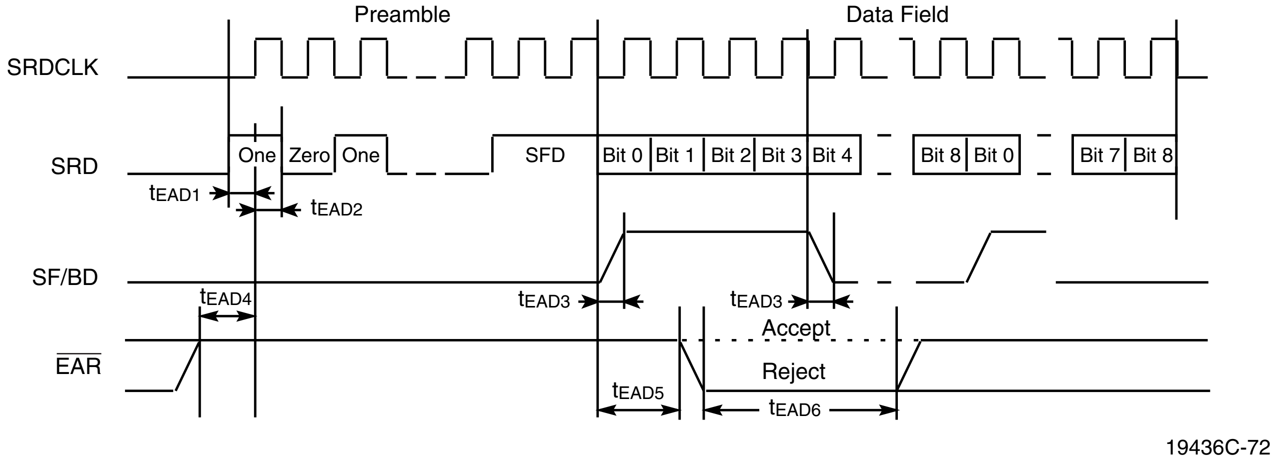
Collision Timing Diagram



19436C-69

Port DO ETD Waveform

SWITCHING WAVEFORMS: EADI

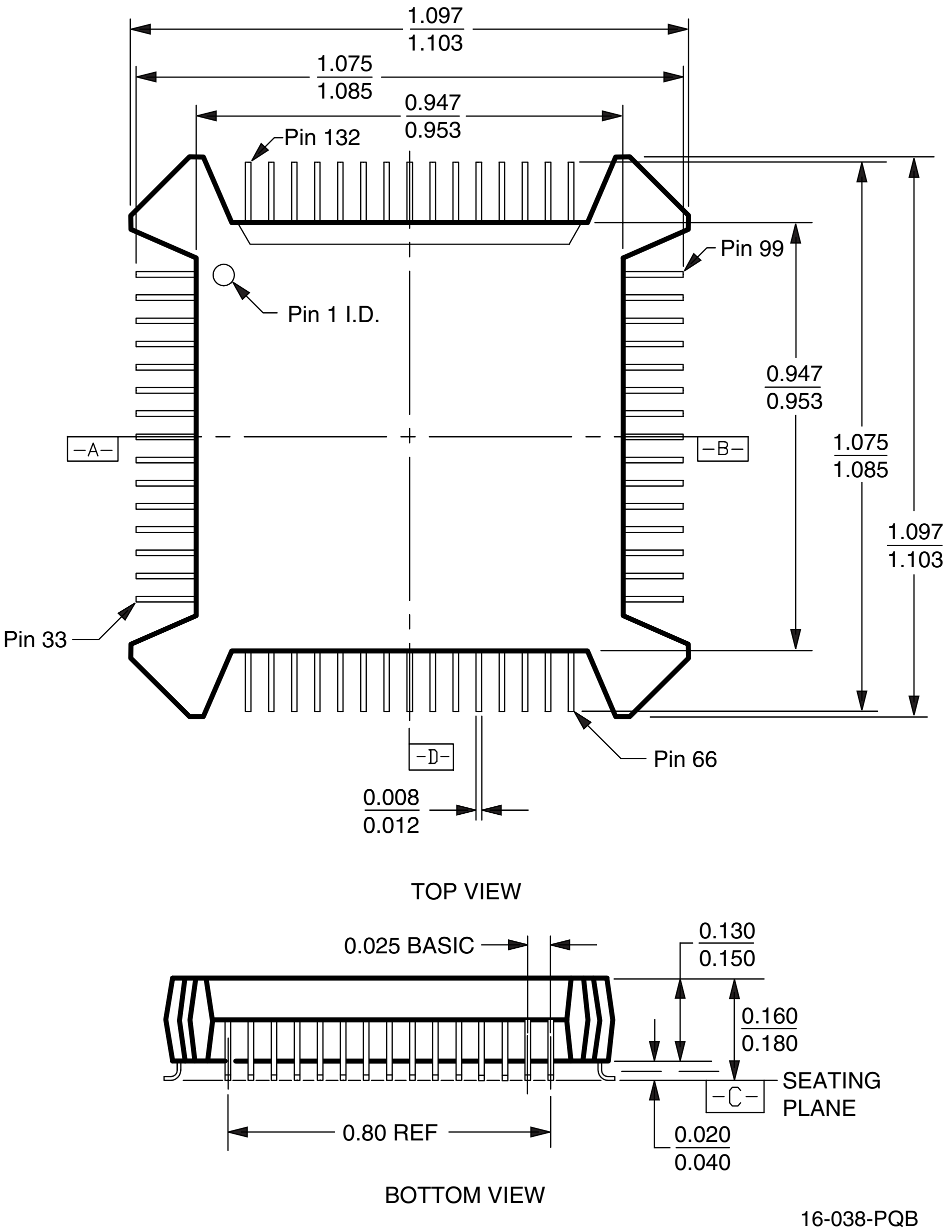


EADI Reject Timing

PHYSICAL DIMENSIONS*

PQB132

Plastic Quad Flat Pack, Trimmed and Formed (measured in inches)



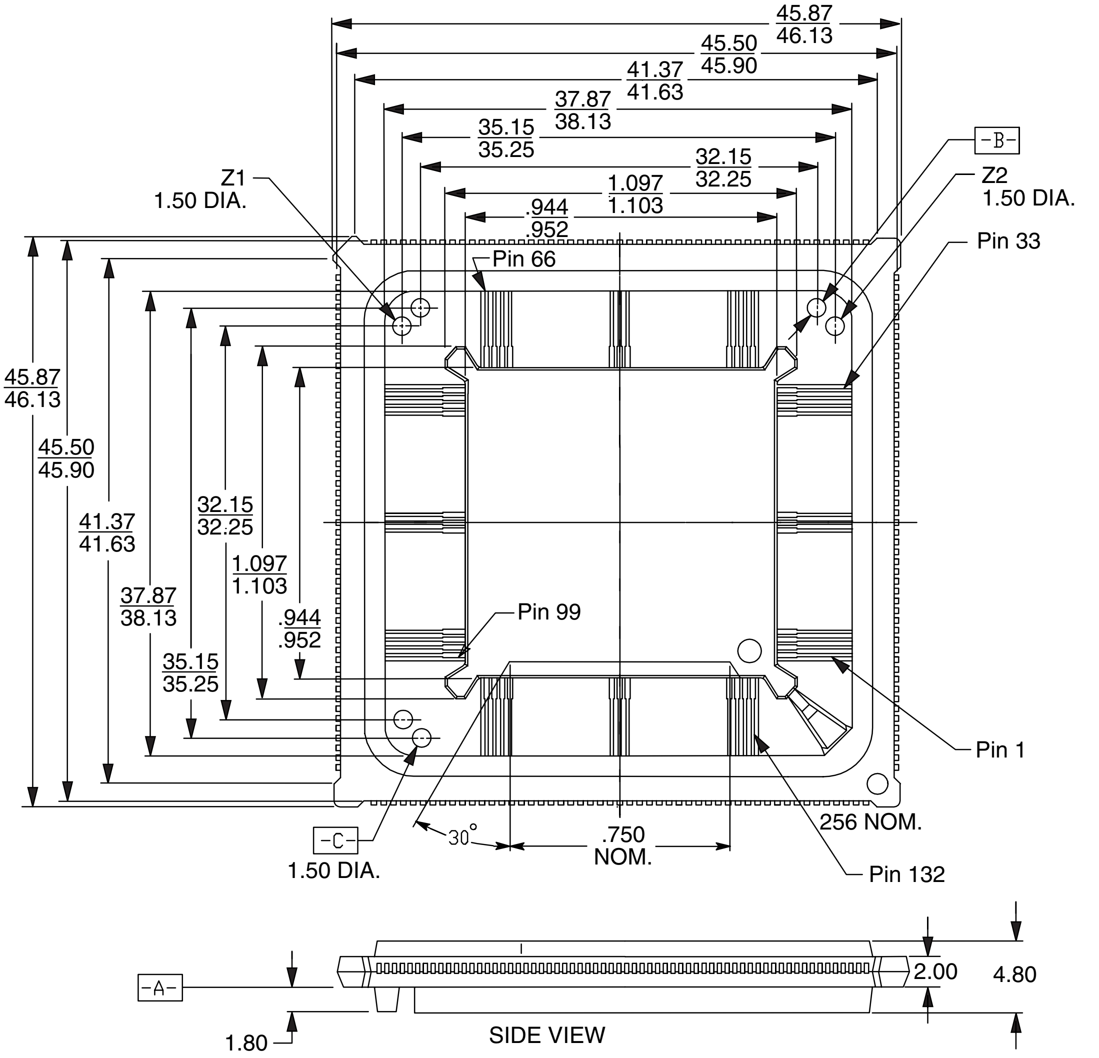
16-038-PQB
PQB132
DB87
7-26-94 ae

Trademarks

PHYSICAL DIMENSIONS

PQB132

Molded Carrier Ring Plastic Quad Flat Pack
(measured in inches, Ring measured in millimeters)

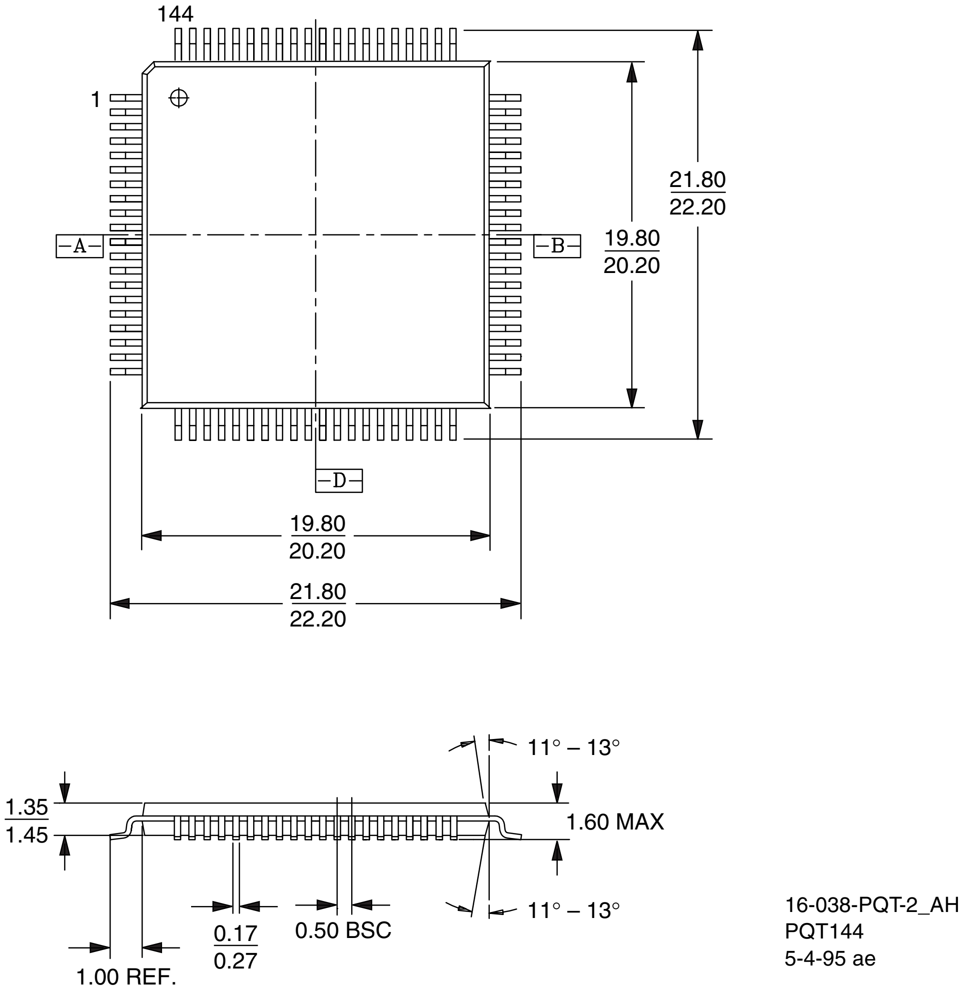


16-0000038-PQB-1
PQB132 (Molded)
DA84
6-14-94 ae

PHYSICAL DIMENSIONS*

PQT144

Thin Quad Flat Pack (measured in inches)



For reference only. BSC is an ANSI standard for Basic Space Centering.

PCnet-PCI II Compatible Media Interface Modules

PCnet-PCI II COMPATIBLE 10BASE-T FILTERS AND TRANSFORMERS

The table below provides a sample list of PCnet-PCI II compatible 10BASE-T filter and transformer modules available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Filters and Transformers	Filters Transformers and Choke	Filters Transformers Dual Chokes	Filters Transformers Resistors Dual Chokes
Bel Fuse	A556-2006-DE	16-pin 0.3 DIL	√			
Bel Fuse	0556-2006-00	14-pin SIP	√			
Bel Fuse	0556-2006-01	14-pin SIP			√	
Bel Fuse	0556-6392-00	16-pin 0.5 DIL			√	
Halo Electronics	FD02-101G	16-pin 0.3 DIL	√			
Halo Electronics	FD12-101G	16-pin 0.3 DIL		√		
Halo Electronics	FD22-101G	16-pin 0.3 DIL			√	
PCA Electronics	EPA1990A	16-pin 0.3 DIL	√			
PCA Electronics	EPA2013D	16-pin 0.3 DIL		√		
PCA Electronics	EPA2162	16-pin 0.3 SIP			√	
Pulse Engineering	PE-65421	16-pin 0.3 DIL	√			
Pulse Engineering	PE-65434	16-pin 0.3 SIL			√	
Pulse Engineering	PE-65445	16-pin 0.3 DIL			√	
Pulse Engineering	PE-65467	12-pin 0.5 SMT				√
Valor Electronics	PT3877	16-pin 0.3 DIL	√			
Valor Electronics	FL1043	16-pin 0.3 DIL			√	

PCnet-PCI II Compatible AUI Isolation Transformers

The table below provides a sample list of PCnet-PCI II compatible AUI isolation transformers available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3 DIL	50 µH
Bel Fuse	S553-0756-AE	16-pin 0.3 SMD	75 µH
Halo Electronics	TD01-0756K	16-pin 0.3 DIL	75 µH
Halo Electronics	TG01-0756W	16-pin 0.3 SMD	75 µH
PCA Electronics	EP9531-4	16-pin 0.3 DIL	50 µH
Pulse Engineering	PE64106	16-pin 0.3 DIL	50 µH
Pulse Engineering	PE65723	16-pin 0.3 SMT	75 µH
Valor Electronics	LT6032	16-pin 0.3 DIL	75 µH
Valor Electronics	ST7032	16-pin 0.3 SMD	75 µH

PCnet-PCI II Compatible DC/DC Converters

The table below provides a sample list of PCnet-PCI II compatible DC/DC converters available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Voltage	Remote On/Off
Halo Electronics	DCU0-0509D	24-pin DIP	5/-9	No
Halo Electronics	DCU0-0509E	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1007P	24-pin DIP	5/-9	No
PCA Electronics	EPC1054P	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1078	24-pin DIP	5/-9	Yes
Valor Electronics	PM7202	24-pin DIP	5/-9	No
Valor Electronics	PM7222	24-pin DIP	5/-9	Yes

MANUFACTURER CONTACT INFORMATION

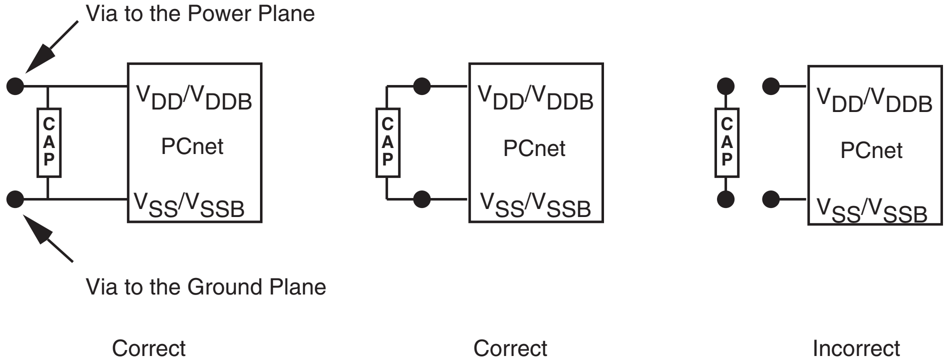
Contact the following companies for further information on their products.

Company	U.S. and Domestic	Asia	Europe
Bel Fuse	Phone: (201) 432-0463 FAX: (201) 432-9542	852-328-5515 852-352-3706	33-1-69410402 33-1-69413320
Halo Electronics	Phone: (415) 969-7313 FAX: (415) 367-7158	65-285-1566 65-284-9466	
PCA Electronics (HPC in Hong Kong)	Phone: (818) 892-0761 FAX: (818) 894-5791	852-553-0165 852-873-1550	33-1-44894800 33-1-42051579
Pulse Engineering	Phone: (619) 674-8100 FAX: (619) 675-8262	852-425-1651 852-480-5974	353-093-24107 353-093-24459
Valor Electronics	Phone: (619) 537-2500 FAX: (619) 537-2525	852-513-8210 852-513-8214	49-89-6923122 49-89-6926542

Recommendation for Power and Ground Decoupling

The mixed analog/digital circuitry in the PCnet-PCI II make it imperative to provide noise-free power and ground connections to the device. Without clean power and ground connections, a design may suffer from high bit error rates or may not function at all. Hence, it is highly recommended that the guidelines presented here are followed to ensure a reliable design.

Decoupling/Bypass Capacitors: Adequate decoupling of the power and ground pins and planes is required by all PCnet-PCI II designs. This includes both low-frequency bulk capacitors and high frequency capacitors. It is recommended that at least one low-frequency bulk (e.g. 22 µF) decoupling capacitor be used in the area of the PCnet-PCI II device. The bulk capacitor(s) should be connected directly to the power and ground planes. In addition, **at least 8** high frequency decoupling capacitors (e.g. 0.1 µF multilayer ceramic capacitors) should be used around the periphery of the PCnet-PCI II device to prevent power and ground bounce from affecting device operation. To reduce the inductance between the power and ground pins and the capacitors, the pins should be connected directly to the capacitors, rather than through the planes to the capacitors. The suggested connection scheme for the capacitors is shown in the figure below. Note also that the traces connecting these pins to the capacitors should be as wide as possible to reduce inductance (15 mils is desirable).

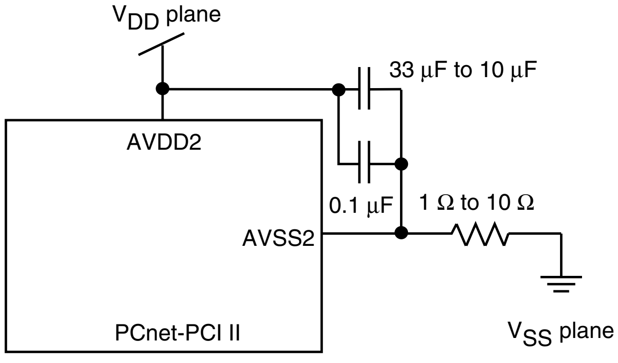


19436C-73

The most critical pins in the layout of a PCnet-PCI II design are the 4 analog power and 2 analog ground pins, AVDD[1–4] and AVSS[1–2], respectively. All of these pins are located in one corner of the device, the "analog corner." Specific functions and layout requirements of the analog power and ground pins are given below.

AVSS1 and AVDD3: These pins provide the power and ground for the Twisted Pair and AUI drivers. In addition AVSS1 serves as the ground for the logic interfaces in the 20 MHz Crystal Oscillator. Hence, these pins can be very noisy. A dedicated 0.1 µF capacitor between these pins is recommended.

AVSS2 and AVDD2: These pins are the most critical pins on the PCnet-PCI II device because they provide the power and ground for the phase-lock loop (PLL) portion of the chip. The voltage-controlled oscillator (VCO) portion of the PLL is sensitive to noise in the 60 kHz – 200 kHz. range. To prevent noise in this frequency range from disrupting the VCO, it is **strongly recommended** that the low-pass filter shown below be implemented on these pins.



19436C-74

To determine the value for the resistor and capacitor, the formula is:

*R * C Š 88*

Where R is in Ohms and C is in microfarads. Some possible combinations are given below. To minimize the voltage drop across the resistor, the R value should not be more than 10 W.

R	C
2.7 Ω	33 µF
4.3 Ω	22 µF
6.8 Ω	15 µF
10 Ω	10 µF

AVSS2 and AVDD2/AVDD4: These pins provide power and ground for the AUI and twisted pair receive circuitry. In addition, as mentioned earlier, AVSS2 and AVDD2 provide power and ground for the phase-lock loop portion of the chip. Except for the filter circuit already mentioned, no specific decoupling is necessary on these pins.

AVDD1: AVDD1 provides power for the control and interface logic in the PLL. Ground for this logic is provided by digital ground pins. No specific decoupling is necessary on this pin.

Special Note for Adapter Cards: In adapter card designs, **it is important to utilize all available powerand ground pins available on the bus edge connector**. In addition, the connection from the bus edge connector to the power or ground plane should be made through more than one via and with wide traces (15 mils desirable) wherever possible. Following these recommendations results in minimal inductance in the power and ground paths. By minimizing this inductance, ground bounce is minimized.

See also the PCnet Family Board Design and Layout Recommendations applications note (PID# 19595) for additional information.

Alternative Method for Initialization

The PCnet-PCI II controller may be initialized by performing I/O writes only. That is, data can be written directly to the appropriate control and status registers (CSR instead of reading from the initialization Block in memory). The registers that must be written are shown in the table below. These register writes are followed by writing the START bit in CSR0.

Control and Status Register	Comment
CSR2	IADR[31:16]*
CSR8	LADRF[15:0]
CSR9	LADRF[31:16]
CSR10	LADRF[47:32]
CSR11	LADRF[63:48]
CSR12	PADR[15:0]
CSR13	PADR[31:16]
CSR14	PADR[47:32]
CSR15	Mode
CSR24-25	BADR
CSR30-31	BADX
CSR47	POLLINT
CSR76	RCVRL
CSR78	XMTRL

Note:

1. *The INIT bit must not be set or the initialization block will be accessed instead.*

* *Needed only if SSIZE32 = 0.*

Look-Ahead Packet Processing (LAPP) Concept

Introduction of the LAPP Concept

A driver for the PCnet-PCI II controller would normally require that the CPU copy receive frame data from the controllers buffer space to the applications buffer space after the entire frame has been received by the control- ler. For applications that use a ping-pong windowing style, the traffic on the network will be halted until the current frame has been completely processed by the entire application stack. This means that the time be- tween last byte of a receive frame arriving at the clients Ethernet controller and the clients transmission of the first byte of the next outgoing frame will be separated by:

1. The time that it takes the clients CPUs interrupt procedure to pass software control from the current task to the driver
2. plus the time that it takes the client driver to pass the header data to the application and request an application buffer
3. plus the time that it takes the application to generate the buffer pointer and then return the buffer pointer to the driver
4. plus the time that it takes the client driver to transfer all of the frame data from the controllers buffer space into the applications buffer space and then call the application again to process the complete frame
5. plus the time that it takes the application to process the frame and generate the next outgoing frame
6. plus the time that it takes the client driver to set up the descriptor for the controller and then write a TDMD bit to CSR0

The sum of these times can often be about the same as the time taken to actually transmit the frames on the wire, thereby yielding a network utilization rate of less than 50%.

An important thing to note is that the PCnet-PCI II controllers data transfers to its buffer space are such that the system bus is needed by the PCnet-PCI II controller for approximately 4% of the time. This leaves 96% of the system bus bandwidth for the CPU to perform some of the inter-frame operations in advance of the completion of network receive activity, if possible. The question then becomes: how much of the tasks that need to be performed between reception of a frame and transmission of the next frame can be performed before the reception of the frame actually ends at the network, and how can the CPU be instructed to perform these tasks during the network reception time?

The answer depends upon exactly what is happening in the driver and application code, but the steps that can be performed at the same time as the receive data are arriving include as much as the first 3 steps and part of the 4 th step shown in the sequence above. By performing these steps before the entire frame has arrived, the frame throughput can be substantially increased.

A good increase in performance can be expected when the first 3 steps are performed before the end of the network receive operation. A much more significant performance increase could be realized if the PCnet-PCI II controller could place the frame data directly into the applications buffer space; (i.e., eliminate the need for step 4.) In order to make this work, it is necessary that the application buffer pointer be determined before the frame has completely arrived, then the buffer pointer in the next descriptor for the receive frame would need to be modified in order to direct the PCnet-PCI II controller to write directly to the application buffer. More details on this operation will be given later.

An alternative modification to the existing system can gain a smaller, but still significant improvement in performance. This alternative leaves step 4 unchanged in that the CPU is still required to perform the copy operation, but it allows a large portion of the copy operation to be done before the frame has been completely received by the controller; i.e., the CPU can perform the copy operation of the receive data from the PCnet-PCI II controllers buffer space into the application buffer space before the frame data has completely arrived from the network. This allows the copy operation of step 4 to be performed concurrently with the arrival of network data, rather than sequentially, following the end of network receive activity.

Outline of the LAPP Flow

This section gives a suggested outline for a driver that utilizes the LAPP feature of the PCnet-PCI II controller.

***Note**: The labels in the following text are used as references in the timeline diagram that follows.*

SETUP:

The driver should set up descriptors in groups of 3, with the OWN and STP bits of each set of three descriptors to read as follows: 11b, 10b, 00b.

An option bit (LAPPEN) exists in CSR3, bit position 5; the software should set this bit; When set, the LAPPEN bit directs the PCnet-PCI II controller to generate an INTERRUPT when STP has been written to a receive descriptor by the PCnet-PCI II controller.

FLOW:

The PCnet-PCI II controller polls the current receive descriptor at some point in time before a message arrives. The PCnet-PCI II controller determines that this receive buffer is OWNed by the PCnet-PCI II controller and it stores the descriptor information to be used when a message does arrive.

N0: Frame preamble appears on the wire, followed by SFD and destination address.

N1: The 64th byte of frame data arrives from the wire. This causes the PCnet-PCI II controller to begin frame data DMA operations to the first buffer.

C0: When the 64th byte of the message arrives, the PCnet-PCI II controller performs a lookahead operation to the next receive descriptor. This descriptor should be owned by the PCnet-PCI II controller.

C1: The PCnet-PCI II controller intermittently requests the bus to transfer frame data to the first buffer as it arrives on the wire.

S1: The driver remains idle.

C2: When the PCnet-PCI II controller has completely filled the first buffer, it writes status to the first descriptor.

C3: When the first descriptor for the frame has been written, changing ownership from the PCnet-PCI II controller to the CPU, the PCnet-PCI II controller will generate an SRP INTERRUPT. (This interrupt appears as a RINT interrupt in CSR0).

S1: The SRP INTERRUPT causes the CPU to switch tasks to allow the PCnet-PCI II controllers driver to run.

C4: During the CPU interrupt-generated task switching, the PCnet-PCI II controller is performing a lookahead operation to the third descriptor. At this point in time, the third descriptor is owned by the CPU.

***Note:** Even though the third buffer is not owned by the PCnet-PCI II controller, existing AMD Ethernet ontrollers will continue to perform data DMA into the buffer space that the controller already owns (i.e., buffer number 2). The controller does not know if buffer space in buffer number 2 will be sufficient or not, for this frame, but it has no way to tell except by trying to move the entire message into that space. Only when the message does not fit will it signal a buffer error condition – there is no need to panic at the point that it discovers that it does not yet own descriptor number 3.*

S2: The first task of the drivers interrupt service routine is to collect the header information from the PCnet-PCI II controllers first buffer and pass it to the application.

S3: The application will return an application buffer pointer to the driver. The driver will add an offset to the application data buffer pointer, since the PCnet-PCI II controller will be placing the first portion of the message into the first and second buffers. (The modified application data buffer pointer will only be directly used by the PCnet-PCI II controller when it reaches the third buffer.) The driver will place the modified data buffer pointer into the final descriptor of the group (#3) and will grant ownership of this descriptor to the PCnet-PCI II controller.

C5: Interleaved with S2, S3 and S4 driver activity, the PCnet-PCI II controller will write frame data to buffer number 2.

S4: The driver will next proceed to copy the contents of the PCnet-PCI II controllers first buffer to the beginning of the application space. This copy will be to the exact (unmodified) buffer pointer that was passed by the application.

S5: After copying all of the data from the first buffer into the beginning of the application data buffer, the driver will begin to poll the ownership bit of the second descriptor. The driver is waiting for the PCnet-PCI II controller to finish filling the second buffer.

C6: At this point, knowing that it had not previously owned the third descriptor, and knowing that the current message has not ended (there is more data in the FIFO), the PCnet-PCI II controller will make a last ditch lookahead to the final (third) descriptor. This time, the ownership will be TRUE (i.e. the descriptor belongs to the controller), because the driver wrote the application pointer into this descriptor and then changed the ownership to give the descriptor to the PCnet-PCI II controller back at S3. Note that if steps S1, S2 and S3 have not completed at this time, a BUFF error will result.

C7: After filling the second buffer and performing the last chance lookahead to the next descriptor, the PCnet-PCI II controller will write the status and change the ownership bit of descriptor number 2.

S6: After the ownership of descriptor number 2 has been changed by the PCnet-PCI II controller, the next driver poll of the 2nd descriptor will show ownership granted to the CPU. The driver now copies the data from buffer number 2 into the middle section of the application buffer space. This operation is interleaved with the C7 and C8 operations.

C8: The PCnet-PCI II controller will perform data DMA to the last buffer, whose pointer is pointing to application space. Data entering the last buffer will not need the infamous double copy that is required by existing drivers, since it is being placed directly into the application buffer space.

N2: The message on the wire ends.

S7: When the driver completes the copy of buffer number 2 data to the application buffer space, it begins polling descriptor number 3.

C9: When the PCnet-PCI II controller has finished all data DMA operations, it writes status and changes ownership of descriptor number 3.

S8: The driver sees that the ownership of descriptor number 3 has changed, and it calls the application to tell the application that a frame has arrived.

S9: The application processes the received frame and generates the next TX frame, placing it into a TX buffer.

S10: The driver sets up the TX descriptor for the PCnet-PCI II controller.

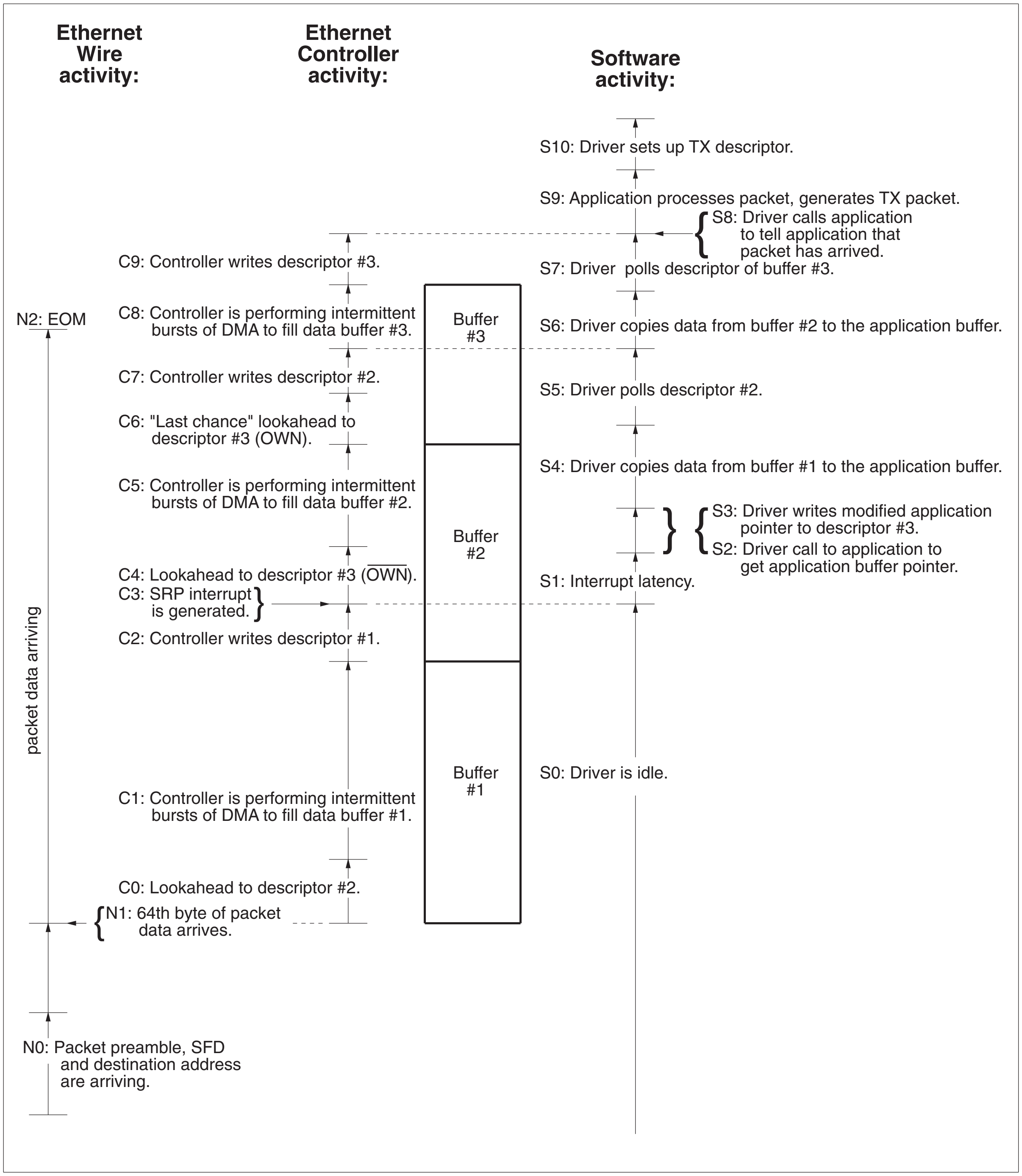
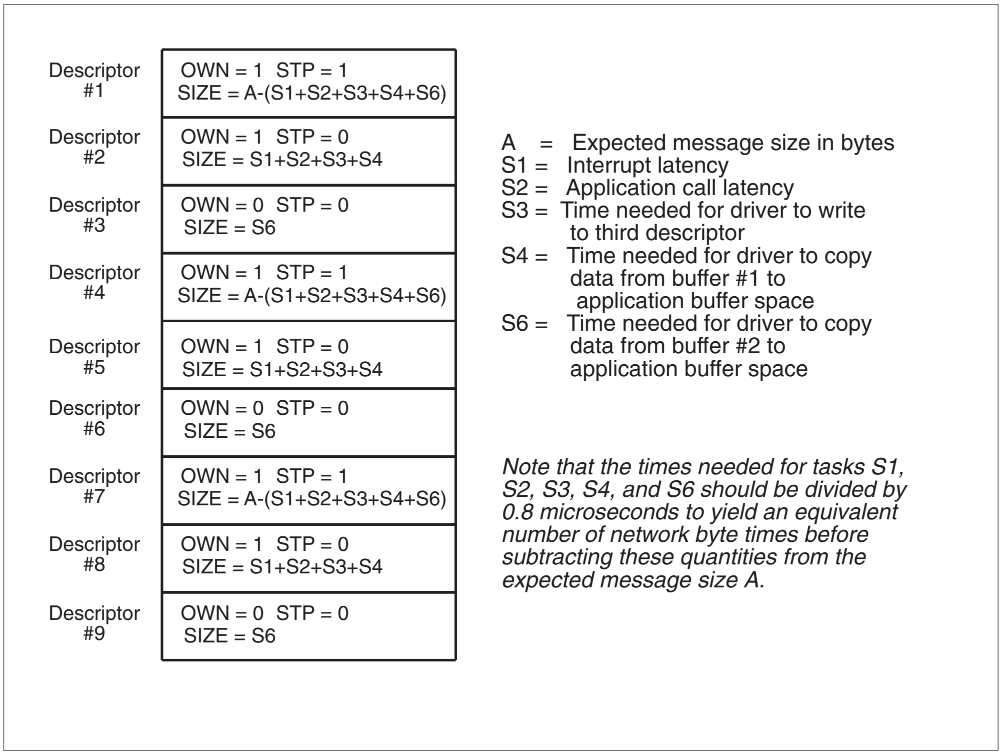

Ethernet Wire activity:
Ethernet Controller activity:
Software activity:
S10: Driver sets up TX descriptor.
S9: Application processes packet, generates TX packet.
S8: Driver calls application to tell application that packet has arrived.
C9: Controller writes descriptor #3.
S7: Driver polls descriptor of buffer #3.
N2: EOM
C8: Controller is performing intermittent bursts of DMA to fill data buffer #3.
Buffer #3
S6: Driver copies data from buffer #2 to the application buffer.
C7: Controller writes descriptor #2.
S5: Driver polls descriptor #2.
C6: "Last chance" lookahead to descriptor #3 (OWN).
S4: Driver copies data from buffer #1 to the application buffer.
C5: Controller is performing intermittent bursts of DMA to fill data buffer #2.
S3: Driver writes modified application pointer to descriptor #3.
Buffer #2
S2: Driver call to application to get application buffer pointer.
C4: Lookahead to descriptor #3 ($\overline{OWN}$).
S1: Interrupt latency.
C3: SRP interrupt is generated.
C2: Controller writes descriptor #1.
packet data arriving
Buffer #1
S0: Driver is idle.
C1: Controller is performing intermittent bursts of DMA to fill data buffer #1.
C0: Lookahead to descriptor #2.
N1: 64th byte of packet data arrives.
N0: Packet preamble, SFD and destination address are arriving.
19436B-75

Figure D1. LAPP Timeline

LAPP Software Requirements

Software needs to set up a receive ring with descriptors formed into groups of 3. The first descriptor of each group should have OWN = 1 and STP = 1, the second descriptor of each group should have OWN = 1 and STP = 0. The third descriptor of each group should have OWN = 0 and STP = 0. The size of the first buffer (as indicated in the first descriptor), should be at least equal to the largest expected header size; however, for maximum efficiency of CPU utilization, the first buffer size should be larger than the header size. It should be equal to the expected number of message bytes, minus the time needed for Interrupt latency and minus the application call latency, minus the time needed for the driver to write to the third descriptor, minus the time needed for the driver to copy data from buffer #1 to the application buffer space, and minus the time needed for the driver to copy data from buffer #2 to the application buffer space. Note that the time needed for the copies performed by the driver depends upon the sizes of the 2nd and 3rd buffers, and that the sizes of the second and third buffers need to be set according to the time needed for the data copy operations! This means that an iterative self-adjusting mechanism needs to be placed into the software to determine the correct buffer sizing for optimal operation. Fixed values for buffer sizes may be used; in such a case, the LAPP method will still provide a significant performance increase, but the performance increase will not be maximized.

The following diagram illustrates this setup for a receive ring size of 9:



Descriptor #1	OWN = 1 STP = 1
	SIZE = A-(S1+S2+S3+S4+S6)

Descriptor #2	OWN = 1 STP = 0
	SIZE = S1+S2+S3+S4

Descriptor #3	OWN = 0 STP = 0
	SIZE = S6

Descriptor #4	OWN = 1 STP = 1
	SIZE = A-(S1+S2+S3+S4+S6)

Descriptor #5	OWN = 1 STP = 0
	SIZE = S1+S2+S3+S4

Descriptor #6	OWN = 0 STP = 0
	SIZE = S6

Descriptor #7	OWN = 1 STP = 1
	SIZE = A-(S1+S2+S3+S4+S6)

Descriptor #8	OWN = 1 STP = 0
	SIZE = S1+S2+S3+S4

Descriptor #9	OWN = 0 STP = 0
	SIZE = S6

A = Expected message size in bytes
S1 = Interrupt latency
S2 = Application call latency
S3 = Time needed for driver to write to third descriptor
S4 = Time needed for driver to copy data from buffer #1 to application buffer space
S6 = Time needed for driver to copy data from buffer #2 to application buffer space

Note that the times needed for tasks S1, S2, S3, S4, and S6 should be divided by 0.8 microseconds to yield an equivalent number of network byte times before subtracting these quantities from the expected message size A.

19436B-76

Figure D2. LAPP 3 Buffer Grouping

LAPP Rules for Parsing of Descriptors

When using the LAPP method, software must use a modified form of descriptor *parsing* as follows:

- Software will examine OWN and STP to determine where a RCV frame begins. RCV frames will only begin in buffers that have OWN = 0 and STP = 1.
- Software shall assume that a frame continues until it finds either ENP = 1 or ERR= 1.
- Software must discard all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for the beginning of a new frame; ENP and ERR should be ignored by software during this search.
- Software cannot change an STP value in the receive descriptor ring after the initial setup of the ring is complete, even if software has ownership of the STP descriptor unless the previous STP descriptor in the ring is also OWNED by the software.

When LAPPEN = 1, then hardware will use a modified form of descriptor parsing as follows:

- The controller will examine OWN and STP to determine where to begin placing a RCV frame. A new RCV frame will only begin in a buffer that has OWN= 1 and STP = 1.
- The controller will always obey the OWN bit for determining whether or not it may use the next buffer for a chain.
- The controller will always mark the end of a frame with either ENP = 1 or ERR= 1.

  The controller will discard all descriptors with OWN = 1 and STP = 0 and move to the next descriptor when searching for a place to begin a new frame. It discards these descriptors by simply changing the ownership bit from OWN=1 to OWN = 0. Such a descriptor is unused for receive purposes by the controller, and the driver must recognize this. (The driver will recognize this if it follows the software rules).

The controller will ignore all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for a place to begin a new frame. In other words, the controller is allowed to skip entries in the ring that it does not own, but only when it is looking for a place to begin a new frame.

Some Examples of LAPP Descriptor Interaction

Choose an expected frame size of 1060 bytes. Choose buffer sizes of 800, 200 and 200 bytes.

- Assume that a 1060 byte frame arrives correctly, and that the timing of the early interrupt and the software is smooth. The descriptors will have changed from:

Descriptor Number	Before the Frame Arrives			After the Frame Arrived			Comments
	OWN	STP	ENP†	OWN	STP	ENP†	(After frame arrival)
1	1	1	X	0	1	0	Bytes 1–800
2	1	0	X	0	0	0	Bytes 801–1000
3	0	0	X	0	0	1	Bytes 1001–1060
4	1	1	X	1	1	X	Controller’s current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

† ENP or ERR

- Assume that instead of the expected 1060 byte frame, a 900 byte frame arrives, either because there was an error in the network, or because this is the last frame in a file transmission sequence.

Descriptor Number	Before the Frame Arrives			After the Frame Arrived			Comments
	OWN	STP	ENP†	OWN	STP	ENP†	(After frame arrival)
1	1	1	X	0	1	0	Bytes 1–800
2	1	0	X	0	0	1	Bytes 801–900
3	0	0	X	0	0	?*	Discarded buffer
4	1	1	X	1	1	X	Controller’s current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

† ENP or ERR

Note that the PCnet-PCI II controller might write a ZERO to ENP location in the 3rd descriptor. Here are the two possibilities:

1. If the controller finishes the data transfers into buffer number 2 after the driver writes the applications modified buffer pointer into the third descriptor, then the controller will write a ZERO to ENP for this buffer and will write a ZERO to OWN and STP.
2. If the controller finishes the data transfers into buffer number 2 before the driver writes the applications modified buffer pointer into the third descriptor, then the controller will complete the frame in buffer number two and then skip the then unowned third buffer. In this case, the PCnet-PCI II controller will not have had the opportunity to RESET the ENP bit in this descriptor, and it is possible that the software left this bit as ENP=1 from the last time through the ring. Therefore, the software must treat the location as a don't care; The rule is, after finding ENP=1 (or ERR=1) in descriptor number 2, the software must ignore ENP bits until it finds the next STP=1.

Assume that instead of the expected 1060 byte frame, a 100 byte frame arrives, because there was an error in the network, or because this is the last frame in a file transmission sequence, or perhaps because it is an acknowledge frame.

* Same as note in case 2 above, except that in this case, it is very unlikely that the driver can respond to the interrupt and get the pointer from the application before the PCnet-PCI II controller has completed its poll of the next descriptors. This means that for almost all occurrences of this case, the PCnet-PCI II controller will not find the OWN bit set for this descriptor and therefore, the ENP bit will almost always contain the old value, since the PCnet-PCI II controller will not have had an opportunity to modify it.

- ** Note that even though the PCnet-PCI II controller will write a ZERO to this ENP location, the software should treat the location as a don't care, since after finding the ENP=1 in descriptor number 2, the software should ignore ENP bits until it finds the next STP=1.

Descriptor Number	Before the Frame Arrives			After the Frame Arrived			Comments
	OWN	STP	ENP†	OWN	STP	ENP†	(After frame arrival)
1	1	1	X	0	1	1	Bytes 1–100
2	1	0	X	0	0	0**	Discarded buffer
3	0	0	X	0	0	?*	Discarded buffer
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

† *ENP or ERR*

Buffer Size Tuning

For maximum performance, buffer sizes should be adjusted depending upon the expected frame size and the values of the interrupt latency and application call latency. The best driver code will minimize the CPU utilization while also minimizing the latency from frame end on the network to frame sent to application from driver (frame latency). These objectives are aimed at increasing throughput on the network while decreasing CPU utilization.

Note that the buffer sizes in the ring may be altered at any time that the CPU has ownership of the corresponding descriptor. The best choice for buffer sizes will maximize the time that the driver is swapped out, while minimizing the time from the last byte written by the PCnet-PCI II controller to the time that the data is passed from the driver to the application. In the diagram, this corresponds to maximizing S0, while minimizing the time between C9 and S8. (The timeline happens to show a minimal time from C9 to S8.)

Note that by increasing the size of buffer number 1, we increase the value of S0. However, when we increase the size of buffer number 1, we also increase the value of S4. If the size of buffer number 1 is too large, then the driver will not have enough time to perform tasks S2, S3, S4, S5 and S6. The result is that there will be delay from the execution of task C9 until the execution of task S8. A p*erfectly timed system will have the values for S5 and S7 at a minimum*.

An average increase in performance can be achieved if the general guidelines of buffer sizes in figure 2 is followed. However, as was noted earlier, the correct sizing for buffers will depend upon the expected message size. There are two problems with relating expected message size with the correct buffer sizing:

1. Message sizes cannot always be accurately predicted, since a single application may expect different message sizes at different times, therefore, the buffer sizes chosen will not always maximize throughput.
2. Within a single application, message sizes might be somewhat predictable, but when the same driver is to be shared with multiple applications, there may not be a common predictable message size.

Additional problems occur when trying to define the correct sizing because the correct size also depends upon the interrupt latency, which may vary from system to system, depending upon both the hardware and the software installed in each system.

In order to deal with the unpredictable nature of the message size, the driver can implement a self tuning mechanism that examines the amount of time spent in tasks S5 and S7 as such: while the driver is polling for each descriptor, it could count the number of poll operations performed and then adjust the number 1 buffer size to a larger value, by adding “t'' bytes to the buffer count, if the number of poll operations was greater than “x''. If fewer than “x'' poll operations were needed for each of S5 and S7, then the software should adjust the buffer size to a smaller value by, subtracting “y'' bytes from the buffer count. Experiments with such a tuning mechanism must be performed to determine the best values for “X'' and “y''.

Note whenever the size of buffer number 1 is adjusted, buffer sizes for buffer number 2 and buffer 3 should also be adjusted.

In some systems, the typical mix of receive frames on a network for a client application consists mostly of large data frames, with very few small frames. In this case, for maximum efficiency of buffer sizing, when a frame arrives under a certain size limit, the driver should not adjust the buffer sizes in response to the short frame.

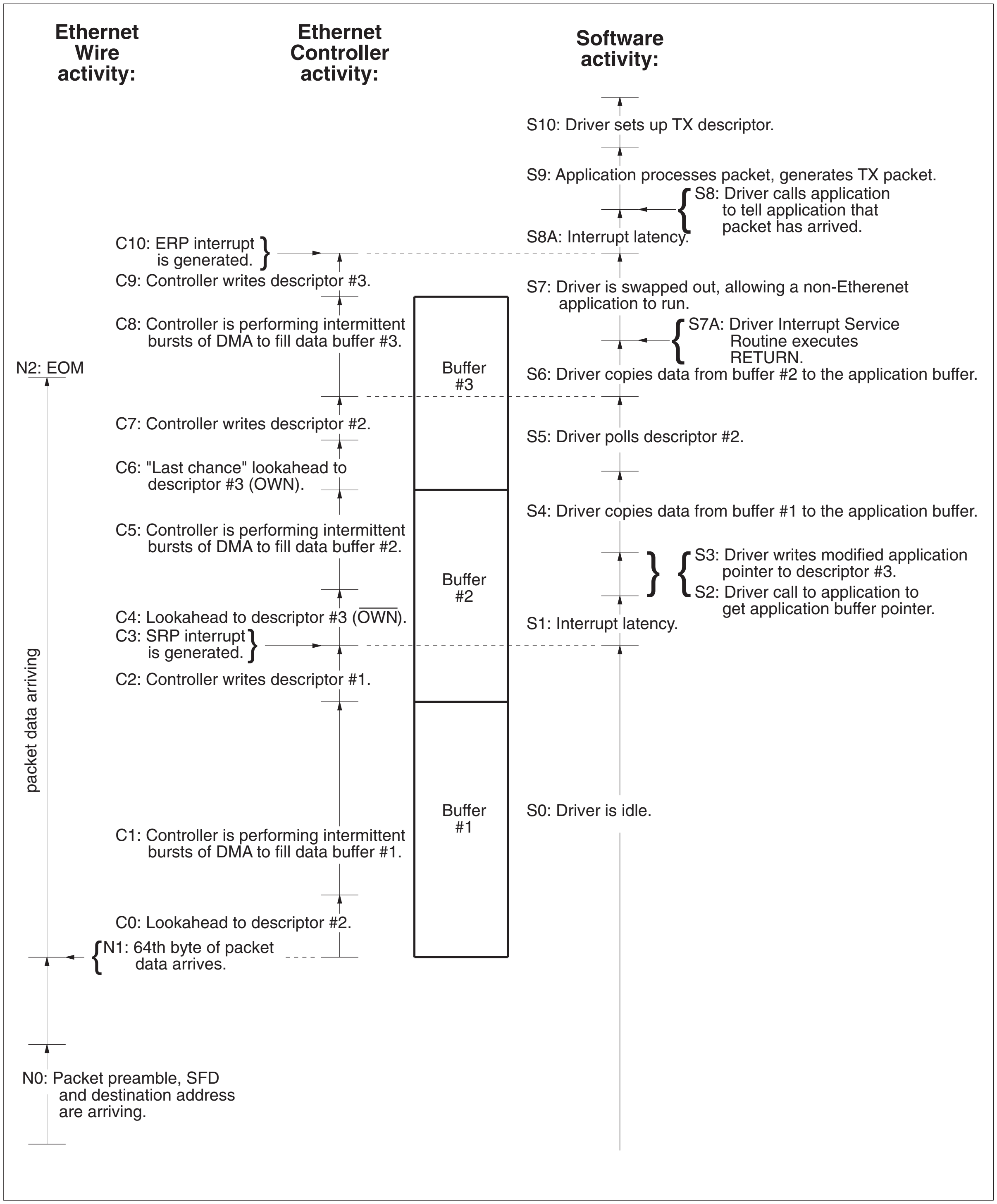
An Alternative LAPP Flow—the TWO Interrupt Method

An alternative to the above suggested flow is to use two interrupts, one at the start of the receive frame and the other at the end of the receive frame, instead of just looking for the SRP interrupt as was described above. This alternative attempts to reduce the amount of time that the software wastes while polling for descriptor own bits. This time would then be available for other CPU tasks. It also minimizes the amount of time the CPU needs for data copying. This savings can be applied to other CPU tasks.

The time from the end of frame arrival on the wire to delivery of the frame to the application is labeled as frame latency. For the one-interrupt method, frame latency is minimized, while CPU utilization increases. For the two-interrupt method, frame latency becomes greater, while CPU utilization decreases.

Note that some of the CPU time that can be applied to non-Ethernet tasks is used for task switching in the CPU. One task switch is required to swap a non-Ethernet task into the CPU (after S7A) and a second task switch is needed to swap the Ethernet driver back in again (at S8A). If the time needed to perform these task switches exceeds the time saved by not polling descriptors, then there is a net loss in performance with this method. Therefore, the LAPP method implemented should be carefully chosen.

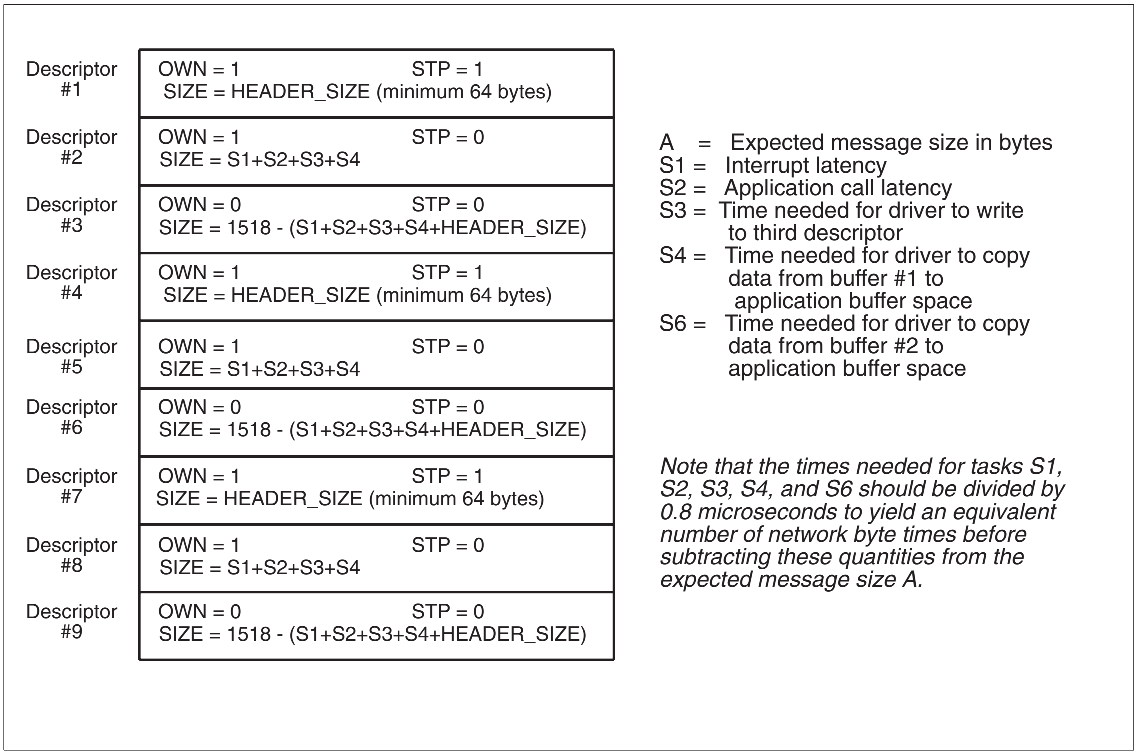
Figure D3 shows the event flow for the two-interrupt method:


Ethernet Wire activity:
Ethernet Controller activity:
Software activity:
S10: Driver sets up TX descriptor.
S9: Application processes packet, generates TX packet.
S8: Driver calls application to tell application that packet has arrived.
S8A: Interrupt latency.
C10: ERP interrupt is generated.
C9: Controller writes descriptor #3.
S7: Driver is swapped out, allowing a non-Etherenet application to run.
C8: Controller is performing intermittent bursts of DMA to fill data buffer #3.
S7A: Driver Interrupt Service Routine executes RETURN.
N2: EOM
Buffer #3
S6: Driver copies data from buffer #2 to the application buffer.
C7: Controller writes descriptor #2.
S5: Driver polls descriptor #2.
C6: "Last chance" lookahead to descriptor #3 (OWN).
S4: Driver copies data from buffer #1 to the application buffer.
C5: Controller is performing intermittent bursts of DMA to fill data buffer #2.
S3: Driver writes modified application pointer to descriptor #3.
Buffer #2
S2: Driver call to application to get application buffer pointer.
C4: Lookahead to descriptor #3 ($\overline{\text{OWN}}$).
S1: Interrupt latency.
C3: SRP interrupt is generated.
C2: Controller writes descriptor #1.
packet data arriving
Buffer #1
S0: Driver is idle.
C1: Controller is performing intermittent bursts of DMA to fill data buffer #1.
C0: Lookahead to descriptor #2.
N1: 64th byte of packet data arrives.
N0: Packet preamble, SFD and destination address are arriving.

19436B-77

Figure D3. LAPP Timeline for TWO-INTERRUPT Method

Figure D4 shows the buffer sizing for the two-interrupt method. Note that the second buffer size will be about the same for each method.



Descriptor #1
OWN = 1 STP = 1
SIZE = HEADER_SIZE (minimum 64 bytes)

Descriptor #2
OWN = 1 STP = 0
SIZE = S1+S2+S3+S4

Descriptor #3
OWN = 0 STP = 0
SIZE = 1518 - (S1+S2+S3+S4+HEADER_SIZE)

Descriptor #4
OWN = 1 STP = 1
SIZE = HEADER_SIZE (minimum 64 bytes)

Descriptor #5
OWN = 1 STP = 0
SIZE = S1+S2+S3+S4

Descriptor #6
OWN = 0 STP = 0
SIZE = 1518 - (S1+S2+S3+S4+HEADER_SIZE)

Descriptor #7
OWN = 1 STP = 1
SIZE = HEADER_SIZE (minimum 64 bytes)

Descriptor #8
OWN = 1 STP = 0
SIZE = S1+S2+S3+S4

Descriptor #9
OWN = 0 STP = 0
SIZE = 1518 - (S1+S2+S3+S4+HEADER_SIZE)

A = Expected message size in bytes
S1 = Interrupt latency
S2 = Application call latency
S3 = Time needed for driver to write to third descriptor
S4 = Time needed for driver to copy data from buffer #1 to application buffer space
S6 = Time needed for driver to copy data from buffer #2 to application buffer space

Note that the times needed for tasks S1, S2, S3, S4, and S6 should be divided by 0.8 microseconds to yield an equivalent number of network byte times before subtracting these quantities from the expected message size A.

19436B-78

Figure D4. LAPP 3 Buffer Grouping for TWO-INTERRUPT Method

There is another alternative which is a marriage of the two previous methods. This third possibility would use the buffer sizes set by the two-interrupt method, but would use the polling method of determining frame end. This will give good frame latency but at the price of very high CPU utilization. And still, there are even-more compromise positions that use various fixed buffer sizes and effectively, the flow of the one-interrupt method. All of these compromises will reduce the complexity of the one-interrupt method by removing the heuristic buffer sizing code, but they all become less efficient than heuristic code would allow.

APPENDIX E

PCnet-PCI II and PCnet-PCI Differences

OVERVIEW

This appendix summarizes the enhancements of the Am79C970A PCnet-PCI II controller over the Am79C970 PCnet-PCI controller. The feature summary is followed by a detailed list of all register bit changes. The document also compares the pinout of the PCnet-PCI II controller with the pinout of the PCnet-PCI and the Am79C974 PCnet-SCSI (also known as Golden Gate) to show that the Flex-I/O footprint is continued to be supported.

NEW FEATURES

- Three Volt support for PCI bus interface
- Full Duplex Ethernet
- 272-byte Transmit FIFO, 256-byte Receive FIFO
- Enhanced PCI bus transfer cycles:
  - — No more address stepping
  - — Initialization Block read in non-burst (default) or burst mode
  - — Added new software style and reordered the descriptor entries to allow burst transfers for both, descriptor read and write accesses
  - — FIFO DMA bursts length programmable from 1 to indefinite
  - — Type of memory command for burst read transfers programmable to be either Memory Read Line or Memory Read Multiple (controlled by MEMCMD, BCR18, bit 9)
  - — Support for fast back-to-back slave transactions even when the first transaction is addressing a different target MEMCMD, BCR18, bit 9)
  - — Enhanced disconnect of I/O burst access
- Allows I/O resources to be memory mapped
- Eight-bit programmable PCI Latency Timer. MIN_GNT and MAX_LAT programmable via EEPROM
- System interrupt for data parity error, master abort or target abort in master cycles
- Network activity is terminated in an orderly sequence after a master or target abort
- Advanced parity error handling. Mode has enable bit and status bit in RMD1 and TMD1. All network activity is terminated in an orderly sequence. Will only work with 32-bit software structures.
- All registers in the PCI configuration space are cleared by H_RESET
- Expansion ROM interface supporting devices of up to 64 K x 8. One external address latch is required.
- Reading from the S_RESET port returns $\overline{\text{TRDY}}$ right away
- $\overline{\text{REQ}}$ deassertion programmable to adapt to the requirements of some embedded systems
- $\overline{\text{INTA}}$ pin programmable for pulse mode to adapt to the requirements of some embedded systems
- Some previously reserved locations in the EEPROM map are now used for new features
- Suspend mode for graceful stop and access to the CSR without reinitialization
- User Interrupt
- Reduced number of transmit interrupts:
  - — Transmit OK disable (CSR5, bit 15). When bit is set to ONE, a transmit interrupt is only generated on frames that suffer an error.
  - — Last Transmit Interrupt. TMD1, bit 28 is read by the PCnet-PCI II controller to determine if an interrupt should be generated at the end of the frame. Only interrupts for successful transmission can be suppressed. Enabled by LTINTEN (CSR5, bit 14).
- Disable Transmit Stop on Underflow (CSR3, bit 6) bit. PCnet-PCI controller recovers automatically from transmit underflow.
- Interrupt indication when coming out of sleep mode
- Interrupt indication for Excessive Deferral
- Address match information in Receive Descriptor
- Asserting $\overline{\text{SLEEP}}$ shuts down the entire device
- S_RESET (reading the RESET register) does not affect the TMAU, except for the T-MAU in snooze mode
- LED registers programmable via EEPROM.
- Magic Packet Mode
- EADI interface. Multiplexed with the same LED pins as for the Am79C965 PCnet-32.

- JTAG interface
- Fourth LED supported
- Pin to disable external transceiver or DC-to-DC converter. Polarity of assertion state programmable.

LIST OF REGISTER BIT CHANGES

PCI Configuration Space

Command Register

- ADSTEP (bit 7) now hardwired to ZERO. Was hardwired to ONE.
- MEMEN (bit 1) now read/write accessible. Was hardwired to ZERO.

Status Register

- PERR (bit 15) now cleared by H_RESET. Was not effected by H_RESET.
- SERR (bit 14) now cleared by H_RESET. Was not effected by H_RESET.
- RMABORT (bit 13) now cleared by H_RESET. Was not effected by H_RESET.
- RTABORT (bit 12) now cleared by H_RESET. Was not effected by H_RESET.
- STABORT (bit 11) now cleared by H_RESET. Was not effected by H_RESET.
- DATAPERR (bit 8) now cleared by H_RESET. Was not effected by H_RESET.
- FBTBC (bit 7) now hardwired to ONE. Was hardwired to ZERO.

Revision ID Register

- This 8-bit register is now hardwired to 1xh. It was hardwired to 0xh.

Latency Timer Register

- This 8-bit register is now read/write accessible. Was hardwired to ZERO.

I/O Base Address Register

- IOBASE (bits 31--5) now cleared by H_RESET. Was not effected by H_RESET.

Memory Mapped I/O Base Address Register

- New 32-bit register. Was reserved, read as ZERO, writes have no effect.

Expansion ROM Base Address Register

- New 32-bit register. Was reserved, read as ZERO, writes have no effect.

Interrupt Line Register

- This 8-bit register is now cleared by H_RESET. Was not effected by H_RESET.

MIN_GNT Register

- New 8-bit register. Was reserved, read as ZERO, writes have no effect.

MAX_LAT Register

- New 8-bit register. Was reserved, read as ZERO, writes have no effect.

Control And Status Registers

CSR0: PCnet-PCI II controller Control and Status Register

- In addition to the existing interrupt flags, INTR (bit 7), the interrupt summary bit, is also affected by the new interrupt flags Excessive Deferral Interrupt (EXDINT), Magic Packet Interrupt (MPINT) Sleep Interrupt (SLPINT), System Interrupt (SINT) and User Interrupt (UINT).

CSR3: Interrupt Masks and Deferral Control

- New bit: DXSUFLO (bit 6), Disable Transmit Stop on Underflow error. Was reserved location, read and written as ZERO.

CSR4: Test and Features Control

- New bit: UINTCMD (bit 7), User Interrupt Command. Was reserved location, read and written as ZERO.
- New bit: UINT (bit 6), User Interrupt. Was reserved location, read as ZERO, written as ONE or ZERO.

CSR5:

- New bit: TOKINTD (bit 15), Transmit OK Interrupt Disable. Was reserved location, read and written as ZERO.
- New bit: LTINTEN (bit 14), Last Transmit Interrupt Enable. Was reserved location, read and written as ZERO.
- New bit: SINT (bit 11), System Interrupt. Was reserved location, read and written as ZERO.
- New bit: SINTE (bit 10), System Interrupt Enable. Was reserved location, read and written as ZERO.
- New bit: SLPINT (bit 9), Sleep Interrupt. Was reserved location, read and written as ZERO.
- New bit: SLPINTE (bit 8), Sleep Interrupt Enable. Was reserved location, read and written as ZERO.
- New bit: EXDINT (bit 7), Excessive Deferral Interrupt. Was reserved location, read and written as ZERO.
- New bit: EXDINTE (bit 6), Excessive Deferral Interrupt Enable. Was reserved location, read and written as ZERO.
- New bit: MPPLBA (bit 5), Magic Packet Physical Logical Broadcast Accept. Was reserved location, read and written as ZERO.
- New bit: MPINT (bit 4), Magic Packet Interrupt. Was reserved location, read and written as ZERO.
- New bit: MPINTE (bit 3), Magic Packet Interrupt Enable. Was reserved location, read and written as ZERO.

- New bit: MPEN (bit 2), Magic Packet Enable. Was reserved location, read and written as ZERO.
- New bit: MPMODE (bit 1), Magic Packet Mode. Was reserved location, read and written as ZERO.
- New bit: SPND (bit 0), Suspend. Was reserved location, read and written as ZERO.

CSR58: Software Style

- New bit: APERREN (bit 10), Advanced Parity Error Handling Enable. Was reserved location, read and written as ZERO.
- SWSTYLE (bits 7–0), Software Style. New option, value of THREE selects new PCnet-PCI controller style that reorders 32-bit descriptor entries to allow burst accesses.

CSR80: DMA Transfer Counter and FIFO Threshold Control

- RCVFW (bits 13–12), Receive FIFO Watermark. Decoding adjusted for the larger FIFO size.
- XMTSP (bits 11–10), Transmit Start Point. Decoding adjusted for the larger FIFO size.
- XMTFW (bits 9–8), Transmit FIFO Watermark. Decoding adjusted for the larger FIFO size.
- DMATC (bits 7–0), DMA Transfer Count. Function of the counter is optimized for the PCI bus environment.

CSR82: Bus Activity Timer

- DMABAT (bits 15–0), DMA Bus Activity Timer. Function of the counter is optimized for the PCI bus environment.

CSR88: Chip ID Lower

- New value: 1003h. Was 0003h.

CSR89: Chip ID Upper

- New value: 0262h. Was 0243h.

CSR100: Bus Timeout

- Default value now 0600h (153.6 µs) to adjust to the larger FIFO size. Default value was 0200h (51.2 µs).

CSR112: Missed Frame Count

- Counter is stopped while the device is in suspend mode

Bus Configuration Registers

BCR2: Miscellaneous Configuration

- New bit: INTLEVEL (bit 7), Interrupt Level. Was reserved location, read and written as ZERO.
- New bit: DXCVRCTL (bit 5), DXCVR Control. Was reserved location, read and written as ZERO.
- New bit: DXCVRPOL (bit 4), DXCVR Polarity. Was reserved location, read and written as ZERO.
- New bit: EADISEL (bit 3), EADI Select. Was reserved location, read and written as ZERO.

BCR4: Link Status LED

- Register is now programmable through the EEPROM
- New bit: MPSE (bit 9), Magic Packet Status Enable. Was reserved location, read and written as ZERO.
- New bit: FDLSE (bit 8), Full Duplex Link Status Enable. Was reserved location, read and written as ZERO.
- COLE (bit 0), Collision Status Enable. Corrected behavior of function. LED will not light up due to SQE test collision signal.

BCR5: LED1 Status

- Register is now programmable through the EEPROM
- New bit: MPSE (bit 9), Magic Packet Status Enable. Was reserved location, read and written as ZERO.
- New bit: FDLSE (bit 8), Full Duplex Link Status Enable. Was reserved location, read and written as ZERO.
- COLE (bit 0), Collision Status Enable. Corrected behavior of function. LED will not light up due to SQE test collision signal.

BCR6: LED2 Status

- New register. Was reserved location, the settings of the register have no effect on the operation of the device.

BCR7: LED3 Status

- Register is now programmable through the EEPROM
- New bit: MPSE (bit 9), Magic Packet Status Enable. Was reserved location, read and written as ZERO.
- New bit: FDLSE (bit 8), Full Duplex Link Status Enable. Was reserved location, read and written as ZERO.
- COLE (bit 0), Collision Status Enable. Corrected behavior of function. LED will not light up due to SQE test collision signal.

BCR9: Full Duplex Control

- New register. Was reserved location, read and written as ZERO.

BCR16: I/O Base Address Lower

- This register is no longer programmable through the EEPROM. The register is reserved and has no effect on the operation of the device. It is only used in the PCnet-32.

BCR17: I/O Base Address Upper

- This register is no longer programmable through the EEPROM. The register is reserved and has no effect on the operation of the device. It is only used in the PCnet-32.

BCR18: Burst Size and Bus Control

- New bits: ROMTMG (bits 15–12), Expansion ROM Timing. Was reserved location, read and written as ZERO.
- New bit: MEMCMD (bit 9), Memory Command. Was reserved location, read and written as ZERO.
- New bit: EXTREQ (bit 8), Extended Request. Was reserved location, read and written as ONE.
- BREADE (bit 6), Burst Read Enable. Extended functionality of bit. Besides enabling burst read accesses to the transmit buffer, BREADE will now also enable burst read accesses to the initialization block and, if SWSTYLE = 3, to the descriptor ring entries.
- BWRITE (bit 5), Burst Write Enable. Extended functionality of bit. Besides enabling burst write accesses to the receive buffer, BWRITE will now also enable burst write accesses to the descriptor ring entries, if SWSTYLE = 3.
- LINBC (bits 2–0), Linear Burst Count. These bits are now reserved and have no effect on the operation of the device.

BCR20: Software Style

- New bit: APERREN (bit 10), Advanced Parity Error Handling Enable. Was reserved location, read and written as ZERO.
- SWSTYLE (bits 7–0), Software Style. New option, value of THREE selects new PCnet-PCI controller style that reorders 32-bit descriptor entries to allow burst accesses.

BCR21: Interrupt Control

- This register is no longer programmable through the EEPROM. The register is reserved and has no effect on the operation of the device. It is only used in the PCnet-32.

BCR22: PCI Latency

- New register. Was reserved location, read and written as ZERO.

Receive Descriptor

RMD1

- New bit: BPE (bit 23), Bus Parity Error. This bit is active only if 32-bit software structures are used for the descriptor ring entries (SWSTYLE = ONE, TWO or THREE) and if APERREN (BCR20, bit 10) is set to ONE. Was reserved location, read and written as ZERO.
- New bit: PAM (bit 22), Physical Address Match. This bit is active only if 32-bit software structures are used for the descriptor ring entries (SWSTYLE = ONE, TWO or THREE). Was reserved location, read and written as ZERO.
- New bit: LAFM (bit 21), Logical Address Filter Match. This bit is active only if 32-bit software structures are used for the descriptor ring entries (SWSTYLE = ONE, TWO or THREE). Was reserved location, read and written as ZERO.
- New bit: BAM (bit 20), Broadcast Address Match. This bit is active only if 32-bit software structures are used for the descriptor ring entries (SWSTYLE = ONE, TWO or THREE). Was reserved location, read and written as ZERO.

Transmit Descriptor

TMD1

- New bit: LTINT (bit 28), Last Transmit Interrupt. This bit is only active, if LTINTEN (CSR5, bit 14) is set to ONE. This bit location is shared with the MORE status bit. The host will write the bit as LTINT and read it as MORE. The P2 will read the bit as LTINT and write it as MORE.
- New bit: BPE (bit 23), Bus Parity Error. This bit is only active, if 32-bit software structures are used for the descriptor ring entries (SWSTYLE = ONE, TWO or THREE) and if APERREN (BCR20, bit 10) is set to ONE. Was reserved location, read and written as ZERO.

LIST OF PIN CHANGES

Pin No.	PCnet-SCSI	PCnet-PCI	PCnet-PCI II	Comment
9	IDSELA	NC	TDI	CIN must be 8 pF maximum. (PCI spec. on IDSEL input).
58	PWDN	NC	$\overline{\text{EAR}}$	Inputs only, $\overline{\text{EAR}}$ is ignored until EADI interface is enabled.
60	SCSICLK	NC	$\overline{\text{EROE}}$	External SCSI components must be depopulated.
62	$\overline{\text{BUSY}}$/NOUT	NOUT	DXCVR/NOUT	External SCSI components must be depopulated.
64	$\overline{\text{SCSI}}$ $\overline{\text{BSY}}$	NC	ERACLK	External SCSI components must be depopulated.
65	$\overline{\text{SCSI}}$ $\overline{\text{ATN}}$	NC	ERA7	External SCSI components must be depopulated.
66	$\overline{\text{SCSI}}$ $\overline{\text{RST}}$	NC	ERA6	External SCSI components must be depopulated.
68	$\overline{\text{SCSI}}$ $\overline{\text{DS}}$	NC	ERA5	External SCSI components must be depopulated.
69	$\overline{\text{SCSI}}$ $\overline{\text{SD1}}$	NC	ERA4	External SCSI components must be depopulated.
70	$\overline{\text{SCSI}}$ $\overline{\text{SD2}}$	NC	ERA3	External SCSI components must be depopulated.
71	$\overline{\text{SCSI}}$ $\overline{\text{SD3}}$	NC	ERA2	External SCSI components must be depopulated.
73	$\overline{\text{SCSI}}$ $\overline{\text{SD4}}$	NC	ERA1	External SCSI components must be depopulated.
74	$\overline{\text{SCSI}}$ $\overline{\text{SD5}}$	NC	ERA0	External SCSI components must be depopulated.
75	$\overline{\text{SCSI}}$ $\overline{\text{SD6}}$	NC	ERD7/TXDAT	External SCSI components must be depopulated.
77	$\overline{\text{SCSI}}$ $\overline{\text{SD7}}$	NC	ERD6/TXEN	External SCSI components must be depopulated.
78	$\overline{\text{SCSI}}$ $\overline{\text{SDP}}$	NC	ERD5	External SCSI components must be depopulated.
80	$\overline{\text{SCSI}}$ $\overline{\text{SEL}}$	NC	ERD4/TXCLK	External SCSI components must be depopulated.
81	$\overline{\text{SCSI}}$ $\overline{\text{REQ}}$	NC	ERD3/CLSN	External SCSI components must be depopulated.
83	$\overline{\text{SCSI}}$ $\overline{\text{ACK}}$	NC	ERD2/RXEN	External SCSI components must be depopulated.
85	$\overline{\text{SCSI}}$ $\overline{\text{MSG}}$	NC	ERD1/RXCLK	External SCSI components must be depopulated.
86	$\overline{\text{SCSI}}$ $\overline{\text{C/D}}$	NC	ERD0/RXDAT	External SCSI components must be depopulated.
87	$\overline{\text{SCSI}}$ $\overline{\text{I/O}}$	NC	LED2/SRDCLK	External SCSI components must be depopulated.
110	EEDO/$\overline{\text{LED3}}$	EEDO/LED3	EEDO/$\overline{\text{LED3}}$/SRD	EADI interface is only active when enabled by setting a bit in a BCR.
112	EESK/$\overline{\text{LED1}}$	EESK/LED1	EESK/$\overline{\text{LED1}}$/SFBD	EADI interface is only active when enabled by setting a bit in a BCR.
116	RESERVED	RESERVED	RESERVED	
118	$\overline{\text{INTB}}$	NC	TCK	TCK is the JTAG clock input. The JTAG interface is inactive, until TCK is running. TCK has an internal pull-up.
124	$\overline{\text{GNTA}}$	NC	TMS	TMS is the JTAG test mode select input. TMS is only active if TCK is running.
127	$\overline{\text{REQA}}$	NC	TDO	JTAG TDO output is tri-state after power-on reset.

Am79C970A PCnet-PCI II

Silicon Errata Report

Am79C970A PC-net-PCI II Rev B2 Silicon Errata

The items below are the known errata for Rev B2 silicon. Rev B2 silicon is the production silicon.

This device has a total of nine known errata. One is AC timing related and the other eight are functional errata. The AC issue is related to an extended maximum limit on tval for specific pins. The eight functional errata are related to the DWIO mode operation, the JTAG mode operation, the GPSI port, bus parking operation, LED status indication, LAPP mode operation, excess deferral interrupt operation, and false BABL indication, respectively.

The **"Description"** section of this document gives an external description of the problem. The **"Implication"** section explains how the device behaves and its impact on the system. The **"Workaround"** section describes a work around for the problem. The **"Status"** section indicates when and how the problem will be fixed.

Current package marking for this revision: Line 1: <Logo>
Line 2: PCnet(tm)-PCI II
Line 3: Am79C970AKC (Assuming package is PQFP)
Line 4: <Date Code> A/B (Where A = Fab 15 & B = Fab 14)
Line 5: (c) 1994 AMD

Value of chip id registers, CSR89+CSR88 [31:0] for this revision = 62621003h. (When read in 16 bit mode.)

PCI Configuration Register (offset 0x08h) = xxxxxx16h.

1) **Description:** The PCnet-PCI II device ***exceeds one PCI timing parameter by one ns.***

Implication: The device exceeds the PCI timing parameter for tval on certain signals. Under worst-case conditions the ***tval for AD, C/BE#, and PAR*** can exceed 11 ns. All other AC timing parameters meet the PCI specification values.

All PCnet-PCI II production devices are screened to this new tval maximum limit, except those material with a mark date code of 9525APA. Devices with this date code were screened to the previous tval max, limit of 14.0 ns for AD, C/BE#, and PAR signals.

Parameter	PCI Spec Min	PCI Spec Max	B2 Min	B2 Max
Valid Delay Timing				
t_{VAL} (AD, C/BE#, PAR)	***2.0***	***11.0***	***same as spec***	***12.0***
t_{VAL} *(all other signals except REQ#)*	2.0	11.0	same as spec	same as spec
t_{VAL} *(REQ#)*	2.0	12.0	same as spec	same as spec
Setup Timing				
t_{SU} *(all signals except GNT#)*	7.0	—	same as spec	—
t_{SU} *(GNT#)*	10.0	—	same as spec	—
Hold Timing				
t_H *(all signals)*	0	—	same as spec	—

*Note: The **B2 Max** column shows the absolute worst-case value for this parameter. This value is guaranteed over the data sheet-specified Vcc and temperature ranges. The worst-case value is determined based on preliminary characterization data. Upon completion of device characterization, this value may be improved.*

Workaround: None.

Status: No current plan to fix this item.

2) **Description:** The ***PERR# output glitches*** when configuring the PCnet-PCI II device to operate ***in DWIO (Double-Word I/O) mode*** and the PERREN bit in the PCI Command Register is set. The glitch occurs near the end of the slave write cycle to offset 10H.

Implication: Due to the glitch on PERR# during the DWIO initialization cycle, tval of PERR# is more than 11 ns during the cycle immediately following the initialization cycle. tval of PERR# could be 12 ns during that cycle. tval of PERR# is guaranteed to be 11 ns on all other cycles.

There is no jeopardy to customers using AMD drivers. AMD drivers do not place the device into DWIO mode. Setting the device into 32-bit I/O mode is usually the first operation after a hardware reset or a software reset, so the workaround described below need only to take place once during DWIO initialization. All software, including diagnostic programs, should be reviewed.

Word I/O mode is not affected.

Workaround: Do not set the PERREN bit (bit 6 in the PCI Command Register) while initializing to DWIO mode. Once the device is in DWIO mode, the PERREN bit can then be set to enable parity error response functions.

Status: No current plan to fix this item.

3) **Description:** The ***DXCVR and ERA[7:0] pins may output unexpected values in JTAG mode.***

Implication: Normal JTAG testing is not affected. NAND Tree testing is not affected.

Workaround: Ignore the outputs on the DXCVR and the ERA[7:0] pins in JTAG mode.

Status: No current plan to fix this item.

4) **Description:** The ***GPSI (General Purpose Serial Interface) mode does not function***, in contrast to previous versions of the PCnet-PCI II data sheet specifications which indicated that it does function.

Implication: The GPSI mode cannot be used. This impacts all applications that interface the PCnet-PCI II device to other devices via the GPSI port, such as interfacing to an external SIA and transceiver combination.

Workaround: None. Do not use the device in GPSI mode. All references to the GPSI mode has been deleted from this version of the PCnet-PCI II datasheet specification.

Status: No current plan to fix this item.

5) **Description:** When ***last-bus-mastering parking*** is selected and when the PCnet-PCI II controller is parked more than 33 PCI clock cycles after the controller has completed a DMA write cycle, if the parking GNT# is asserted at the same time of PCnet-PCI II controller's assertion of REQ#, in an extreme boundary case, the following DMA write may contain incorrect data.

Implication: The device only has this problem under the above described extreme boundary condition.

Workaround: Do not use last-bus-master parking. Or, when using last-bus-master parking, ensure that the PCnet-PCI II controller's parking GNT# is asserted within 33 PCI clocks after the controller has completed a bus mastering cycle.

Status: No current plan to fix this item.

6) **Description:** The ***Link Status LED appears to be stuck ON*** when the TMAU receive pins (RXD±) are receiving negative polarity link pulses and the Disable Automatic Polarity Correction bit (DAPC; CSR15, bit 11) is set.

Implication: The Link Status LED may not indicate correct link status.

Workaround: Do not set the DAPC bit to one. Always use the automatic polarity correction feature.

Status: No current plan to fix this item.

7) **Description:** The ***LAPP mode does not function properly*** on the PCnet-PCI II device. In LAPP mode, the PCnet-PCI II device might skip buffers when writing a receive frame to memory.

Implication: The LAPP function cannot be used.

Workaround: None.

Status: No current plan to fix this item.

8) **Description:** When a ***transmit frame has excessively deferred*** to receive activity, the EXDEF bit is not set in the transmit descriptor.

Implication: Software cannot use the EXDEF bit in the transmit descriptor as an indication of excessive deferral.

Workaround: The excessive deferral interrupt (EXDINT; CSR5, bit 7) does function correctly. User can enable this interrupt by setting the EXDINTE bit in CSR5.

Status: No current plan to fix this item.

9) **Description:** The PCnet-PCI device will ***intermittently give BABL error indications*** when the network traffic has frames equal to or greater than 1518 bytes.

Implication: False BABL errors on the receiving station can be passed up to the upper layer software if the PCnet-PCI II device is just coming out of deferral and the multi-purpose counter used to count the number of bytes received reaches 1518 at the same time. If the network is heavily loaded with full-size frames, then the probability of a false BABL error is high.

Workaround: There are two possible workarounds.

1. If the user has no intention to transmit frames larger than 1518 bytes, then the BABL bit may be masked to ignore babble errors. In this case the false babble error will not cause an interrupt, nor will it be passed to the higher level software.
2. Check to see if the device is transmitting in ISR (Interrupt Service Routine), which is induced by the BABL error. The BCRs which control the LED settings can be programmed to indicate a transmit activity, assuming the interrupt latency is not longer than one minimum IFG (inter-frame gap) time.

```
If (ISR_LATENCY <9.6 us)
    True_bable_err=BABL*(TINT + XMT_LED)
    {i.e. False_bable_err=~(BABL*(TINT + XMT_LED))}
else
    Cannot tell if the BABL error is true or false just by reading BABL, TINT, XMT_LED bits in ISR.
```

Status: No current plan to fix this item.

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AM's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Trademarks