Am79C971

PCnet™-*FAST*
Single-Chip Full-Duplex 10/100 Mbps Ethernet Controller for PCI Local Bus

DISTINCTIVE CHARACTERISTICS

- **Single-chip Fast Ethernet controller for the Peripheral Component Interconnect (PCI) local bus**
  - **— 32-bit glueless PCI host interface**
  - **— Supports PCI clock frequency from DC to 33 MHz independent of network clock**
  - **— Supports network operation with PCI clock from 15 MHz to 33 MHz**
  - **— High performance bus mastering architecture with integrated Direct Memory Access (DMA) Buffer Management Unit for low CPU and bus utilization**
  - **— PCI specification revision 2.1 compliant**
  - **— Supports PCI Subsystem/Subvendor ID/ Vendor ID programming through the EEPROM interface**
  - **— Supports both PCI 5.0-V and 3.3-V signaling environments**
  - **— Plug and Play compatible**
  - **— Supports an unlimited PCI burst length**
  - **— Big endian and little endian byte alignments supported**
- **Integrated 10BASE-T and 10BASE-2/5 (AUI) Physical Layer Interface**
  - **— Single-chip IEEE/ANSI 802.3, IEC/ISO 8802-3 and Blue Book Ethernet-compliant solution**
  - **— Automatic Twisted-Pair receive polarity detection and correction**
  - **— Internal 10BASE-T transceiver with Smart Squelch to Twisted-Pair medium**
  - **— IEEE 802.3-compliant auto-negotiable 10BASE-T interface**
- **Supports General Purpose Serial Interface (GPSI)**
- **Media Independent Interface (MII) for connecting external 10- or 100-Megabit per second (Mbps) transceivers**
  - **— IEEE 802.3-compliant MII**
  - **— Intelligent Auto-Poll™ external PHY status monitor and interrupt**
  - **— Includes intelligent on-chip Network Port Manager that provides auto-port selection between MII, on-chip 10BASE-T port, and AUI without software support**
  - **— Supports both auto-negotiable and non auto-negotiable external PHYs**
  - **— Supports 10BASE-T, 100BASE-TX/FX, 100BASE-T4, and 100BASE-T2 IEEE 802.3-compliant MII PHYs at full- or half-duplex**
- **Internal/external loopback capabilities on all ports**
- **Supports patented External Address Detection Interface (EADI)**
  - **— Receive frame tagging support for inter-networking applications**
- **Dual-speed CSMA/CD (10 Mbps and 100 Mbps) Media Access Controller (MAC) compliant with IEEE/ANSI 802.3 and Blue Book Ethernet standards**
- **Full-duplex operation supported in AUI, 10BASE-T, MII, and GPSI ports with independent Transmit (TX) and Receive (RX) channels**
- **Flexible buffer architecture**
  - **— Large independent internal TX and RX FIFOs**
  - **— SRAM-based FIFO buffer extension supporting up to 128 kilobytes (Kbytes)**
  - **— 1/2 Gigabit per second (Gbps) internal data bandwidth**
  - **— Programmable FIFO watermarks for both TX and RX operations**
  - **— RX frame queuing for high latency PCI bus host operation**
  - **— Programmable allocation of buffer space between RX and TX queues**
- **EEPROM interface supports jumperless design and provides through-chip programming**
  - **— Supports full programmability of half-/full-duplex operation for external 100 Mbps PHYs through EEPROM mapping**
- **Extensive LED status support**

Publication# **20550** Rev: **E** Amendment: **/0**
Issue Date: **May 2000**

- **Supports up to 1 Megabyte (Mbyte) optional Boot PROM and Flash for diskless node application**
- **Look-Ahead Packet Processing (LAPP) data handling technique reduces system overhead by allowing protocol analysis to begin before the end of a receive frame**
- **Includes Programmable Inter Packet Gap (IPG) to address less network aggressive MAC controllers**
- **Offers the Modified Back-Off algorithm to address the *Ethernet Capture Effect***
- **IEEE 1149.1-compliant JTAG Boundary Scan test access port interface and NAND tree test mode for board-level production connectivity test**
- **Implements low-power management for critical battery powered application and green PCs**
  - **— Includes two power-saving sleep modes (sleep and snooze)**
  - **— Integrated Magic Packet™ technology support for remote power of networked PCs**
- **Software compatible with AMD PCnet Family and LANCE/C-LANCE register and descriptor architecture**
- **Compatible with the existing PCnet Family driver/diagnostic software**
- **Available in 160-pin TQFP and 176-pin TQFP packages**

GENERAL DESCRIPTION

The Am79C971 controller is a single-chip 32-bit full-duplex, 10/100-Megabit per second (Mbps) 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 and bridge/router designs. The bus master architecture provides high data throughput in the system and low CPU and system bus utilization. The Am79C971 controller is fabricated with AMD's advanced low-power Complementary Metal Oxide Semiconductor (CMOS) process to provide low operating and standby current for power sensitive applications.

The Am79C971 controller is a complete Ethernet node integrated into a single VLSI device. It contains a bus interface unit, a Direct Memory Access (DMA) Buffer Management Unit, an ISO/IEC 8802-3 (IEEE 802.3)-compliant Media Access Controller (MAC), a large Transmit FIFO and a large Receive FIFO, SRAM-based FIFO extension with support for up to 128K bytes of external frame buffering, an IEEE 802.3u-compliant MII, an IEEE 802.3-compliant Twisted-Pair Transceiver Media Attachment Unit (10BASE-T MAU), and an IEEE 802.3-compliant Attachment Unit Interface (AUI). Both proprietary full-duplex and IEEE 802.3 compliant half-duplex operation are supported on the MII, AUI, GPSI, and 10BASE-T MAU interfaces. 10-Mbps operation is supported through the MII, AUI, and 10BASE-T MAU interfaces, and 100 Mbps operation is supported through the MII. The 10BASE-T MAU interface includes an IEEE 802.3-compliant auto-negotiation implementation, which will automatically negotiate between half- and full-duplex with another IEEE 802.3-compliant auto-negotiation 10BASE-T device.

The Am79C971 controller is register compatible with the LANCE (Am7990) Ethernet controller, the C-LANCE (Am79C90) Ethernet controller, and all Ethernet controllers in the PCnet Family *except* ILACC (Am79C900), including the PCnet-ISA controller (Am79C960),PCnet-ISA+ controller (Am79C961), PCnet-ISA II controller (Am79C961A), PCnet-32 controller (Am79C965), PCnet-PCI controller (Am79C970), and PCnet-PCI II controller (Am79C970A). The Buffer Management Unit supports the LANCE and PCnet descriptor software models.

The 32-bit multiplexed bus interface unit provides a direct interface to the PCI local bus, simplifying the design of an Ethernet node in a PC system. The Am79C971 controller provides the complete interface to an Expansion ROM or Flash device 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 non-PC applications. The Am79C971 controller's advanced CMOS design allows the bus interface to be connected to either a +5-V or a +3.3-V signaling environment. A compliant IEEE 1149.1 JTAG test interface for board-level testing is also provided, as well as a NAND tree test structure for those systems that cannot support the JTAG interface.

The Am79C971 controller supports auto-configuration in the PCI configuration space. Additional Am79C971 controller configuration parameters, including the unique IEEE physical address, can be read from an external non-volatile memory (EEPROM) immediately following system reset.

The integrated Manchester encoder/decoder (MENDEC) eliminates the need for an external Serial Interface Adapter (SIA) in the system. The built-in GPSI allows the MENDEC to be bypassed.

In addition, the device provides programmable on-chip LED drivers for transmit, receive, collision, receive polarity, link integrity, activity, link active, address match, full-duplex, MII select, 100 Mbps, or jabber status. The Am79C971 controller also provides an EADI to allow external hardware address filtering in internetworking applications and a receive frame tagging feature.

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-*FAST* controller also has integrated functions to support Magic Packet technology, an inexpensive technology that allows remote wake up of networked PCs.

The controller has the capability to automatically select either the MII, AUI, or Twisted-Pair transceiver. Only one interface is active at any one time. Any of the network interfaces can be programmed to operate in either half-duplex or full-duplex mode (AUI full-duplex only supports the 10BASE-F standard).

The dual Transmit and Receive FIFOs optimize system overhead, providing sufficient latency tolerance at 10 Mbps and for 100-Mbps systems where low latencies can be guaranteed during frame transmission and reception.

In highly loaded 10-Mbps systems, such as servers or when using the controller in a 100-Mbps environment, the additional frame buffering capability provided by a 16-bit wide SRAM interface provides high performance and high latency tolerance on the system bus and network.

The Am79C971 controller can use up to 128 Kbytes of SRAM as an extension of its dual Transmit and Receive FIFOs. When no SRAM is used, the Am79C971 controller's FIFOs are programmed to bypass the SRAM interface.

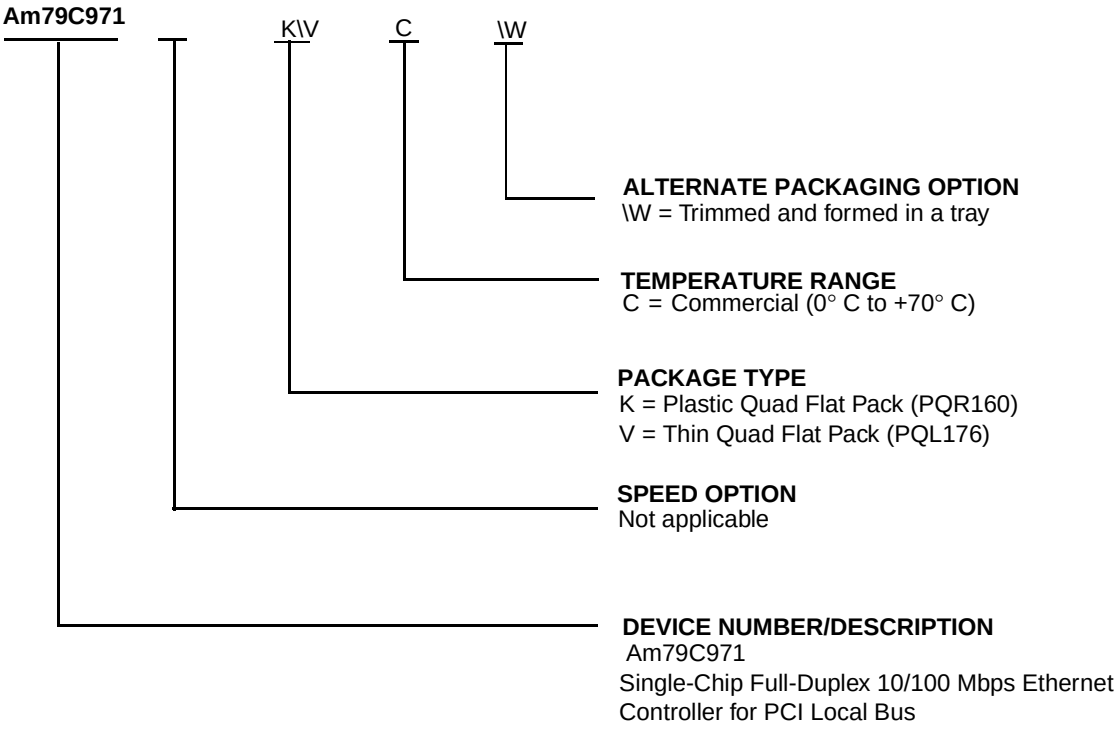
IMPORTANT NOTE: A "No SRAM configuration" is only valid for 10Mb mode. In 100Mb mode, SRAM is mandatory and must always be used.

ISO/IEC 8802-3 and IEEE 802.3 will be used interchangeably when referring to half-duplex 10 Mbps networks. IEEE 802.3 or IEEE 802.3u will be used interchangeably only when referring to half-duplex 100-Mbps Ethernet networks, since the IEEE standard is not ISO approved yet. Full-duplex is a proprietary standard and is not approved by IEEE or ISO.

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 the elements below.

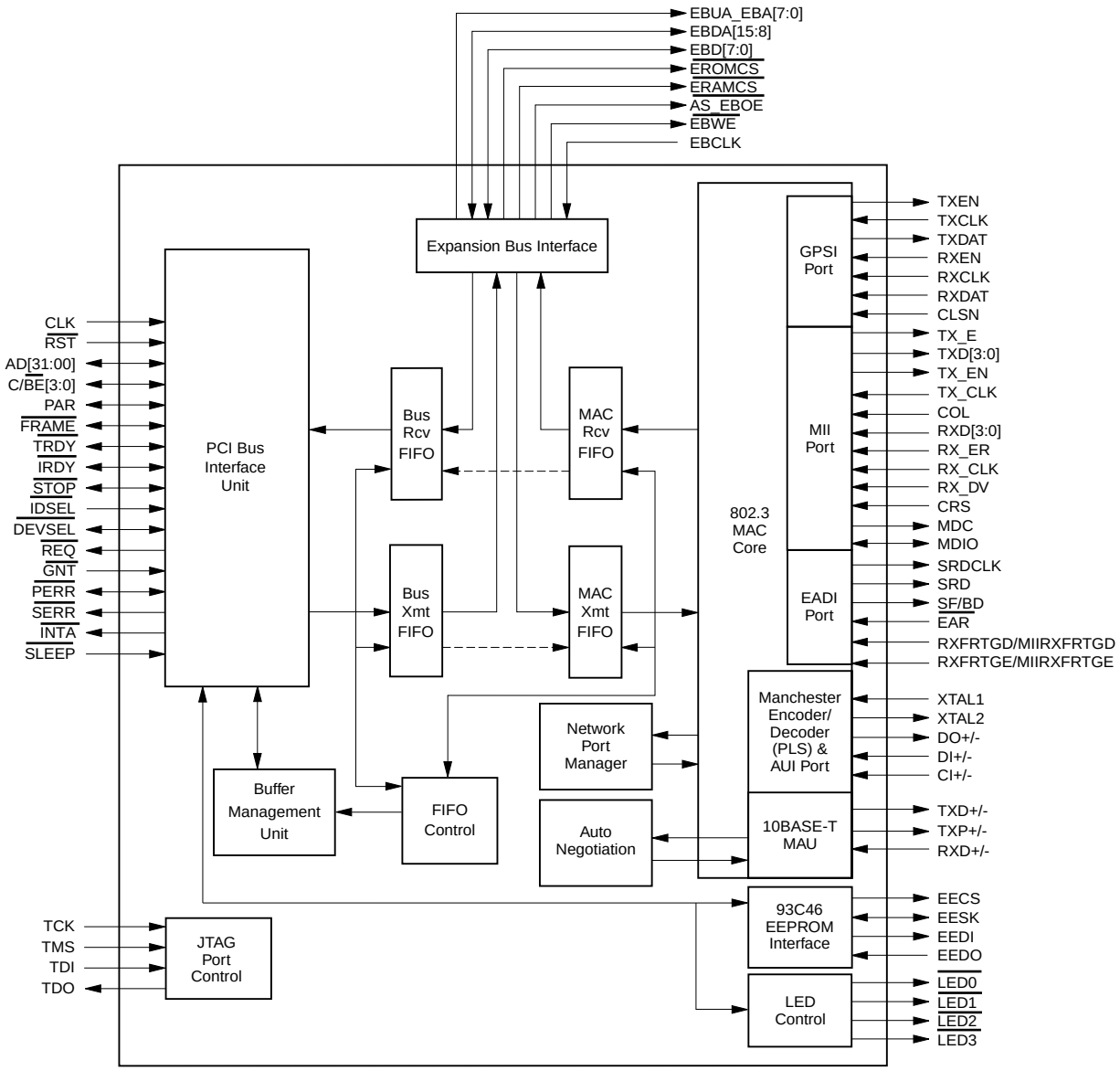


Valid Combinations	
Am79C971	KC\W, VC\W

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.

BLOCK DIAGRAM



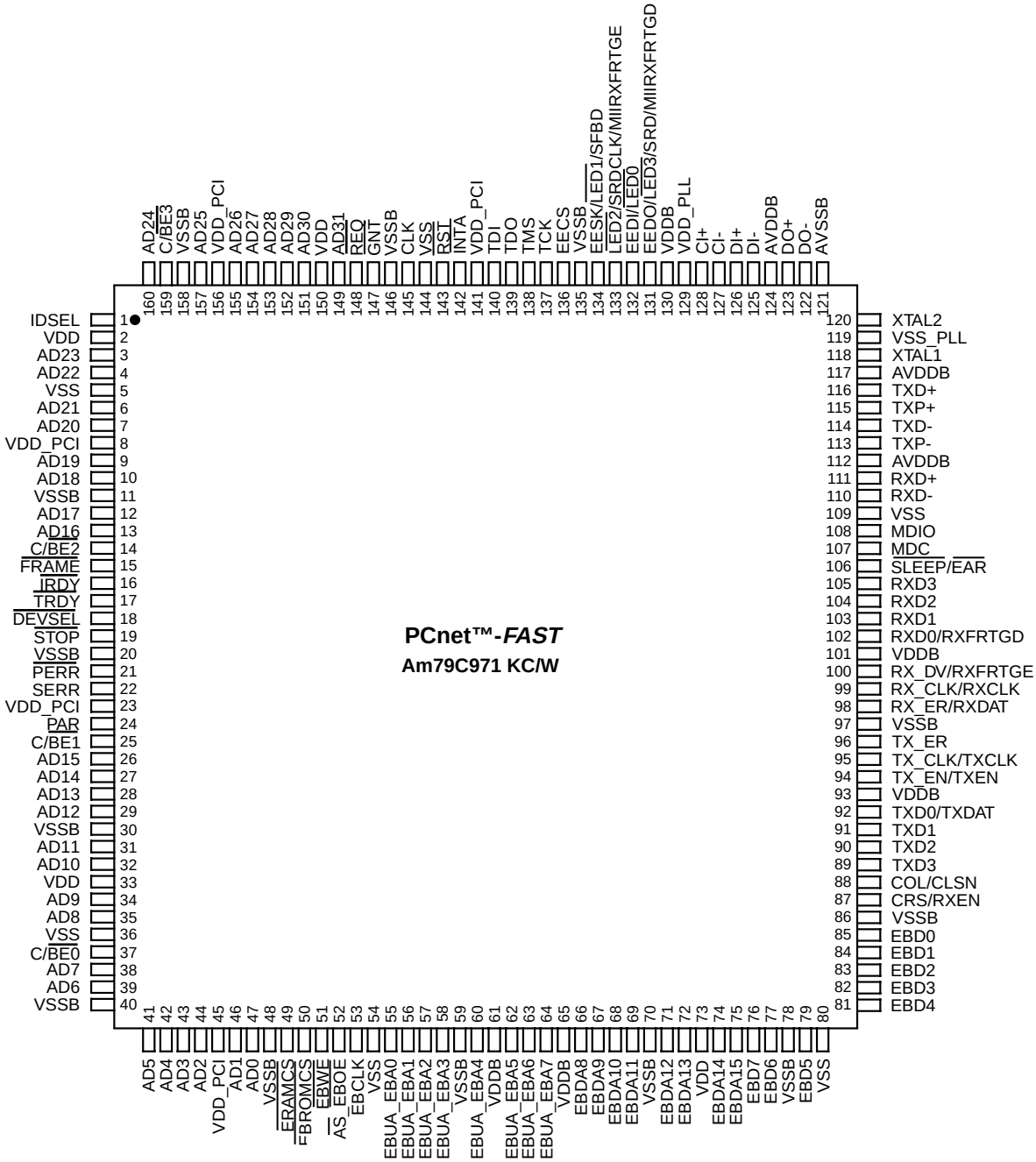
20550D-1

TABLE OF CONTENTS

RELATED AMD PRODUCTS

Part No.	Description
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am7996	IEEE 802.3/Ethernet/Cheapernet Tap Transceiver
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Am79865	100 Mbps Physical Data Transmitter (PDT)
An79866A	100 Mbps Physical Data Receiver (PDR)
Am79C870	Quad 100BASE-X Transceiver
Am79C871	Quad 100BASE-X Repeater Transceiver
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller (with Microsoft® Plug n' Play support)
Am79C961A	PCnet-ISA II Single-Chip Full-Duplex Ethernet Controller (with Microsoft® Plug n' Play support)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller (for 486 and VL buses)
Am79C970A	PCnet-PCI II Single-Chip Full-Duplex Ethernet Controller for PCI Local Bus
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)

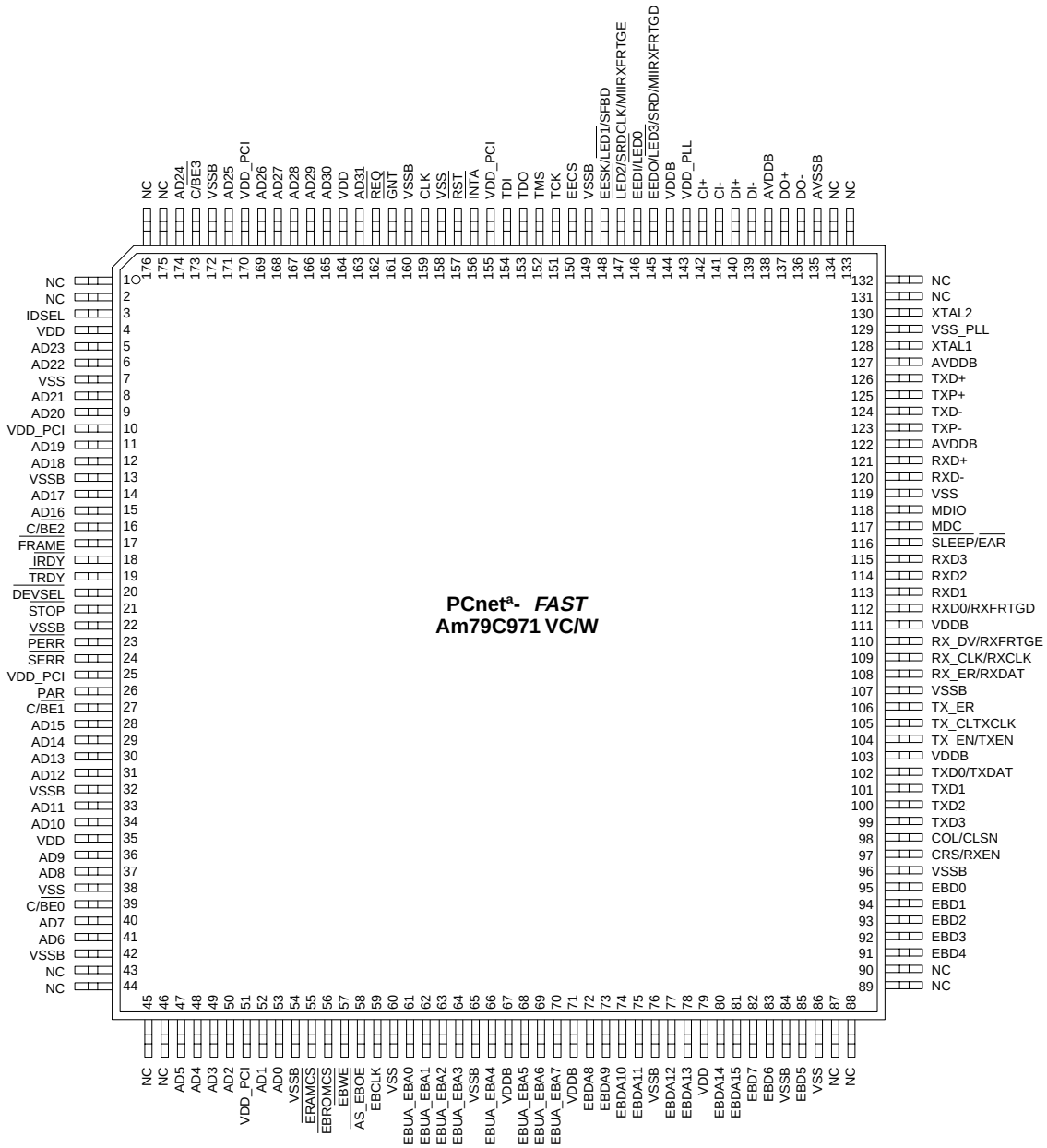
CONNECTION DIAGRAM (PQR160)



Pin 1 is marked for orientation.

20550D-2

CONNECTION DIAGRAM (PQL176)



20550D-3

Pin 1 is marked for orientation.

PIN DESIGNATIONS (PQR160)

Listed By Pin Number

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	IDSEL	41	AD5	81	EBD4	121	AVSSB
2	VDD	42	AD4	82	EBD3	122	DO-
3	AD23	43	AD3	83	EBD2	123	DO+
4	AD22	44	AD2	84	EBD1	124	AVDDB
5	VSS	45	VDD_PCI	85	EBD0	125	DI-
6	AD21	46	AD1	86	VSSB	126	DI+
7	AD20	47	AD0	87	CRS/RXEN	127	CI-
8	VDD_PCI	48	VSSB	88	COL/CLSN	128	CI+
9	AD19	49	ERAMCS	89	TXD3	129	VDD_PLL
10	AD18	50	EROMCS	90	TXD2	130	VDDB
11	VSSB	51	EBWE	91	TXD1	131	EEDO/LED3/SRD/ MIIRXFRTGD
12	AD17	52	AS_EBOE	92	TXD0/TXDAT	132	EED1/LED0
13	AD16	53	EBCLK	93	VDDB	133	LED2/SRDCLK/ MIIRXFRTGE
14	C/BE2	54	VSS	94	TX_EN/TXEN	134	EESK/LED1/SFBD
15	FRAME	55	EBUA_EBA0	95	TX_CLK/TXCLK	135	VSSB
16	IRDY	56	EBUA_EBA1	96	TX_ER	136	EECS
17	TRDY	57	EBUA_EBA2	97	VSSB	137	TCK
18	DEVSEL	58	EBUA_EBA3	98	RX_ER/RXDAT	138	TMS
19	STOP	59	VSSB	99	RX_CLK/RXCLK	139	TDO
20	VSSB	60	EBUA_EBA4	100	RX_DV/RXFRTGE	140	TDI
21	PERR	61	VDDB	101	VDDB	141	VDD_PCI
22	SERR	62	EBUA_EBA5	102	RXD0/RXFRTGD	142	INTA
23	VDD_PCI	63	EBUA_EBA6	103	RXD1	143	RST
24	PAR	64	EBUA_EBA7	104	RXD2	144	VSS
25	C/BE1	65	VDDB	105	RXD3	145	CLK
26	AD15	66	EBDA8	106	SLEEP/EAR	146	VSSB
27	AD14	67	EBDA9	107	MDC	147	GNT
28	AD13	68	EBDA10	108	MDIO	148	REQ
29	AD12	69	EBDA11	109	VSS	149	AD31
30	VSSB	70	VSSB	110	RXD-	150	VDD
31	AD11	71	EBDA12	111	RXD+	151	AD30
32	AD10	72	EBDA13	112	AVDDB	152	AD29
33	VDD	73	VDD	113	TXP-	153	AD28
34	AD9	74	EBDA14	114	TXD-	154	AD27
35	AD8	75	EBDA15	115	TXP+	155	AD26
36	VSS	76	EBD7	116	TXD+	156	VDD_PCI
37	C/BE0	77	EBD6	117	AVDDB	157	AD25
38	AD7	78	VSSB	118	XTAL1	158	VSSB
39	AD6	79	EBD5	119	VSS_PLL	159	C/BE3
40	VSSB	80	VSS	120	XTAL2	160	AD24

PIN DESIGNATIONS (PQL176)

Listed By Pin Number

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	NC	45	NC	89	NC	133	NC
2	NC	46	NC	90	NC	134	NC
3	IDSEL	47	AD5	91	EBD4	135	AVSSB
4	VDD	48	AD4	92	EBD3	136	DO-
5	AD23	49	AD3	93	EBD2	137	DO+
6	AD22	50	AD2	94	EBD1	138	AVDDB
7	VSS	51	VDD_PCI	95	EBD0	139	DI-
8	AD21	52	AD1	96	VSSB	140	DI+
9	AD20	53	AD0	97	CRS/RXEN	141	CI-
10	VDD_PCI	54	VSSB	98	COL/CLSN	142	CI+
11	AD19	55	$\overline{\text{ERAMCS}}$	99	TXD3	143	VDD_PLL
12	AD18	56	$\overline{\text{EROMCS}}$	100	TXD2/RXEN	144	VDDB
13	VSSB	57	$\overline{\text{EBWE}}$	101	TXD1	145	EEDO/$\overline{\text{LED3}}$/SRD/ MIIRXFRTGD
14	AD17	58	AS_$\overline{\text{EBOE}}$	102	TXD0/TXDAT	146	EED1/$\overline{\text{LED0}}$
15	AD16	59	EBCLK	103	VDDB	147	$\overline{\text{LED2}}$/SRDCLK/ MIIRXFRTGE
16	C/$\overline{\text{BE2}}$	60	VSS	104	TX_EN/TXEN	148	EESK/$\overline{\text{LED1}}$/SFBD
17	$\overline{\text{FRAME}}$	61	EBUA_EBA0	105	TX_CLK/TXCLK	149	VSSB
18	$\overline{\text{IRDY}}$	62	EBUA_EBA1	106	TX_ER	150	EECS
19	$\overline{\text{TRDY}}$	63	EBUA_EBA2	107	VSSB	151	TCK
20	$\overline{\text{DEVSEL}}$	64	EBUA_EBA3	108	RX_ER/RXDAT	152	TMS
21	$\overline{\text{STOP}}$	65	VSSB	109	RX_CLK/RXCLK	153	TDO
22	VSSB	66	EBUA_EBA4	110	RX_DV/RXFRTGE	154	TDI
23	$\overline{\text{PERR}}$	67	VDDB	111	VDDB	155	VDD_PCI
24	$\overline{\text{SERR}}$	68	EBUA_EBA5	112	RXD0/RXFRTGD	156	$\overline{\text{INTA}}$
25	VDD_PCI	69	EBUA_EBA6	113	RXD1	157	$\overline{\text{RST}}$
26	PAR	70	EBUA_EBA7	114	RXD2	158	VSS
27	C/$\overline{\text{BE1}}$	71	VDDB	115	RXD3	159	CLK
28	AD15	72	EBDA8	116	$\overline{\text{SLEEP}}$/$\overline{\text{EAR}}$	160	VSSB
29	AD14	73	EBDA9	117	MDC	161	$\overline{\text{GNT}}$
30	AD13	74	EBDA10	118	MDIO	162	$\overline{\text{REQ}}$
31	AD12	75	EBDA11	119	VSS	163	AD31
32	VSSB	76	VSSB	120	RXD-	164	VDD
33	AD11	77	EBDA12	121	RXD+	165	AD30
34	AD10	78	EBDA13	122	AVDDB	166	AD29
35	VDD	79	VDD	123	TXP-	167	AD28
36	AD9	80	EBDA14	124	TXD-	168	AD27
37	AD8	81	EBDA15	125	TXP+	169	AD26
38	VSS	82	EBD7	126	TXD+	170	VDD_PCI
39	C/$\overline{\text{BE0}}$	83	EBD6	127	AVDDB	171	AD25
40	AD7	84	VSSB	128	XTAL1	172	VSSB
41	AD6	85	EBD5	129	VSS_PLL	173	C/$\overline{\text{BE3}}$
42	VSSB	86	VSS	130	XTAL2	174	AD24
43	NC	87	NC	131	NC	175	NC
44	NC	88	NC	132	NC	176	NC

PIN DESIGNATIONS (PQR160, PQL176)

Listed By Group

Pin Name	Pin Function	Type[1]	Driver	No. of Pins
PCI Bus Interface				
AD[31:0]	Address/Data Bus	IO	TS3	32
C/$\overline{BE}$[3:0]	Bus Command/Byte Enable	IO	TS3	4
CLK	Bus Clock	I	NA	1
$\overline{DEVSEL}$	Device Select	IO	STS6	1
$\overline{FRAME}$	Cycle Frame	IO	STS6	1
$\overline{GNT}$	Bus Grant	I	NA	1
IDSEL	Initialization Device Select	I	NA	1
$\overline{INTA}$	Interrupt	O	OD6	1
$\overline{IRDY}$	Initiator Ready	IO	STS6	1
PAR	Parity	IO	TS3	1
$\overline{PERR}$	Parity Error	IO	STS6	1
$\overline{REQ}$	Bus Request	O	TS3	1
$\overline{RST}$	Reset	I	NA	1
$\overline{SERR}$	System Error	IO	OD6	1
$\overline{STOP}$	Stop	IO	STS6	1
$\overline{TRDY}$	Target Ready	IO	STS6	1
Board Interface				
$\overline{LED0}$	LED0	O	LED	1
$\overline{LED1}$	LED1	O	LED	1
$\overline{LED2}$	LED2	O	LED	1
$\overline{LED3}$	LED3	O	LED	1
$\overline{SLEEP}$	Sleep Mode	I	NA	1
XTAL1	Crystal Input	I	NA	1
XTAL2	Crystal Output	O	XTAL	1
EEPROM Interface				
EECS	Serial EEPROM Chip Select	O	O6	1
EEDI	Serial EEPROM Data In	O	LED	1
EEDO	Serial EEPROM Data Out	I	NA	1
EESK	Serial EEPROM Clock	IO	LED	1
Expansion ROM Interface				
AS_$\overline{EBOE}$	Address Strobe/Expansion Bus Output Enable	O	O6	1
EBCLK	Expansion Bus Clock	I	NA	1
EBD[7:0]	Expansion Bus Data [7:0]	IO	TS6	8
EBDA[15:8]	Expansion Bus Data/Address [15:8]	IO	TS6	8
EBUA_EBA[7:0]	Expansion Bus Upper Address /Expansion Bus Address [7:0]	O	O6	8
$\overline{EBWE}$	Expansion Bus Write Enable	O	O6	1
$\overline{ERAMCS}$	Expansion Bus RAM Chip Select	O	O6	1
$\overline{EROMCS}$	Expansion Bus ROM Chip Select	O	O6	1

Note:

1. Not including test features

PIN DESIGNATIONS

Listed By Group

Pin Name	Pin Function	Type[1]	Driver	No. of Pins
Media Independent Interface (MII)				
COL	Collision	I	NA	1
CRS	Carrier Sense	I	NA	1
MDC	Management Data Clock	O	OMII2	1
MDIO	Management Data I/O	IO	TSMII	1
RX_CLK	Receive Clock	I	NA	1
RXD[3:0]	Receive Data	I	NA	4
RX_DV	Receive Data Valid	I	NA	1
RX_ER	Receive Error	I	NA	1
TX_CLK	Transmit Clock	I	NA	1
TXD[3:0]	Transmit Data	O	OMII1	4
TX_EN	Transmit Data Enable	O	OMII1	1
TX_ER	Transmit Error	O	OMII1	1
Attachment Unit Interface (AUI)				
CI±	AUI Collision	I	NA	1
DI±	AUI Data In	I	NA	1
DO±	AUI Data Out	O	DO	1
10BASE-T Interface				
RXD+/RXD-	Receive Differential Pair	I	NA	2
TXD+/TXD-	Transmit Differential Pair	O	TDO	2
TXP+/TXP-	Transmit Pre-distortion Differential Pair	O	TPO	2
General Purpose Serial Interface (GPSI)				
CLSN	Collision	IO	NA	1
RXCLK	Receive Clock	I	NA	1
RXDAT	Receive Data	I	NA	1
RXEN	Receive Enable	I	NA	1
TXCLK	Transmit Clock	I	NA	1
TXDAT	Transmit Data	O	O6	1
TXEN	Transmit Enable	O	O6	1
External Address Detection Interface (EADI)				
$\overline{EAR}$	External Address Reject Low	I	NA	1
SFBD	Start Frame Byte Delimiter	O	LED	1
SRD	Serial Receive Data	IO	LED	1
SRDCLK	Serial Receive Data Clock	IO	LED	1
RXFRTGD/MIIRXFRTGD	Receive Frame Tag Data/MII Receive Frame Tag Data	I	NA	1
RXFRTGE/MIIRXFRTGE	Receive Frame Tag Enable/MII Receive Frame Tag Enable	I	NA	1
IEEE 1149.1 Test Access Port Interface (JTAG)				
TCK	Test Clock	I	NA	1
TDI	Test Data In	I	NA	1
TDO	Test Data Out	O	TS6	1
TMS	Test Mode Select	I	NA	1

Note:

1. Not including test features.

PIN DESIGNATIONS

Listed By Group

Pin Name	Pin Function	Type[1]	Driver	No. of Pins
Power Supplies				
AVDDB	Analog I/O Buffer Power	P	NA	3
AVSSB	Analog I/O Buffer Ground	P	NA	1
VDD_PLL	Analog PLL Power	P	NA	1
VSS_PLL	Analog PLL Ground	P	NA	1
VDD	Digital Power	P	NA	4
VSS	Digital Ground	P	NA	6
VDDB	I/O Buffer Power	P	NA	5
VSSB	I/O Buffer Ground	P	NA	13
VDD_PCI	PCI I/O Buffer Power	P	NA	5

Note:

1. Not including test features.

Listed By Driver Type

The following table describes the various types of output drivers used in the Am79C971 controller. All I_{OL} and I_{OH} values shown in the table apply to 5 V signaling. See the *DC Characteristics* section 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. Their characteristics and the one of the XTAL output are described in the *DC Characteristics* section.

Name	Type	I_{OL} (mA)	I_{OH} (mA)	Load (pF)
LED	LED	12	-0.4	50
OMII1	Totem Pole	4	-4	50
OMII2	Totem Pole	4	-4	390
O6	Totem Pole	6	-0.4	50
OD6	Open Drain	6	NA	50
STS6	Sustained Tri-State	6	-2	50
TS3	Tri-State	3	-2	50
TS6	Tri-State	6	-2	50
TSMII	Tri-State	4	-4	470

PIN DESCRIPTIONS

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[07:0] are defined as the least significant byte (LSB) and AD[31:24] are defined as the most significant byte (MSB). For FIFO data transfers, the Am79C971 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 Am79C971 controller is a bus master, AD[31:2] will address the active Double Word (DWord). The Am79C971 controller always drives AD[1:0] to '00' during the address phase indicating linear burst order. When the Am79C971 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 Am79C971 controller when performing bus master write and slave read operations. Data on AD[31:0] is latched by the Am79C971 controller when performing bus master read and slave write operations.

When $\overline{RST}$ is active, AD[31:0] are inputs for NAND tree testing.

C/$\overline{BE}$[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/$\overline{BE}$[3:0] define the bus command. During the data phase, C/$\overline{BE}$[3:0] are used as byte enables. The byte enables define which physical byte lanes carry meaningful data. C/$\overline{BE}$0 applies to byte 0 (AD[07:0]) and C/$\overline{BE}$3 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 $\overline{RST}$ is active, C/$\overline{BE}$[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 Am79C971 controller normally operates over a frequency range of 10 to 33 MHz on the PCI bus due to networking demands. See the *Frequency Demands for Network Operation* section for details. The Am79C971 controller will support a clock frequency of 0 MHz after certain precautions are taken to ensure data integrity. This clock or a derivation is not used to drive any network functions.

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

$\overline{DEVSEL}$

Device Select **Input/Output**

The Am79C971 controller drives $\overline{DEVSEL}$ when it detects a transaction that selects the device as a target. The device samples $\overline{DEVSEL}$ to detect if a target claims a transaction that the Am79C971 controller has initiated.

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

$\overline{FRAME}$

Cycle Frame **Input/Output**

$\overline{FRAME}$ is driven by the Am79C971 controller when it is the bus master to indicate the beginning and duration of a transaction. $\overline{FRAME}$ is asserted to indicate a bus transaction is beginning. $\overline{FRAME}$ is asserted while data transfers continue. $\overline{FRAME}$ is deasserted before the final data phase of a transaction. When the Am79C971 controller is in slave mode, it samples $\overline{FRAME}$ to determine the address phase of a transaction.

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

$\overline{GNT}$

Bus Grant **Input**

This signal indicates that the access to the bus has been granted to the Am79C971 controller.

The Am79C971 controller supports bus parking. When the PCI bus is idle and the system arbiter asserts $\overline{GNT}$ without an active $\overline{REQ}$ from the Am79C971 controller, the device will drive the AD[31:0], C/$\overline{BE}$[3:0] and PAR lines.

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

IDSEL

Initialization Device Select **Input**

This signal is used as a chip select for the Am79C971 controller during configuration read and write transactions.

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

1. Not including test features.

$\overline{\text{INTA}}$

Interrupt Request **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, UINT, MCCIINT, MCCINT, MPDTINT, MAPINT, MREINT, and STINT. Each status flag has either a mask or an enable bit which allows for suppression of $\overline{\text{INTA}}$ assertion. Table 1 shows the flag meanings.

Table 1. Interrupt Flags

Name	Description	Mask Bit	Interrupt Bit
BABL	Babble	CSR3, bit 14	CSR0, bit 14
EXDINT	Excessive Deferral	CSR5, bit 6	CSR5, bit 7
IDON	Initialization Done	CSR3, bit 8	CSR0, bit 8
JAB	Jabber	CSR4, bit 0	CSR4, bit 1
MERR	Memory Error	CSR3, bit 11	CSR0, bit 11
MISS	Missed Frame	CSR3, bit 12	CSR0, bit 12
MFCO	Missed Frame Count Overflow	CSR4, bit 8	CSR4, bit 9
MPINT	Magic Packet Interrupt	CSR5, bit 3	CSR5, bit 4
RCVCCO	Receive Collision Count Overflow	CSR4, bit 4	CSR4, bit 5
RINT	Receive Interrupt	CSR3, bit 10	CSR0, bit 10
SLPINT	Sleep Interrupt	CSR5, bit 8	CSR5, bit 9
SINT	System Error	CSR5, bit 10	CSR5, bit 11
TINT	Transmit Interrupt	CSR3, bit 9	CSR0, bit 9
TXSTRT	Transmit Start	CSR4, bit 2	CSR4, bit 3
UINT	User Interrupt	CSR4, bit 7	CSR4, bit 6
MCCIINT	Internal MII Management Command Complete Interrupt	CSR7, bit 2	CSR7, bit 3
MCCINT	MII Management Command Complete Interrupt	CSR7, bit 4	CSR7, bit 5
MPDTINT	MII PHY Detect Transition Interrupt	CSR7, bit 0	CSR7, bit 1

Table 1. Interrupt Flags

Name	Description	Mask Bit	Interrupt Bit
MAPINT	MII Auto-Poll Interrupt	CSR7, bit 6	CSR7, bit 7
MREINT	MII Management Frame Read Error Interrupt	CSR7, bit 8	CSR7, bit 9
STINT	Software Timer Interrupt	CSR7, bit 10	CSR7, bit 11

By default $\overline{\text{INTA}}$ is an open-drain output. For applications that need a high-active edge-sensitive interrupt signal, the $\overline{\text{INTA}}$ pin can be configured for this mode by setting INTLEVEL (BCR2, bit 7) to 1.

When $\overline{\text{RST}}$ is active, $\overline{\text{INTA}}$ is the output for NAND tree testing.

$\overline{\text{IRDY}}$

Initiator Ready **Input/Output**

$\overline{\text{IRDY}}$ indicates the ability of the initiator of the transaction to complete the current data phase. $\overline{\text{IRDY}}$ is used in conjunction with $\overline{\text{TRDY}}$. 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 Am79C971 controller is a bus master, it asserts $\overline{\text{IRDY}}$ during all write data phases to indicate that valid data is present on AD[31:0]. During all read data phases, the device asserts $\overline{\text{IRDY}}$ to indicate that it is ready to accept the data.

When the Am79C971 controller is the target of a transaction, it checks $\overline{\text{IRDY}}$ during all write data phases to determine if valid data is present on AD[31:0]. During all read data phases, the device checks $\overline{\text{IRDY}}$ to determine if the initiator is ready to accept the data.

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

PAR

Parity **Input/Output**

Parity is even parity across AD[31:0] and C/$\overline{\text{BE}}$[3:0]. When the Am79C971 controller is a bus master, it generates parity during the address and write data phases. It checks parity during read data phases. When the Am79C971 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 $\overline{\text{RST}}$ 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 Am79C971 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 1. During any master write transaction, the Am79C971 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 Am79C971 controller asserts REQ pin as a signal that it wishes to become a bus master. REQ is driven high when the Am79C971 controller does not request the bus. During Magic Packet™ mode, the REQ pin will not be driven.

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

RST

Reset Input

When RST is asserted low, then the Am79C971 controller performs an internal system reset of the type H_RESET (HARDWARE_RESET, see section on RESET). RST must be held for a minimum of 30 clock periods. While in the H_RESET state, the Am79C971 controller will disable or deassert all outputs. RST may be asynchronous to clock when asserted or deasserted.

When RST is active, NAND tree testing is enabled.

SERR

System Error **Input/Output**

During any slave transaction, the Am79C971 controller asserts SERR when it detects an address parity error, and reporting of the error is enabled by setting PEREN (PCI Command register, bit 6) and SERREN (PCI Command register, bit 8) to 1.

By default SERR is an open-drain output. For component test, it can be programmed to be an active-high totem-pole output.

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

STOP

Stop **Input/Output**

In slave mode, the Am79C971 controller drives the STOP signal to inform the bus master to stop the current transaction. In bus master mode, the Am79C971 controller checks STOP to determine if the target wants to disconnect the current transaction.

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

TRDY

Target Ready **Input/Output**

TRDY indicates the ability of the target of the transaction to complete the current data phase. 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 Am79C971 controller is a bus master, it checks 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 TRDY to determine if the target is ready to accept the data.

When the Am79C971 controller is the target of a transaction, it asserts 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 TRDY to indicate that it is ready to accept the data.

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

Board Interface

Note: *Before programming the LED pins, see the description of LEDPE in BCR2, bit 12 first.*

LED0

LED0 **Output**

This output is designed to directly drive an LED. By default, LED0 indicates an active link connection on the 10BASE-T interface. This pin can also be programmed to indicate other network status (see BCR4). The LED0 pin polarity is programmable, but by default it is active LOW. When the LED0 pin polarity is programmed to active LOW, the output is an open drain driver. When the LED0 pin polarity is programmed to active HIGH, the output is a totem pole driver.

Note: *The LED0 pin is multiplexed with the EEDI pin.*

When RST is active, LED0 is an input for NAND tree testing.

LED1

LED1 **Output**

This output is designed to directly drive an LED. By default, LED1 indicates receive activity on the network. This pin can also be programmed to indicate other network status (see BCR5). The LED1 pin polarity is programmable, but by default, it is active LOW. When the LED1 pin polarity is programmed to active LOW, the output is an open drain driver. When the LED1 pin po-

larity is programmed to active HIGH, the output is a totem pole driver.

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

The $\overline{\text{LED1}}$ pin is also used during EEPROM Auto-Detection to determine whether or not an EEPROM is present at the Am79C971 controller interface. At the last rising edge of CLK while $\overline{\text{RST}}$ is active LOW, $\overline{\text{LED1}}$ is sampled to determine the value of the EEDET bit in BCR19. It is important to maintain adequate hold time around the rising edge of the CLK at this time to ensure a correctly sampled value. A sampled HIGH value means that an EEPROM is present, and EEDET will be set to 1. A sampled LOW value means that an EEPROM is not present, and EEDET will be set to 0. See the *EEPROM Auto-Detection* section 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.

When $\overline{\text{RST}}$ is active, $\overline{\text{LED1}}$ is an input for NAND tree testing.

$\underline{\text{WARNING}}$: The input signal level of $\overline{\text{LED1}}$ must be insured for correct EEPROM detection before the deassertion of $\overline{\text{RST}}$.

$\overline{\text{LED2}}$

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. When the $\overline{\text{LED2}}$ pin polarity is programmed to active LOW, the output is an open drain driver. When the $\overline{\text{LED2}}$ pin polarity is programmed to active HIGH, the output is a totem pole driver.

Note: *The $\overline{\text{LED2}}$ pin is multiplexed with the SRDCLK pin and the MIIRXFRTGE pins.*

When $\overline{\text{RST}}$ is active, $\overline{\text{LED2}}$ is an input for NAND tree testing.

$\overline{\text{LED3}}$

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. When the $\overline{\text{LED3}}$ pin polarity is programmed to active LOW, the output is an open drain driver. When the $\overline{\text{LED3}}$ pin polarity is programmed to active HIGH, the output is a totem pole driver.

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 I_{OL} 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 on *LED Support*.

Note: *The $\overline{\text{LED3}}$ pin is multiplexed with the EEDO, SRD, MIIRXFRTGD pins.*

When $\overline{\text{RST}}$ is active, LED3 is an input for NAND tree testing.

$\overline{\text{SLEEP}}$

Sleep **Input**

When $\overline{\text{SLEEP}}$ is asserted, the Am79C971 controller performs an internal system reset of the H_RESET type and then proceeds into a power savings mode. All Am79C971 controller outputs will be placed in their normal reset condition. All Am79C971 controller inputs will be ignored except for the $\overline{\text{SLEEP}}$ pin itself. The system must refrain from starting the network operations of the Am79C971 controller for 0.5 seconds following the deassertion of the $\overline{\text{SLEEP}}$ pin in order to allow internal analog circuits to stabilize.

For effects with the Magic Packet™ modes, see the *Magic Packet* section.

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

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

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

Note: *The $\overline{\text{SLEEP}}$ pin is multiplexed with the $\overline{\text{EAR}}$ pin.*

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

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 on *External Crystal Characteristics* for more details. This clock is always required whether or not the internal 10BASE-T/AUI ports are enabled. If the internal PHY is

not used, ±10% accuracy is sufficient for the clock source.

Note: *When the Am79C971 controller is in coma mode, there is an internal 22 kΩ resistor from XTAL1 to ground. If an external source drives XTAL1, some power consumption 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 $\overline{SLEEP}$ and deassertion of $\overline{REQ}$.*

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 XTAL2 should be left unconnected.

EEPROM Interface

EECS

EEPROM Chip Select **Output**

This pin is designed to directly interface to a serial EEPROM that uses the 93C46 EEPROM interface protocol. EECS is connected to the EEPROM's chip select pin. It is controlled by either the Am79C971 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 2.

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

EEDI

EEPROM Data In **Output**

This pin is designed to directly interface to a serial EEPROM that uses the 93C46 EEPROM interface protocol. EEDI is connected to the EEPROM's data input pin. It is controlled by either the Am79C971 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 0.

Note: *The EEDI pin is multiplexed with the $\overline{LED0}$ pin.*

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

EEDO

EEPROM Data Out **Input**

This pin is designed to directly interface to a serial EEPROM that uses the 93C46 EEPROM interface protocol. EEDO is connected to the EEPROM's data output pin. It is controlled by either the Am79C971 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by reading from BCR19, bit 0.

Note: *The EEDO pin is multiplexed with the $\overline{LED3}$, MIIRXFRTGD, and SRD pins.*

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

EESK

EEPROM Serial clock **Input/Output**

This pin is designed to directly interface to a serial EEPROM that uses the 93C46 EEPROM interface protocol. EESK is connected to the EEPROM's clock pin. It is controlled by either the Am79C971 controller directly during a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 1.

Note: *The EESK pin is multiplexed with the $\overline{LED1}$ and SFBD pins.*

The EESK pin is also used during EEPROM Auto-Detection to determine whether or not an EEPROM is present at the Am79C971 controller interface. At the rising edge of the last CLK edge while $\overline{RST}$ 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 1. A sampled LOW value means that an EEPROM is not present, and EEDET will be set to 0. See the *EEPROM Auto-Detection* section 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 to resolve the EEDET setting.

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

WARNING: The input signal level of EESK must be valid for correct EEPROM detection before the deassertion of $\overline{RST}$.

Expansion Bus Interface

EBUA_EBA[7:0]

Expansion Bus Upper Address/
Expansion Bus Address [7:0] **Output**

The EBUA_EBA[7:0] pins provide the least and most significant bytes of address on the Expansion Bus. The most significant address byte (address bits [15:8] during SRAM accesses; address bits [19:16] during boot device accesses) is valid on these pins at the beginning of an SRAM or boot device access, at the rising edge of AS_$\overline{EBOE}$. This upper address byte must be stored externally in a D flip-flop. During subsequent cycles of an SRAM or boot device access, address bits [7:0] are present on these pins.

All EBUA_EBA[7:0] outputs are forced to a constant level to conserve power while no access on the Expansion Bus is being performed.

EBDA[15:8]

Expansion Bus Data/Address [15:8] Input/Output

When $\overline{\text{ERAMCS}}$ is asserted, EBDA[15:8] contain the data bits [15:8] for SRAM accesses. When $\overline{\text{EROMCS}}$ is asserted low, EBDA[15:8] contain address bits [15:8] for boot device accesses.

The EBDA[15:8] signals are driven to a constant level to conserve power while no access on the Expansion Bus is being performed.

$\overline{\text{EBD}}$[7:0]

Expansion Bus Data [7:0] Input/Output

The EBD[7:0] pins provide data bits [7:0] for RAM/ROM accesses. The EBD[7:0] signals are internally forced to a constant level to conserve power while no access on the Expansion Bus is being performed.

$\overline{\text{EROMCS}}$

Expansion ROM Chip Select Output

$\overline{\text{EROMCS}}$ serves as the chip select for the boot device. It is asserted low during the data phases of boot device accesses.

$\overline{\text{ERAMCS}}$

Expansion RAM Chip Select Output

$\overline{\text{ERAMCS}}$ is asserted during SRAM read and write operations on the expansion bus.

AS_$\overline{\text{EBOE}}$

Address Strobe/Expansion Bus Output Enable Output

AS_$\overline{\text{EBOE}}$ functions as the address strobe for the upper address bits on the EBUA_EBA[7:0] pins and as the output enable for the Expansion Bus.

As an address strobe, a rising edge on AS_$\overline{\text{EBOE}}$ is supplied at the beginning of SRAM and boot device accesses. This rising edge provides a clock edge for a ‘374 D-type edge-triggered flip-flop which must store the upper address byte during Expansion Bus accesses for EPROM/Flash/SRAM.

AS_$\overline{\text{EBOE}}$ is asserted active LOW during boot device and SRAM read operations on the expansion bus and is deasserted during boot device and SRAM write operations.

$\overline{\text{EBWE}}$

Expansion Bus Write Enable Output

$\overline{\text{EBWE}}$ provides the write enable for write accesses to the SRAM devices and/or Flash device.

EBCLK

Expansion Bus Clock Input

EBCLK may be used as the fundamental clock to drive the Expansion Bus access cycles. The actual internal clock used to drive the Expansion Bus cycles depends on the values of the EBCS and CLK_FAC settings in BCR27. Refer to the SRAM Interface Bandwidth Requirements section for details on determining the required EBCLK frequency. If a clock source other than the EBCLK pin is programmed (BCR27, bits 5:3) to be used to run the Expansion Bus interface, this input should be tied to VDD through a 4.7 kΩ resistor.

EBCLK is not used to drive the bus interface, internal buffer management unit, or the network functions.

Media Independent Interface

TX_CLK

Transmit Clock Input

TX_CLK is a continuous clock input that provides the timing reference for the transfer of the TX_EN, TXD[3:0], and TX_ER signals out of the Am79C971 device. TX_CLK must provide a nibble rate clock (25% of the network data rate). Hence, an MII transceiver operating at 10 Mbps must provide a TX_CLK frequency of 2.5 MHz and an MII transceiver operating at 100 Mbps must provide a TX_CLK frequency of 25 MHz.

Note: *The TX_CLK pin is multiplexed with the TXCLK pin.*

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

If the MII port is not selected, the TX_CLK pin can be left floating.

TXD[3:0]

Transmit Data Output

TXD[3:0] is the nibble-wide MII transmit data bus. Valid data is generated on TXD[3:0] on every TX_CLK rising edge while TX_EN is asserted. While TX_EN is deasserted, TXD[3:0] values are driven to a 0. TXD[3:0] transitions synchronous to TX_CLK rising edges.

Note: *The TXD[0] pin is multiplexed with the TXDAT pin.*

When $\overline{\text{RST}}$ is active, TXD[3:0] are inputs for NAND tree testing.

If the MII port is not selected, the TXD[3:0] pins can be left floating.

TX_EN

Transmit Enable Output

TX_EN indicates when the Am79C971 device is presenting valid transmit nibbles on the MII. While TX_EN is asserted, the Am79C971 device generates TXD[3:0] and TX_ER on TX_CLK rising edges. TX_EN is asserted with the first nibble of preamble and remains asserted throughout the duration of a packet until it is deasserted prior to the first TX_CLK following the final

nibble of the frame. TX_EN transitions synchronous to TX_CLK rising edges.

Note: *The TX_EN pin is multiplexed with the TXEN pin.*

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

If the MII port is not selected, the TX_EN pin can be left floating.

TX_ER

Transmit Error **Output**

TX_ER is an output that, if asserted while TX_EN is asserted, instructs the MII PHY device connected to the Am79C971 device to transmit a code group error. TX_ER is unused and is reserved for future use and will always be driven to a logical zero.

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

If the MII port is not selected, the TX_ER pin can be left floating.

COL

Collision **Input**

COL is an input that indicates that a collision has been detected on the network medium.

Note: *The COL pin is multiplexed with the CLSN pin.*

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

If the MII port is not selected, the COL pin can be left floating.

CRS

Carrier Sense **Input**

CRS is an input that indicates that a non-idle medium, due either to transmit or receive activity, has been detected.

Note: *The CRS pin is multiplexed with the RXEN pin.*

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

If the MII port is not selected, the CRS pin can be left floating.

RX_CLK

Receive Clock **Input**

RX_CLK is a clock input that provides the timing reference for the transfer of the RX_DV, RXD[3:0], and RX_ER signals into the Am79C971 device. RX_CLK must provide a nibble rate clock (25% of the network data rate). Hence, an MII transceiver operating at 10 Mbps must provide an RX_CLK frequency of 2.5 MHz and an MII transceiver operating at 100 Mbps must provide an RX_CLK frequency of 25 MHz. When the external PHY switches the RX_CLK and TX_CLK, it ***must*** provide glitch-free clock pulses.

Note: *The RX_CLK pin is multiplexed with the RXCLK pin.*

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

If the MII port is not selected, the RX_CLK pin can be left floating.

RXD[3:0]

Receive Data **Input**

RXD[3:0] is the nibble-wide MII receive data bus. Data on RXD[3:0] is sampled on every rising edge of RX_CLK while RX_DV is asserted. RXD[3:0] is ignored while RX_DV is de-asserted.

When the EADI is enabled (EADISEL, BCR2, bit 3) and the Receive Frame Tagging is enabled (RXFRTG, CSR7, bit 14) and the MII is *not* selected, the RXD[0] pin becomes a data input pin for the Receive Frame Tag (RXFRTGD). See the *Receive Frame Tagging* section for details.

Note: *The RXD[0] pin is multiplexed with the RXFRTGD pin.*

When $\overline{RST}$ is active, RXD[3:0] are inputs for NAND tree testing.

If the MII port is not selected, the RXS[3:0] pin can be left floating.

RX_DV

Receive Data Valid **Input**

RX_DV is an input used to indicate that valid received data is being presented on the RXD[3:0] pins and RX_CLK is synchronous to the receive data. In order for a frame to be fully received by the Am79C971 device on the MII, RX_DV must be asserted prior to the RX_CLK rising edge, when the first nibble of the Start of Frame Delimiter is driven on RXD[3:0], and must remain asserted until after the rising edge of RX_CLK, when the last nibble of the CRC is driven on RXD[3:0]. RX_DV must then be deasserted prior to the RX_CLK rising edge which follows this final nibble. RX_DV transitions are synchronous to RX_CLK rising edges.

When the EADI is enabled (EADISEL, BCR2, bit 3), the Receive Frame Tagging is enabled (RXFRTG, CSR7, bit 14), and the MII is *not* selected, the RX_DV pin becomes a data input enable pin for the Receive Frame Tag (RXFRTGE). See the *Receive Frame Tagging* section for details.

Note: *The RX_DV pin is multiplexed with the RXFRTGE pin.*

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

If the MII port is not selected, the RX_DV pin can be left floating.

RX_ER

Receive Error **Input**

RX_ER is an input that indicates that the MII transceiver device has detected a coding error in the receive frame currently being transferred on the RXD[3:0] pins. When RX_ER is asserted while RX_DV is asserted, a CRC error will be indicated in the receive descriptor for the incoming receive frame. RX_ER is ignored while RX_DV is deasserted. Special code groups generated on RXD while RX_DV is deasserted are ignored (e.g., Bad SSD in TX and IDLE in T4). RX_ER transitions are synchronous to RX_CLK rising edges.

Note: *The RX_ER pin is multiplexed with the RXDAT pin.*

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

If the MII port is not selected, the RX_ER pin can be left floating.

MDC

Management Data Clock **Output**

MDC is a non-continuous clock output that provides a timing reference for bits on the MDIO pin. During MII management port operations, MDC runs at a nominal frequency of 2.5 MHz. When no management operations are in progress, MDC is driven LOW. The MDC is derived from the external 20-MHz crystal.

If the MII port is not selected, the MDC pin can be left floating.

MDIO

Management Data I/O **Input/Output**

MDIO is the bidirectional MII management port data pin. MDIO is an output during the header portion of the management frame transfers and during the data portions of write transfers. MDIO is an input during the data portions of read data transfers. When an operation is not in progress on the management port, MDIO is not driven. MDIO transitions from the Am79C971 controller are synchronous to MDC Falling edges.

If the PHY is attached through an MII physical connector, then the MDIO pin should be externally pulled down to V_{SS} with a 10-kΩ ±5% resistor. If the PHY is on board, then the MDIO pin should be externally pulled up to V_{CC} with a 10-kΩ ±5% resistor.

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

Attachment Unit Interface

CI±

Collision In **Input**

CI± is a differential input pair signaling the Am79C971 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. CI± operates at pseudo ECL levels.

If the CI± pins are not used, they should be tied together.

DI±

Data In **Input**

DI± is a differential input pair to the Am79C971 controller carrying Manchester encoded data from the network. DI± operates at pseudo ECL levels.

If the DI± pins are not used, they should be tied together.

DO±

Data Out **Output**

DO± is a differential output pair from the Am79C971 controller for transmitting Manchester encoded data to the network. DO± operates at pseudo ECL levels.

If the AUI is not used, DO± should be left floating for minimum power consumption.

10BASE-T Interface

RXD±

10BASE-T Receive Data **Input**

RXD± are 10BASE-T port differential receivers. If the 10BASE-T interface is not used in a design, RXD+ and RXD- should be connected to each other.

TXD±

10BASE-T Transmit Data **Output**

TXD± are 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.

General Purpose Serial Interface

CLSN

Collision **Input**

CLSN is an input that indicates a collision has occurred on the network.

Note: *The CLSN pin is multiplexed with the COL pin.*

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

RXCLK

Receive Clock **Input**

RXCLK is an input. The rising edges of the RXCLK signal are used to sample the data on the RXDAT input whenever the RXEN input is HIGH.

Note: *The RXCLK pin is multiplexed with the RX_CLK pin.*

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

RXDAT

Receive Data **Input**

RXDAT is an input. The rising edges of the RXCLK signal are used to sample the data on the RXDAT input whenever the RXEN input is HIGH.

Note: *The RXDAT pin is multiplexed with the RX_ER pin.*

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

RXEN

Receive Enable **Input**

RXEN is an input. When this signal is HIGH, it indicates to the core logic that the data on the RXDAT input pin is valid.

Note: *The RXEN pin is multiplexed with the CRS pin.*

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

TXCLK

Transmit Clock **Input**

TXCLK is an input that provides a clock signal for MAC activity, both transmit and receive. The rising edges of the TXCLK can be used to validate TXDAT output data.

Note: *The TXCLK pin is multiplexed with the TX_CLK pin.*

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

TXDAT

Transmit Data **Output**

TXDAT is an output that provides the serial bit stream for transmission, including preamble, SFD, data, and FCS field, if applicable.

Note: *The TXDAT pin is multiplexed with the TXD[0] pin.*

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

TXEN

Transmit Enable **Output**

TXEN is an output that provides an enable signal for transmission. Data on the TXDAT pin is not valid unless the TXEN signal is HIGH.

Note: *The TXEN pin is multiplexed with the TX_EN pin.*

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

External Address Detection Interface

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 OR'd with the value on the $\overline{EAR}$ pin. The $\overline{EAR}$ pin is defined as $\overline{REJECT}$. The pin value is OR'd with the internal address detection result to determine if the current frame should be accepted or rejected.

The $\overline{EAR}$ pin ***must not*** be left unconnected, it should be tied to VDD through a resistor.

Note: *The $\overline{EAR}$ pin is multiplexed with the $\overline{SLEEP}$ pin.*

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

SFBD

Start Frame-Byte Delimiter **Output**

For the Internal PHY during External Address Detection:

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 when operating at 10 Mbps) after detecting the second "1" in the SFD (Start of Frame Delimiter) of a received frame. SFBD will subsequently toggle every 4 bit times (1.25 MHz frequency when operating at 10 Mbps) with each rising edge indicating the first bit of each subsequent byte of the received serial bit stream. See the *EADI Rejection Timing with Internal PHY* timing diagram for details. SFBD will be active only during frame reception.

For the External PHY attached to the Media Independent Interface during External Address Detection:

An initial rising edge on the SFBD signal indicates that a start of valid data is present on the RXD[3:0] pins. SFBD will go high for one nibble time (400 ns when operating at 10 Mbps and 40 ns when operating at 100 Mbps) one RX_CLK period after RX_DV has been asserted and RX_ER is deasserted and the detection of

the SFD (Start of Frame Delimiter) of a received frame. Data on the RXD[3:0] will be the start of the destination address field. SFBD will subsequently toggle every nibble time (1.25 MHz frequency when operating at 10 Mbps and 12.5 MHz frequency when operating at 100 Mbps) indicating the first nibble of each subsequent byte of the received nibble stream. The RX_CLK should be used in conjunction with the SFBD to latch the correct data for external address matching. SFBD will be active only during frame reception.

Note: *The SFBD pin is multiplexed with the EESK and $\overline{LED1}$ pins.*

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

SRD

Serial Receive Data **Input/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.

When the EADI is enabled (EADISEL, BCR2, bit 3) and the Receive Frame Tagging is enabled (RXFRTG, CSR7, bit 14) and the MII is selected, the SRD pin becomes a data input pin for the Receive Frame Tag (MIIRXFRTGD). See the *Receive Frame Tagging* section for details.

Note: *When the MII port is selected, SRD will not generate transitions and receive data must be derived from the Media Independent Interface RXD[3:0] pins.*

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

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

SRDCLK

Serial Receive Data Clock **Input/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.

When the EADI is enabled (EADISEL, BCR2, bit 3), the Receive Frame Tagging is enabled (RXFRTG, CSR7, bit 14), and the MII is selected, the SRDCLK pin becomes a data input enable pin for the Receive Frame Tag (MIIRXFRTGE). See the *Receive Frame Tagging* section for details.

Note: *When the MII port is selected, SRDCLK will not generate transitions and the receive clock must be derived from the MII RX_CLK pin.*

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

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

RXFRTGD

Receive Frame Tag Data **Input**

When the EADI is enabled (EADISEL, BCR2, bit 3), the Receive Frame Tagging is enabled (RXFRTG, CSR7, bit 14), and the MII is not selected, the RXFRTGD pin becomes a data input pin for the Receive Frame Tag. See the *Receive Frame Tagging* section for details.

Note: *The RXFRTGD pin is multiplexed with the RXD[0] pin.*

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

RXFRTGE

Receive Frame Tag Enable **Input**

When the EADI is enabled (EADISEL, BCR2, bit 3), the Receive Frame Tagging is enabled (RXFRTG, CSR7, bit 14), and the MII is not selected, the RXFRTGE pin becomes a data input enable pin for the Receive Frame Tag. See the *Receive Frame Tagging* section for details.

Note: *The RXFRTGE pin is multiplexed with the RX_DV pin.*

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

MIIRXFRTGD

MII Receive Frame Tag Enable **Input/Output**

When the EADI is enabled (EADISEL, BCR2, bit 3), the Receive Frame Tagging is enabled (RXFRTG, CSR7, bit 14), and the MII is selected, the MIIRXFRTGD pin becomes a data input pin for the Receive Frame Tag. See the *Receive Frame Tagging* section for details.

Note: *The MIIRXFRTGD pin is multiplexed with the SRD pin.*

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

MIIRXFRTGE

MII Receive Frame Tag Enable **Input/Output**

When the EADI is enabled (EADISEL, BCR2, bit 3), the Receive Frame Tagging is enabled (RXFRTG, CSR7, bit 14), and the MII is selected, the MIIRXFRTGE pin becomes a data input enable pin for the Receive Frame Tag. See the *Receive Frame Tagging* section for details.

Note: *The MIIRXFRTGE pin is multiplexed with the SRDCLK pin.*

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

IEEE 1149.1 (1990) 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.

TDI

Test Data In **Input**

TDI is the test data input path to the Am79C971 controller. The pin has an internal pull up resistor.

TDO

Test Data Out **Output**

TDO is the test data output path from the Am79C971 controller. The pin is tri-stated when the JTAG port is inactive.

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.

Power Supply Pins

AVDDB

Analog Power (3 Pins) **Power**

There are three analog +5 V supply pins that provide power for the Twisted Pair and AUI drivers. Hence, they are very noisy. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to *Appendix B, Recommendation for Power and Ground Decoupling,* for details.

AVSSB

Analog Ground (1 Pins) **Power**

There is one analog ground pin that provides ground for the Twisted Pair and AUI drivers. Hence, it is very noisy. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to *Appendix B, Recommendation for Power and Ground Decoupling,* for details.

VDD_PLL

PLL Power (1 Pin) **Power**

There is one analog PLL +5 V supply pin. Special attention should be paid to the printed circuit board layout to avoid excessive noise on this line. Refer to *Appendix B, Recommendation for Power and Ground Decoupling,* for details.

VSS_PLL

PLL Ground (1 Pin) **Power**

There is one analog PLL ground pin. Special attention should be paid to the printed circuit board layout to avoid excessive noise on this line. Refer to *Appendix B, Recommendation for Power and Ground Decoupling,* for details.

VDDB

I/O Buffer Power (5 Pins) **Power**

There are five power supply pins that are used by the input/output buffer drivers. All VDDB pins must be connected to a +5 V supply.

VSSB

I/O Buffer Ground (13 Pins) **Power**

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

VDD_PCI

PCI I/O Buffer Power (5 Pins) **Power**

There are five power supply pins that are used by the PCI input/output buffer drivers. In a system with +5 V signaling environment, all VDD_PCI pins must be connected to a +5 V supply. In a system with +3.3 V signaling environment, all VDD_PCI pins must be connected to a +3.3 V supply.

VDD

Digital Power (4 Pins) **Power**

There are four power supply pins that are used by the internal digital circuitry. All VDD pins must be connected to a +5 V supply.

VSS

Digital Ground (6 Pins) **Power**

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

BASIC FUNCTIONS

System Bus Interface

The Am79C971 controller is designed to operate as a bus master during normal operations. Some slave I/O accesses to the Am79C971 controller are required in normal operations as well. Initialization of the Am79C971 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 Am79C971 controller. The EEPROM read operation is performed through the 93C46 EEPROM interface. The ISO 8802-3 (IEEE/ANSI 802.3) Ethernet Address may reside within the serial EEPROM. Some Am79C971 controller configuration registers may also be programmed by the EEPROM read operation.

The Address PROM, on-chip board-configuration registers, and the Ethernet controller registers occupy 32 bytes of address space. I/O and memory mapped I/O accesses 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 Am79C971 controller supports a ROM or Flash-based (both referred to as the *Expansion ROM* throughout this specification) boot device of up to 1 Mbyte in size. The host can map the boot device to any memory address that aligns to a 1-Mbyte boundary by modifying the Expansion ROM Base Address register in the PCI configuration space.

Software Interface

The software interface to the Am79C971 controller is divided into three parts. One part is the PCI configuration registers used to identify the Am79C971 controller and 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 Am79C971 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 Am79C971 controller. The Am79C971 controller occupies 32 bytes of address space that must begin on a 32-byte block boundary. The address space can be mapped into I/O or memory space (memory mapped I/O). The I/O Base Address Register in the PCI Configuration Space controls the start address of the address space if it is mapped to I/O space. The Memory Mapped I/O Base Address Register controls 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 Am79C971 controller operating mode, to enable and disable various features, to monitor operating status, and to request particular functions to be executed by the Am79C971 controller.

The third portion of the software interface is the descriptor and buffer areas that are shared between the software and the Am79C971 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 Am79C971 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 Am79C971 controller can be connected to an IEEE 802.3 or proprietary network via one of four network interfaces. The Media Independent Interface (MII) provides an IEEE 802.3-compliant nibble-wide interface to an external 100- and/or 10-Mbps transceiver device. The Attachment Unit Interface (AUI) provides an ISO 8802-3 (IEEE/ANSI 802.3) defined differential interface. On-board MAU and or off-board MAU connection with or without an AUI cable is supported. The 10BASE-T interface provides a twisted-pair Ethernet port, which is ISO 8802-3 (IEEE/ANSI 802.3)-compliant, and contains the auto-negotiation capability, which is IEEE 802.3u-compliant. The General Purpose Serial Interface (GPSI) allows bypassing the Manchester Encoder/Decoder (MENDEC) and is functionally equivalent to the GPSI found on the LANCE.

While in auto-selection mode, the interface in use is determined by the Network Port Manager. If the quiescent state of the MII MDIO pin is HIGH, the MII is activated. If the MII MDIO pin is LOW, the Am79C971 device checks the link status on the 10BASE-T port. If the 10BASE-T link status is good, the 10BASE-T port is selected. If there is no active link status, then the device assumes an AUI connection. The 10BASE-T port will continue to monitor the link status while the AUI is active. The software driver can override this automatic configuration at anytime by disabling the auto-selection and forcing a network port to be attached to the internal MAC. The GPSI port can only be enabled by disabling the auto-selection and manually selecting the GPSI as the network port.

The Am79C971 controller supports half-duplex and full-duplex operation on all four network interfaces (i.e., AUI, 10BASE-T, GPSI, and MII).

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. Table 2 shows the response of the Am79C971 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, and Reset registers
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, and Reset registers Read of the Expansion Bus
0111	Memory Write	Memory mapped I/O write of CSR, BCR, and APROM
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 Am79C971 PCI configuration space with a configuration read or write command. The Am79C971 controller will assert DEVSEL during the address phase when IDSEL is asserted, AD[1:0] are both 0, and the access is a configuration cycle. AD[7:2] select the DWord location in the configuration space. The Am79C971 controller ignores AD[10:8], because it is a single function device. AD[31:11] are don't care.

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

The active bytes within a DWord are determined by the byte enable signals. Eight-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 Am79C971 controller will assert TRDY on the third clock of the data phase.

The Am79C971 controller does not support burst transfers for access to configuration space. When the host keeps FRAME asserted for a second data phase, the Am79C971 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 (see section on RESET) is on-going, the Am79C971 controller will terminate the access on the PCI bus with a disconnect/retry response.

The Am79C971 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 1. The Am79C971 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 Am79C971 controller asserts DEVSEL on the second clock after FRAME is asserted (medium timing).

Slave I/O Transfers

After the Am79C971 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 APROM locations. If configured for regular I/O mode, the Am79C971 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 Am79C971 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 Am79C971 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 Am79C971 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. See Figure 1 and Figure 2.

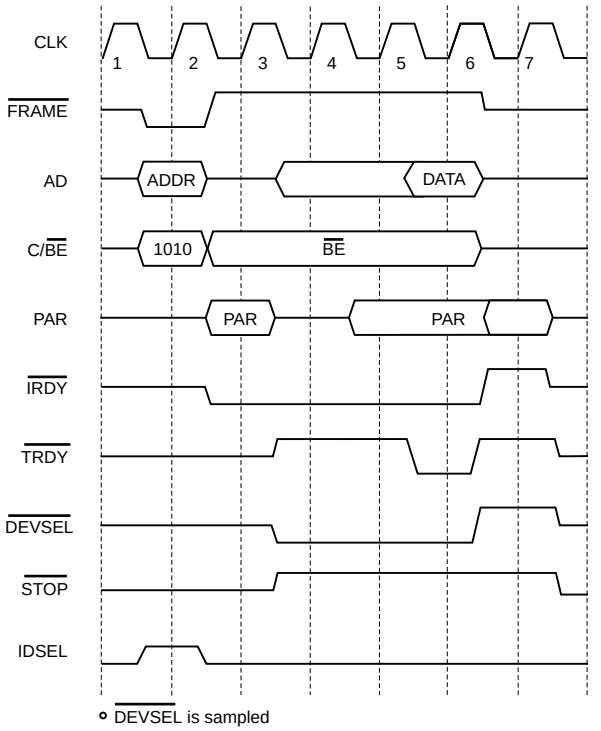


Figure 1. Slave Configuration Read

The Am79C971 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 Am79C971 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. Eight-bit, 16-bit, and 32-bit non-burst transactions are supported. The Am79C971 controller decodes all 32 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 Am79C971 controller is six to seven 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 Am79C971 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 Am79C971 controller will disconnect the transfer.

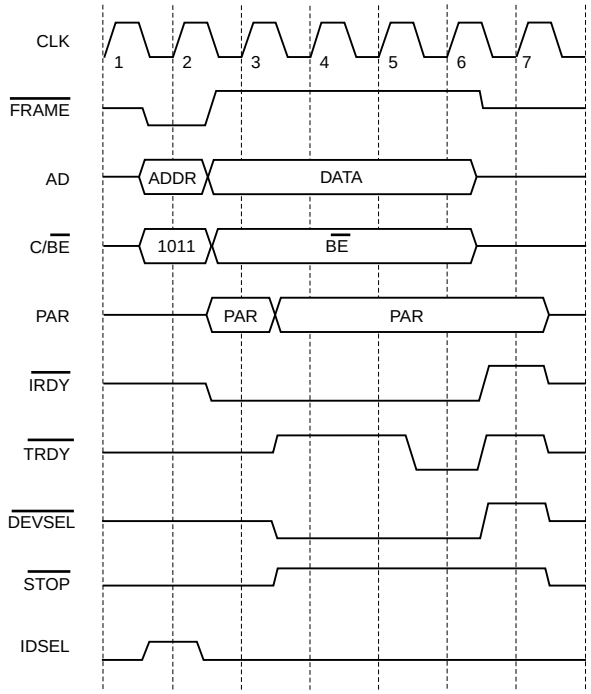
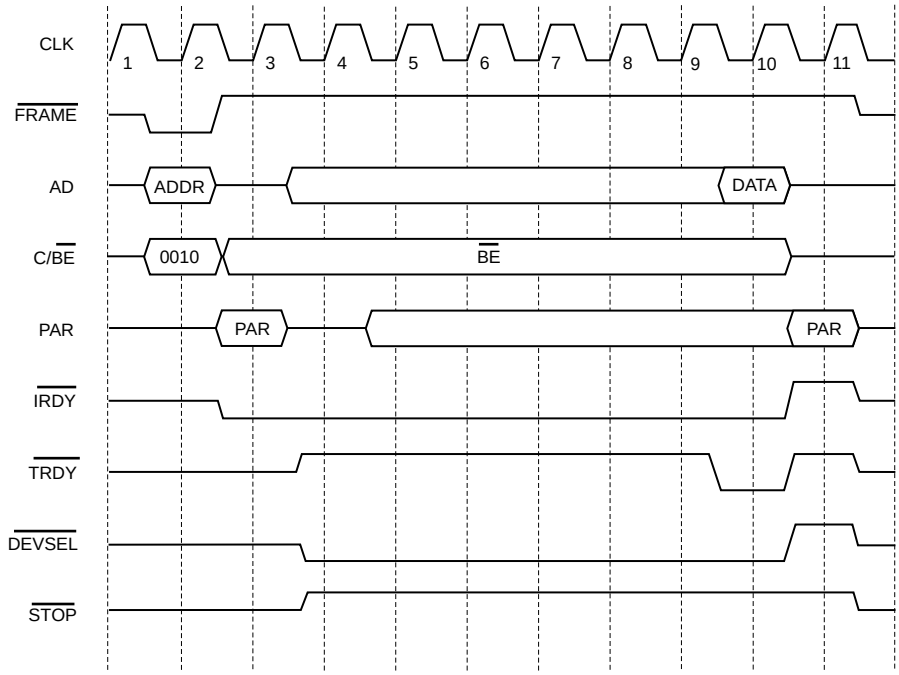


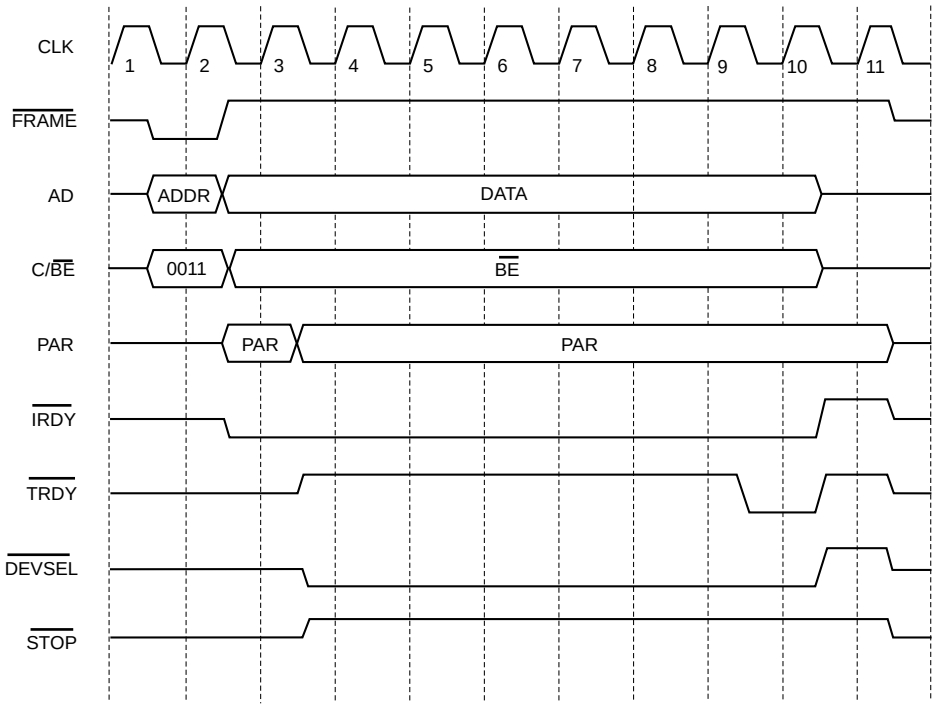
Figure 2. Slave Configuration Write

The Am79C971 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 1. The Am79C971 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 DEVSEL, TRDY, and STOP signals, since the Am79C971 controller asserts DEVSEL on the second clock after FRAME is asserted (medium timing) See Figure 3 and Figure 4.



20550D-6

Figure 3. Slave Read Using I/O Command



20550D-7

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 Am79C971 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 1. After the Expansion ROM is enabled, the Am79C971 controller will assert DEVSEL on all memory read accesses with an address between ROMBASE and ROMBASE + 1M - 4. The Am79C971 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 Am79C971 controller from claiming any memory cycles not intended for it.

The Am79C971 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 (Figure 5) assumes that ROMTMG (BCR18, bits 15-12) is at its default value.

Note: *The Expansion ROM should be read only during PCI configuration time for the PCI system.*

When the host tries to write to the Expansion ROM, the Am79C971 controller will claim the cycle by asserting DEVSEL. TRDY will be asserted one clock cycle later. The write operation will have no effect. Writes to the Expansion ROM are done through the BCR30 Expansion Bus Data Port. See the section on the *Expansion Bus Interface* for more details.

The Am79C971 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 1. The Am79C971 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 DEVSEL, TRDY, and STOP signals, since the Am79C971 controller asserts DEVSEL on the second clock after FRAME is asserted (medium timing). See Figure 5.

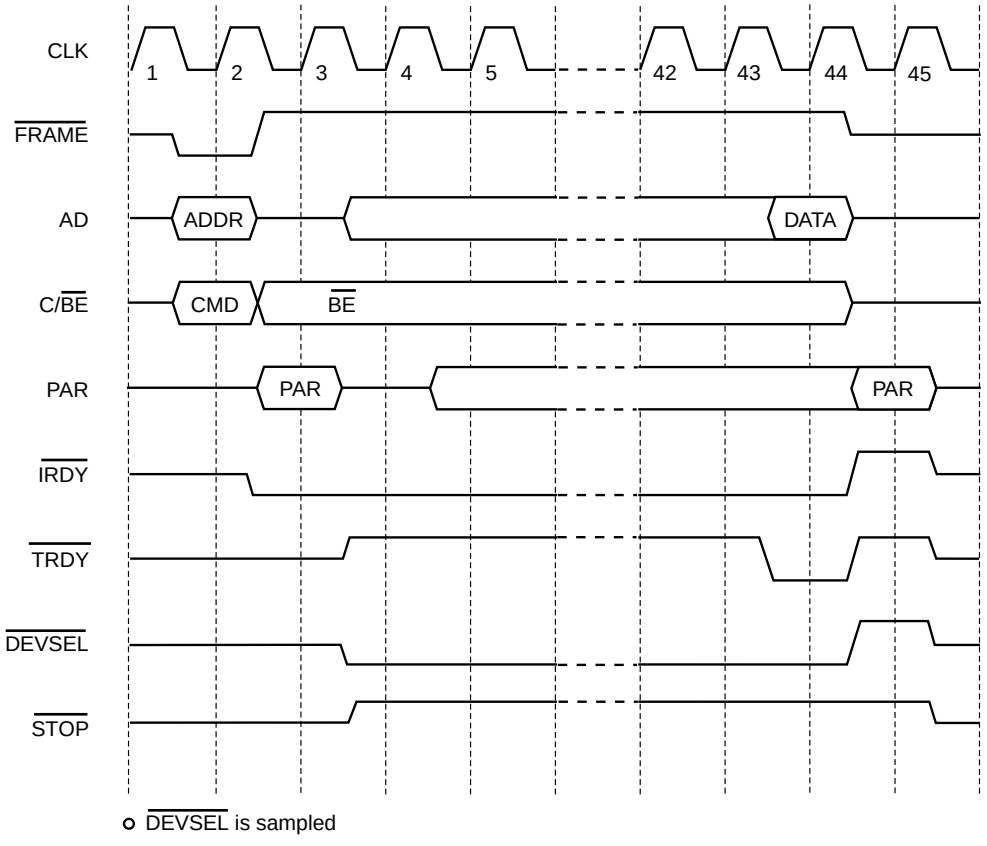


Figure 5. Expansion ROM Read

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 EBD pins together.

Slave Cycle Termination

There are three scenarios besides normal completion of a transaction where the Am79C971controller is the target of a slave cycle and it will terminate the access.

Disconnect When Busy

The Am79C971controller cannot service any slave access while it is reading the contents of the EEPROM. Simultaneous access is not possible to avoid conflicts, since the EEPROM is used to initialize some of the PCI configuration space locations and most of the BCRs. The EEPROM read operation will always happen automatically after the deassertion of the $\overline{RST}$ 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 Am79C971controller will disconnect any slave access where it is the target by asserting $\overline{STOP}$ together with $\overline{DEVSEL}$, while driving $\overline{TRDY}$ high. $\overline{STOP}$ will stay asserted until the end of the cycle.

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 Am79C971controller 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. See Figure 6.

Disconnect Of Burst Transfer

The Am79C971controller does not support burst access to the configuration space, the I/O resources, or to the Expansion Bus. The host indicates a burst transaction by keeping $\overline{FRAME}$ asserted during the data phase. When the Am79C971controller sees $\overline{FRAME}$ and $\overline{IRDY}$ asserted in the clock cycle before it wants to assert $\overline{TRDY}$, it also asserts $\overline{STOP}$ at the same time. The transfer of the first data phase is still successful, since $\overline{IRDY}$ and $\overline{TRDY}$ are both asserted. See Figure 7.

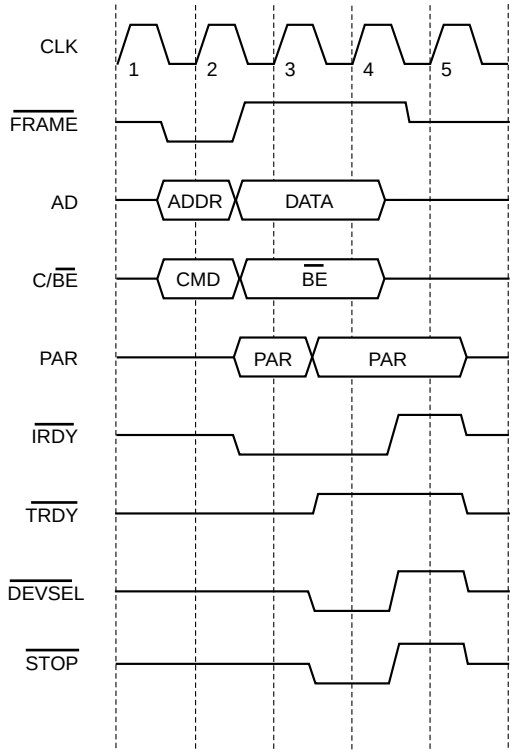


Figure 6. Disconnect Of Slave Cycle When Busy

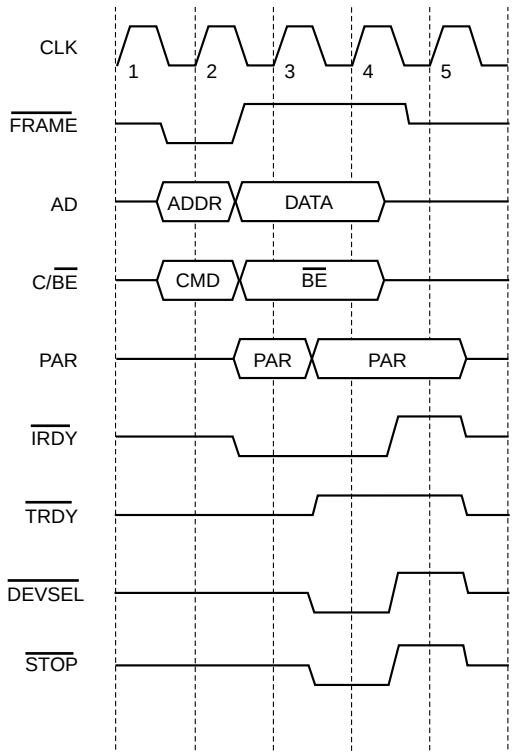
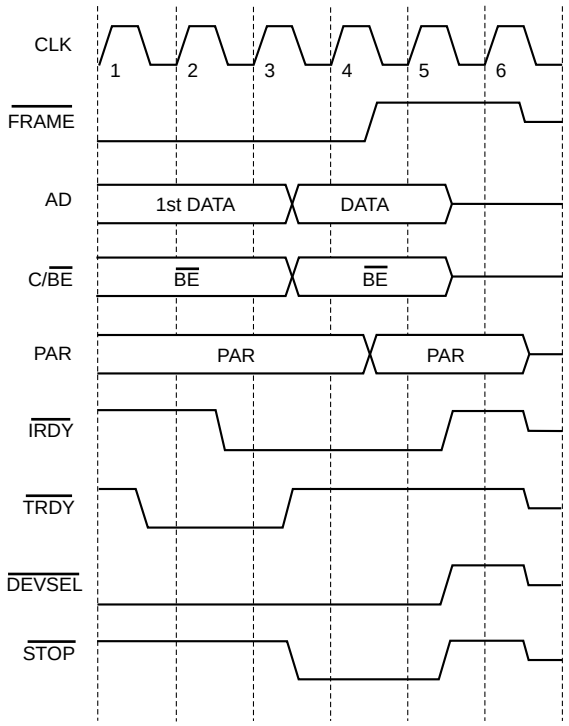


Figure 7. Disconnect Of Slave Burst Transfer - No Host Wait States

If the host is not yet ready when the Am79C971 controller asserts TRDY, the device will wait for the host to assert IRDY. When the host asserts IRDY and FRAME is still asserted, the Am79C971controller 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. See Figure 8.

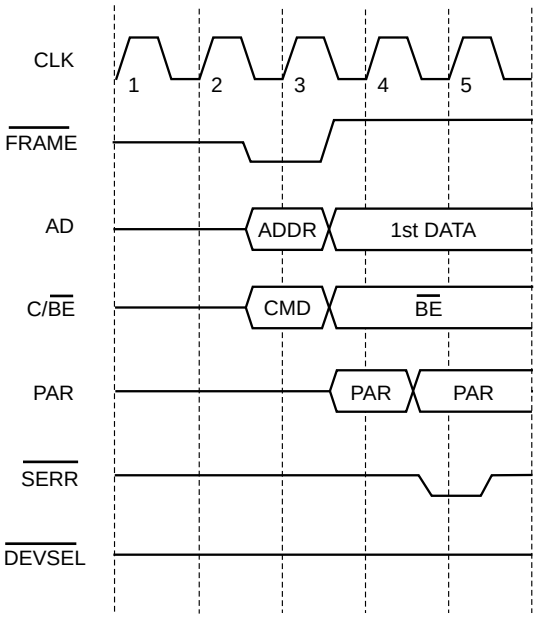


20550D-11

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

Parity Error Response

When the Am79C971controller is not the current bus master, it samples the AD[31:0], C/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 1. When reporting of that error is enabled by setting SERREN (PCI Command register, bit 8) and PERREN (PCI Command register, bit 6) to 1, the Am79C971controller also drives the SERR signal low for one clock cycle and sets SERR (PCI Status register, bit 14) to 1. The assertion of SERR follows the address phase by two clock cycles. The Am79C971controller will not assert DEVSEL for a PCI transaction that has an address parity error when PERREN and SERREN are set to 1. See Figure 9.



20550D-12

Figure 9. 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 Am79C971controller as target, the device samples the AD[31:0] and C/BE[3:0] lines for parity on the clock edge, and data is transferred as indicated by the assertion of IRDY and TRDY. 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 1, PERR is asserted one clock later. The parity error will always set PERR (PCI Status register, bit 15) set to 1 even when PERREN is cleared to 0. The Am79C971controller will finish a transaction that has a data parity error in the normal way by asserting TRDY. The corrupted data will be written to the addressed location.

Figure 10 shows a transaction that suffered a parity error at the time data was transferred (clock 7, IRDY and TRDY are both asserted). 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 PERR s driven low, the Am79C971controller drives PERR high for one clock cycle, since PERR is a sustained tri-state signal.

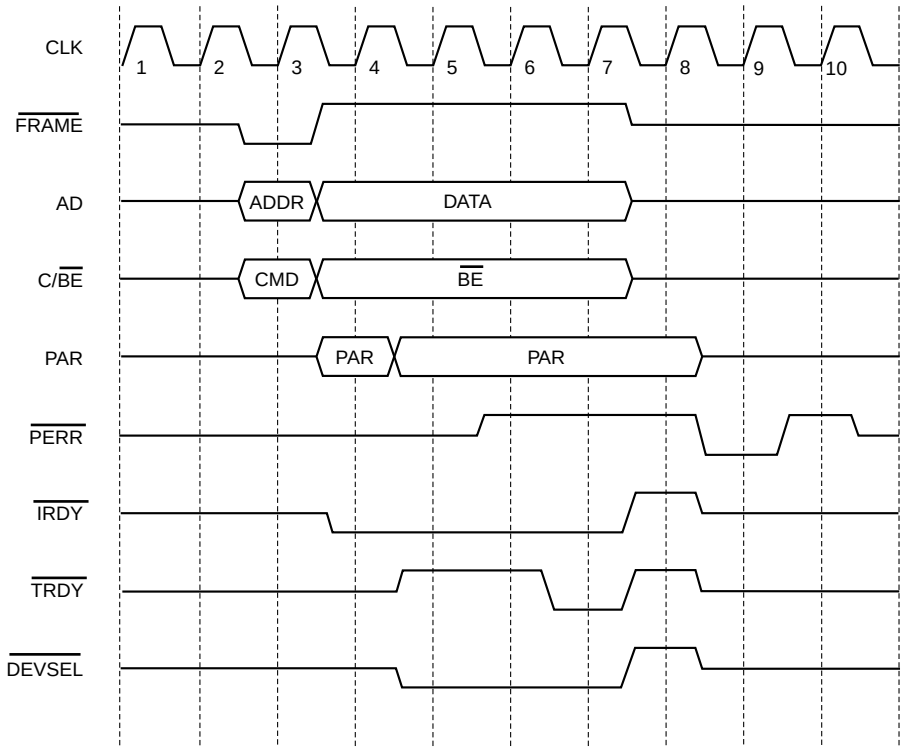


Figure 10. Slave Cycle Data Parity Error Response

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. Table 3 shows the usage of PCI commands by the Am79C971controller 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 Am79C971microcode will determine when a DMA transfer should be initiated. The first step in any Am79C971bus 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 $\overline{REQ}$ signal and ownership is granted by the arbiter through the $\overline{GNT}$ signal.

Figure 11 shows the Am79C971controller bus acquisition. REQ is asserted and the arbiter returns GNT while another bus master is transferring data. The Am79C971 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 Am79C971 controller does not use address stepping which is reflected by ADSTEP (bit 7) in the PCI Command register being hardwired to 0.

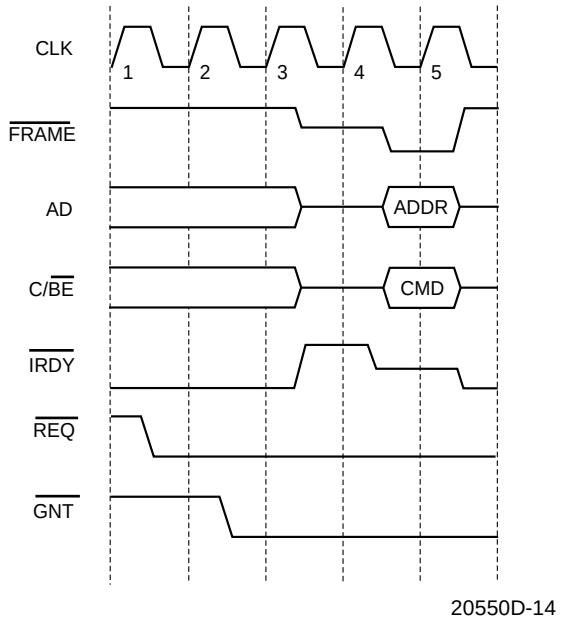


Figure 11. Bus Acquisition

In burst mode, the deassertion of REQ depends on the setting of EXTREQ (BCR18, bit 8). If EXTREQ is cleared to 0, REQ is deasserted at the same time as FRAME is asserted. (The Am79C971 controller never performs more than one burst transaction within a single bus mastership period.) If EXTREQ is set to 1, the Am79C971 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 and independent of subsequent setting of 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.

Bus Master DMA Transfers

There are four primary types of DMA transfers. The Am79C971 controller uses non-burst as well as burst cycles for read and write access to the main memory.

Basic Non-Burst Read Transfer

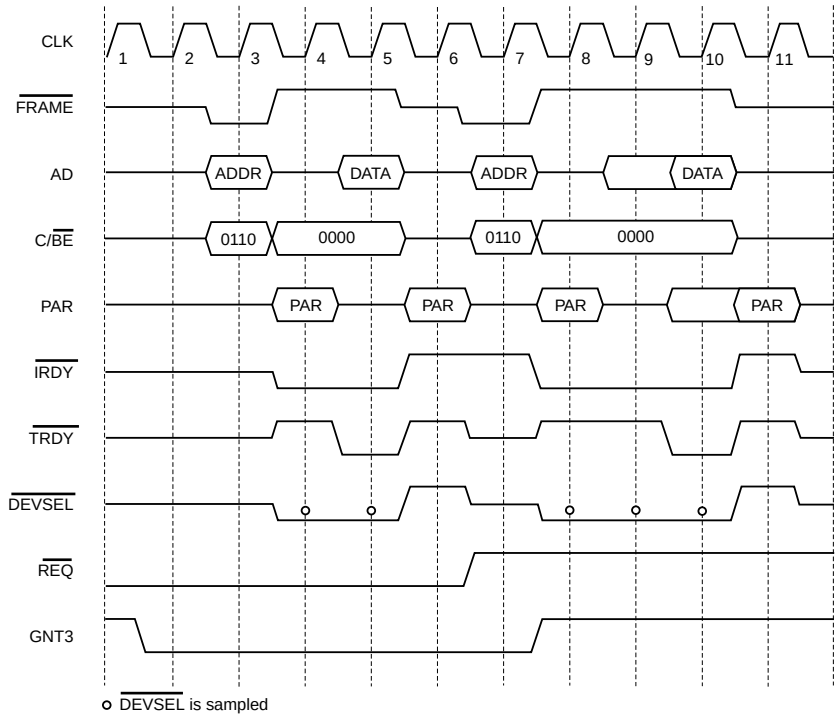
By default, the Am79C971 controller uses non-burst cycles in all bus master read operations. All Am79C971 controller non-burst read accesses are of the PCI command type Memory Read (type 6). Note that during a non-burst read operation, all byte lanes will always be active. The Am79C971 controller will internally discard unneeded bytes.

The Am79C971 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 Am79C971 controller supports zero wait state read cycles. It asserts IRDY immediately after the address phase and at the same time starts sampling DEVSEL. Figure 12 shows two non-burst read transactions. The first transaction has zero wait states. In the second transaction, the target extends the cycle by asserting TRDY one clock later.

Basic Burst Read Transfer

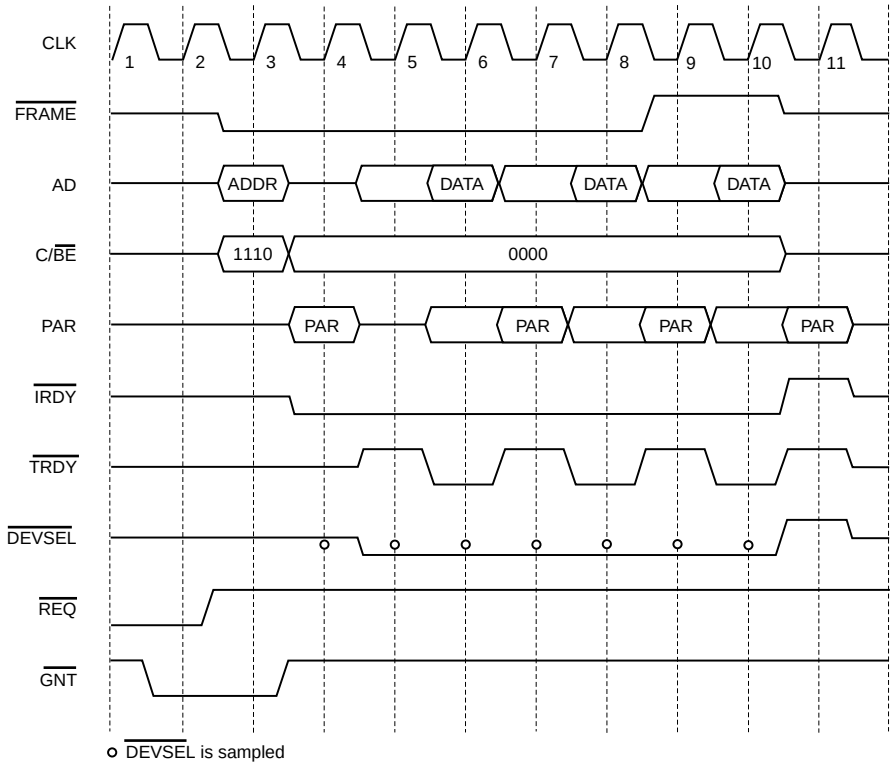
The Am79C971 controller supports burst mode for all bus master read operations. The burst mode must be enabled by setting BREADE (BCR18, bit 6). To allow burst transfers in descriptor read operations, the Am79C971 controller must also be programmed to use SWSTYLE 3 (BCR20, bits 7-0). All burst read accesses 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 0, 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 to1, 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 0 during the address phase indicating a linear burst order. Note that during a burst read operation, all byte lanes will always be active. The Am79C971 controller will internally discard unneeded bytes.

The Am79C971 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 Am79C971 controller supports zero wait state read cycles. It asserts IRDY immediately after the address phase and at the same time starts sampling DEVSEL. FRAME is deasserted when the next to last data phase is completed. Figure 13 shows a typical burst read access. The Am79C971 controller arbitrates for the bus, is granted access, 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 0, therefore, REQ is deasserted in the same cycle as FRAME is asserted.



20550D-15

Figure 12. Non-Burst Read Transfer



20550D-16

Figure 13. Burst Read Transfer (EXTREQ = 0, MEMCMD = 0)

Basic Non-Burst Write Transfer

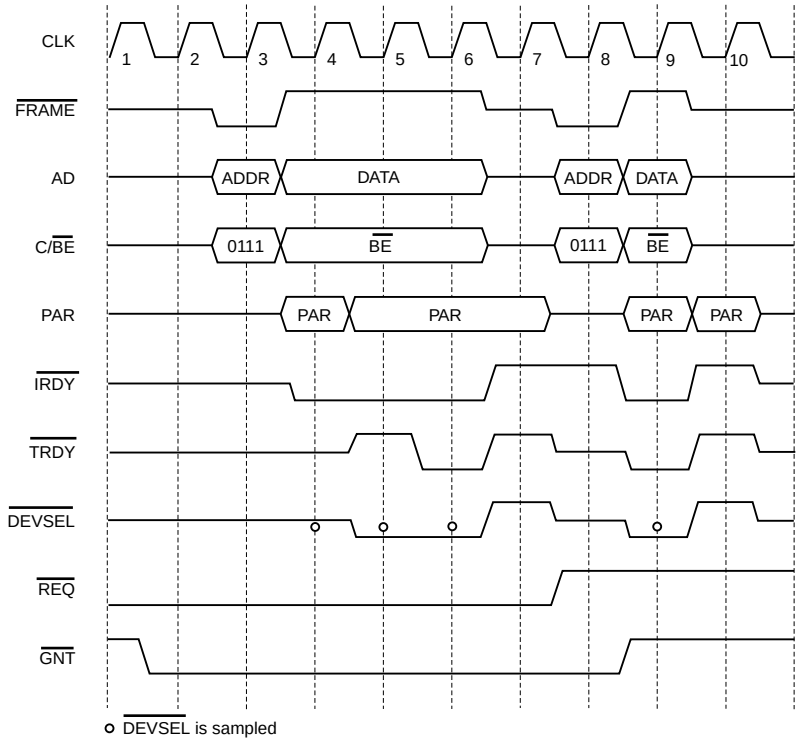
By default, the Am79C971 controller uses non-burst cycles in all bus master write operations. All Am79C971 controller non-burst write accesses are of the PCI command type Memory Write (type 7). The byte enable signals indicate the byte lanes that have valid data.The Am79C971 controller typically performs more than one non-burst write transaction within a single bus mastership period. FRAME is dropped between consecutive non-burst write cycles. REQ, however, stays asserted until FRAME is asserted for the last transaction. The Am79C971 supports zero wait state write cycles except with descriptor write transfers. (See the section *Descriptor DMA Transfers* for the only exception.) It asserts IRDY immediately after the address phase.

Figure 14 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 Am79C971 controller asserts IRDY.

Basic Burst Write Transfer

The Am79C971 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 Am79C971 controller must also be programmed to use SWSTYLE 3 (BCR20, bits 7-0). All Am79C971 controller burst write transfers are of the PCI command type Memory Write (type 7). AD[1:0] will both be 0 during the address phase indicating a linear burst order. The byte enable signals indicate the byte lanes that have valid data.

The Am79C971 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 Am79C971 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.) The device asserts IRDY immediately after the address phase and at the same time starts sampling DEVSEL. FRAME is deasserted when the next to last data phase is completed.



20550D-17

Figure 14. Non-Burst Write Transfer

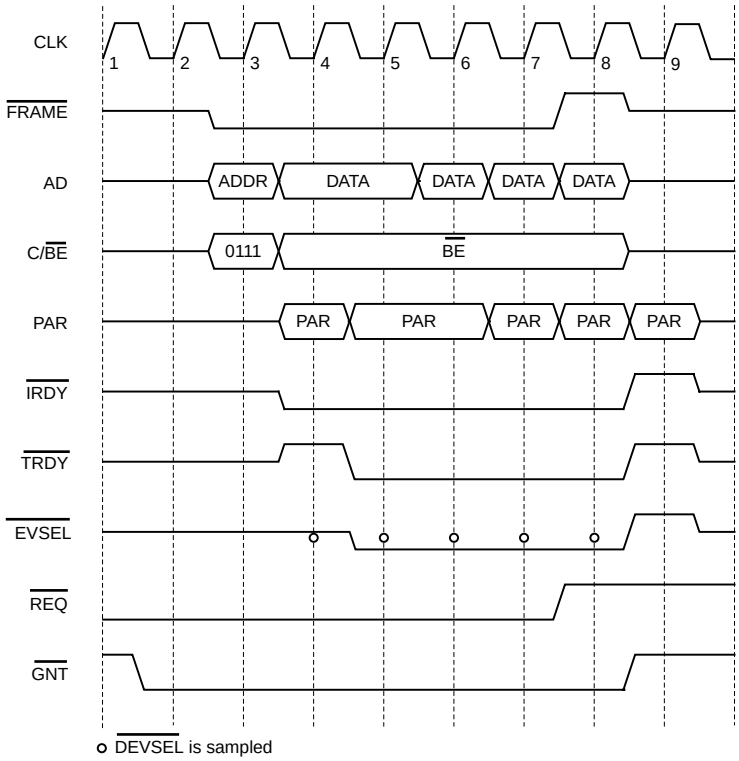
Figure 15 shows a typical burst write access. The Am79C971 controller arbitrates for the bus, is granted access, and writes four 32-bit words (DWords) to the system memory and then releases the bus. In this example, the memory system extends the data phase of the first access by one wait state. The following three data phases take one clock cycle each, which is determined by the timing of TRDY. The example assumes that EXTREQ (BCR18, bit 8) is set to 1, therefore, REQ is not deasserted until the next to last data phase is finished.

Target Initiated Termination

When the Am79C971 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

Figure 16 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 Am79C971 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 7. The Am79C971 controller will again request the bus after two clock cycles, if it wants to transfer more data. The starting address of the new transfer will be the address of the next non-transferred data.



20550D-18

Figure 15. Burst Write Transfer (EXTREQ = 1)

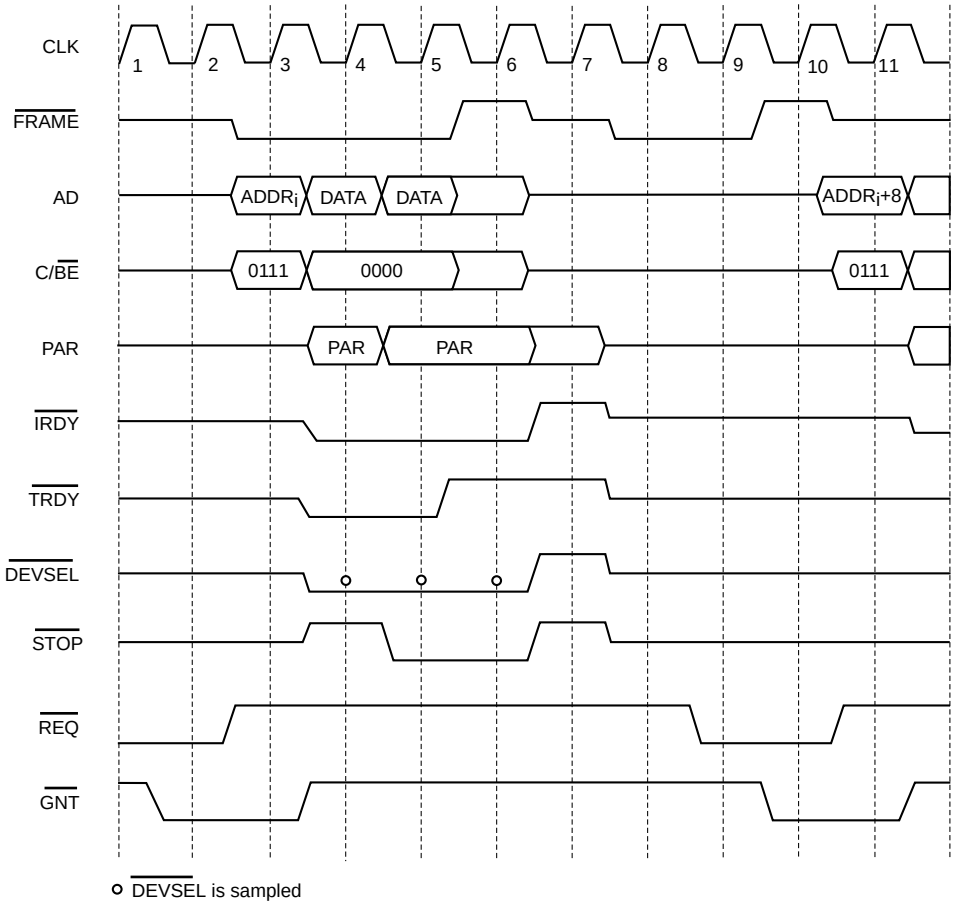


Figure 16. Disconnect With Data Transfer

Disconnect Without Data Transfer

Figure 17 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 Am79C971 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 7. The Am79C971 controller will again request the bus after two clock cycles to retry the last transfer. The starting address of the new transfer will be the address of the last non-transferred data.

Target Abort

Figure 18 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 Am79C971 controller cannot make any assumption about the success of the previous data transfers in the current transaction. The Am79C971 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 Am79C971 controller cannot recover from a target abort event. The Am79C971 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.

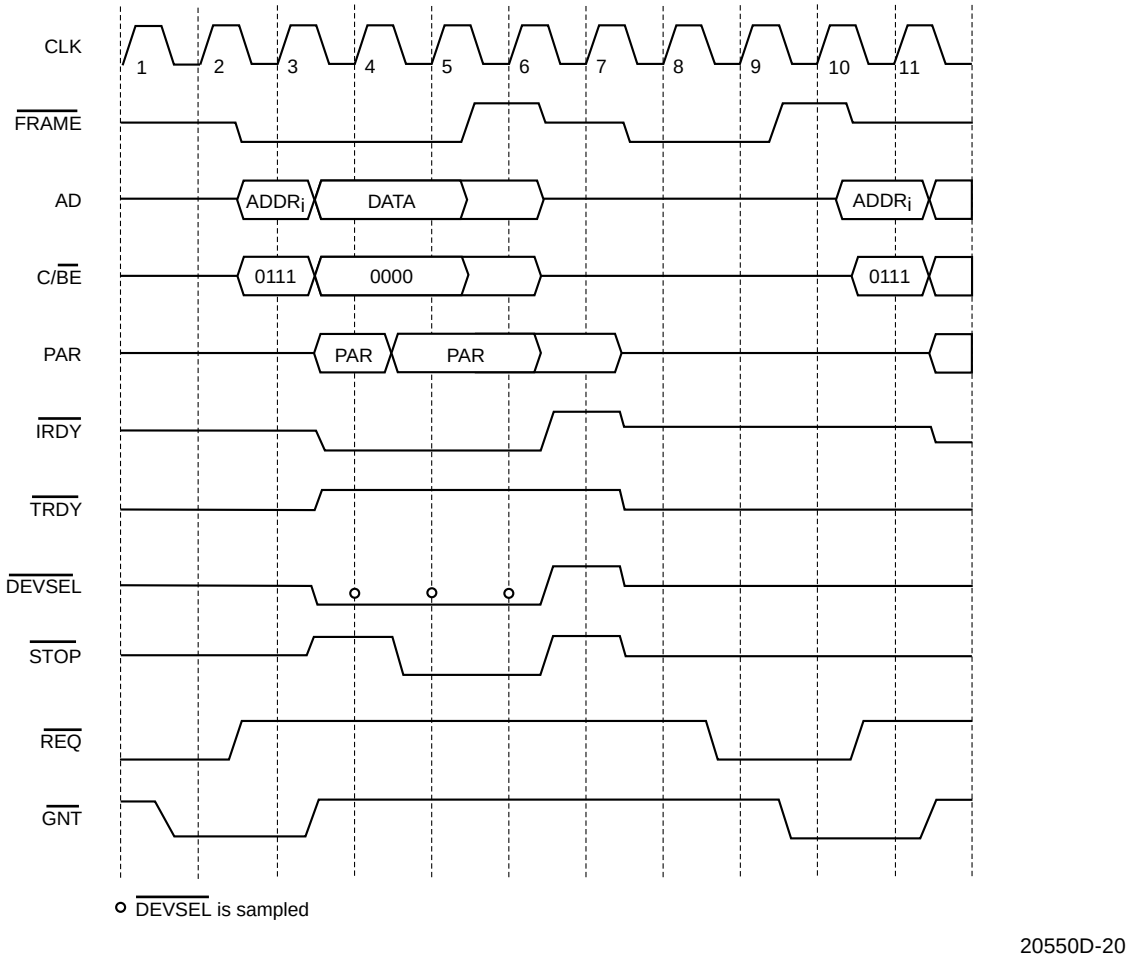


Figure 17. Disconnect Without Data Transfer

RTABORT (PCI Status register, bit 12) will be set to indicate that the Am79C971 controller has received a target abort. In addition, SINT (CSR5, bit 11) will be set to 1. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to 1. 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.

Master Initiated Termination

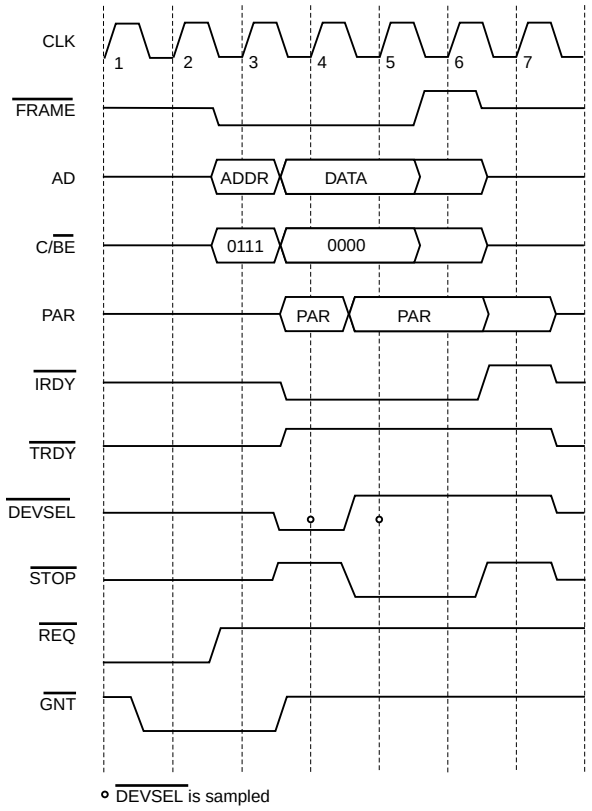
There are three scenarios besides normal completion of a transaction where the Am79C971 controller will terminate the cycles it produces on the PCI bus.

Preemption During Non-Burst Transaction

When the Am79C971 controller performs multiple non-burst transactions, it keeps REQ asserted until the assertion of FRAME for the last transaction. When GNT is removed, the Am79C971 controller will finish the current transaction and then release the bus. If it is not the last transaction, REQ will remain asserted to regain bus ownership as soon as possible. See Figure 19.

Preemption During Burst Transaction

When the Am79C971 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 GNT at any time during the transaction. The Am79C971 controller will ignore the deassertion of GNT and continue with data transfers, as long as the PCI Latency Timer is not expired. When the Latency Timer is 0 and GNT is deasserted, the Am79C971 controller will finish the current data phase, deassert FRAME, finish the last data phase, and release the bus. If EXTREQ (BCR18, bit 8) is cleared to 0, it will immediately assert REQ to regain bus ownership as soon as possible. If EXTREQ is set to 1, REQ will stay asserted.



20550D-21

Figure 18. Target Abort

When the preemption occurs after the counter has counted down to 0, the Am79C971 controller will finish the current data phase, deassert 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 Am79C971 controller. The host can determine this bus bandwidth requirement by reading the PCI MAX_LAT and MIN_GNT registers.

Figure 20 assumes that the PCI Latency Timer has counted down to 0 on clock 7.

Master Abort

The Am79C971 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 Am79C971 controller.

The Am79C971 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.

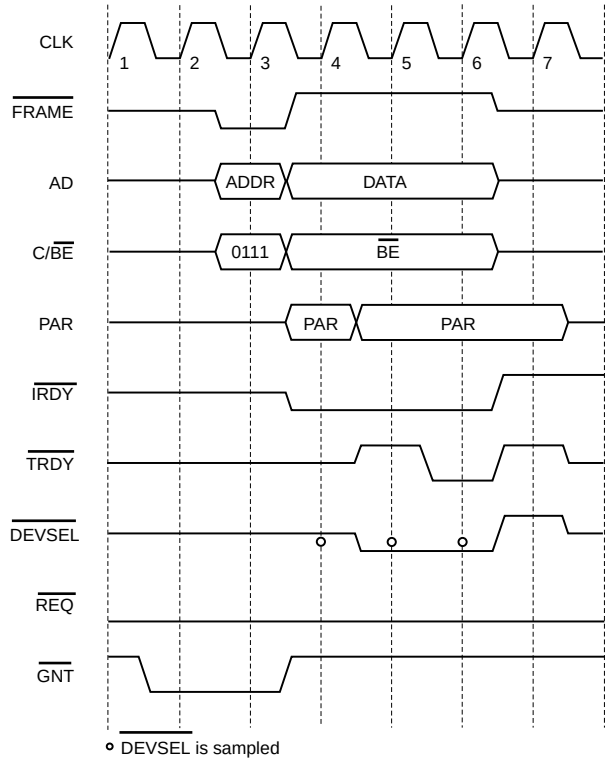
RMABORT (in the PCI Status register, bit 13) will be set to indicate that the Am79C971 controller has terminated its transaction with a master abort. In addition, SINT (CSR5, bit 11) will be set to 1. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to 1. This mechanism 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. See Figure 21.

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 Am79C971 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 1. When reporting of that error is enabled by setting PERREN (PCI Command register, bit 6) to 1, the Am79C971 controller also drives the PERR signal low and sets DATAPERR (PCI Status register, bit 8) to 1. The assertion of PERR follows the corrupted data/byte enables by two clock cycles and PAR by one clock cycle.

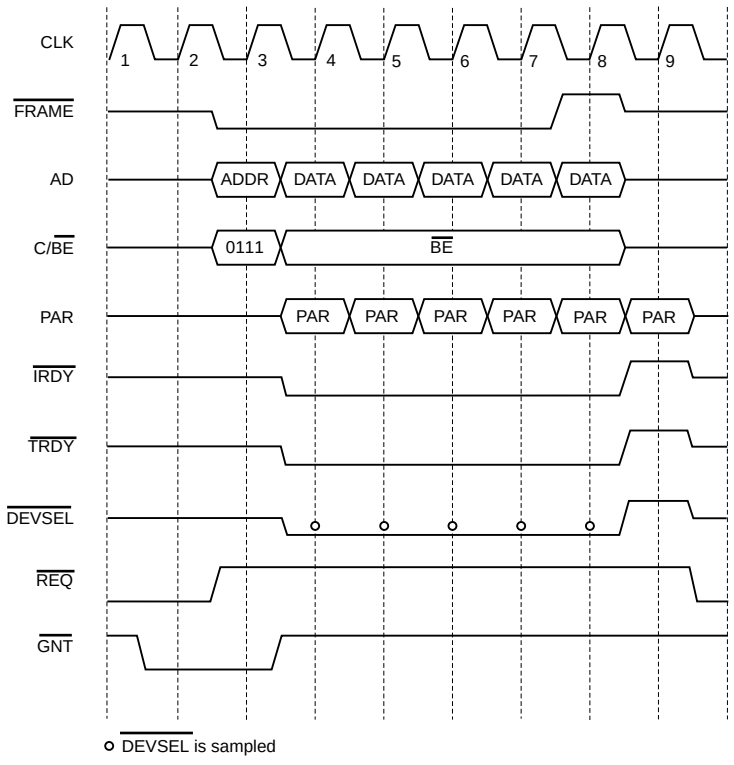
Figure 22 shows a transaction that has a parity error in the data phase. The Am79C971 controller asserts PERR on clock 8, two clock cycles after data is valid. The data on clock 5 is not checked for parity, since on a read access PAR is only required to be valid one clock after the target has asserted TRDY. The Am79C971 controller then drives PERR high for one clock cycle, since PERR is a sustained tri-state signal.

During every data phase of a DMA write operation, the Am79C971 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 1. When PERREN (PCI Command register, bit 6) is set to 1, the Am79C971 controller also sets DATAPERR (PCI Status register, bit 8) to 1.



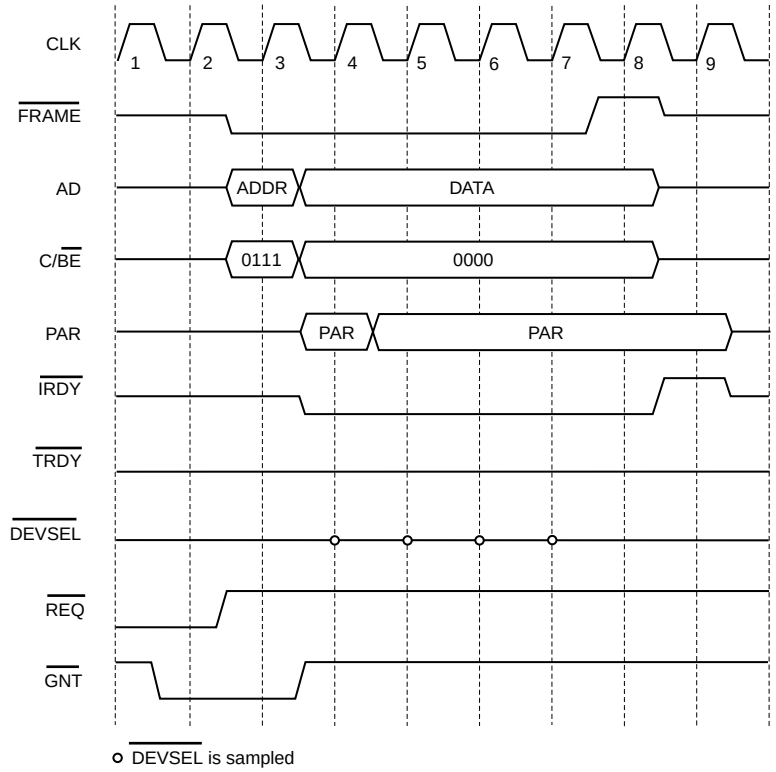
20550D-22

Figure 19. Preemption During Non-Burst Transaction



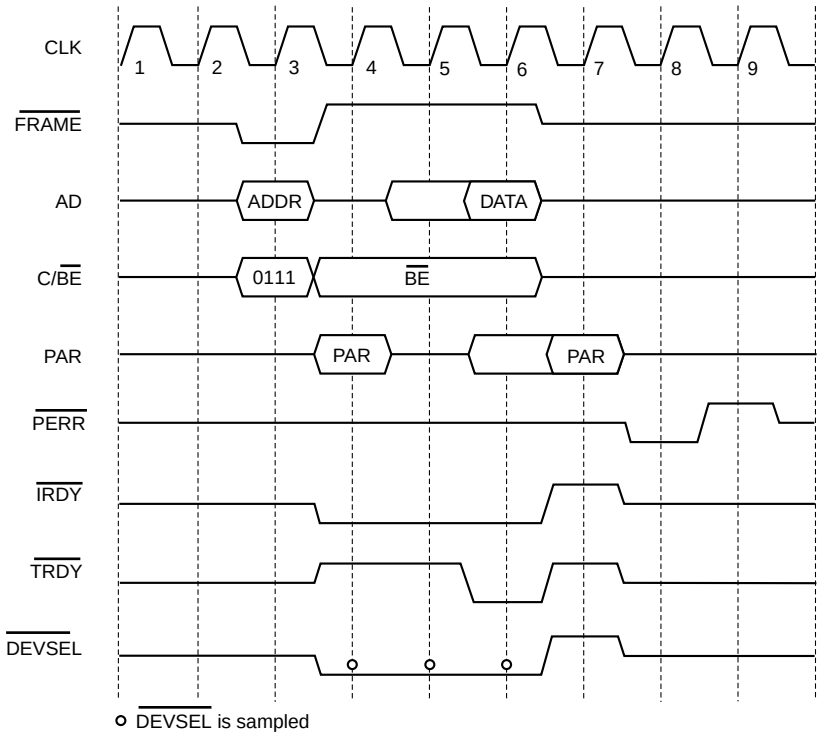
20550D-23

Figure 20. Preemption During Burst Transaction



20550D-24

Figure 21. Master Abort



20550D-25

Figure 22. Master Cycle Data Parity Error Response

Whenever the Am79C971 controller is the current bus master and a data parity error occurs, SINT (CSR5, bit 11) will be set to 1. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to 1. 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).

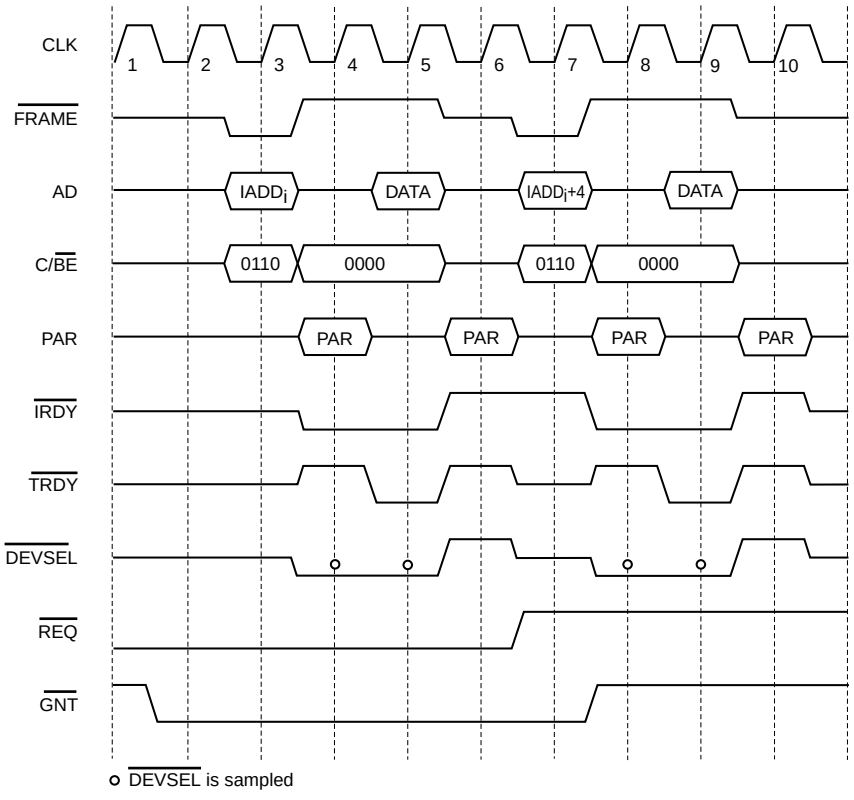
By default, a data parity error does not affect the state of the MAC engine. The Am79C971 controller treats the data in all bus master transfers that have a parity error as if nothing has happened. All network activity continues.

Advanced Parity Error Handling

For all DMA cycles, the Am79C971 controller provides a second, more advanced level of parity error handling. This mode is enabled by setting APERREN (BCR20, bit 10) to 1. When APERREN is set to 1, 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 2 or 3 to program the Am79C971 controller to use 32-bit software structures. The Am79C971 controller will react in the following way when a data parity error occurs:

- Initialization block read: STOP (CSR0, bit 2) is set to 1 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 1 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 1 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.



20550D-26

Figure 23. Initialization Block Read In Non-Burst Mode

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 Am79C971 controller is the target of the transfer.

Initialization Block DMA Transfers

During execution of the Am79C971 controller bus master initialization procedure, the Am79C971 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 1 (i.e., the initialization block is organized as 32-bit software structures), there are seven 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 0 (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 Am79C971 supports two transfer modes for reading the initialization block: non-burst and burst mode, with burst mode being the preferred mode when the Am79C971 controller is used in a PCI bus application. See Figure 23 and Figure 24.

When BREADE is cleared to 0 (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.

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

Descriptor DMA Transfers

Am79C971 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 Am79C971 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 Am79C971 controller performs descriptor read operations.

When SWSTYLE is set to 0 or 2, all descriptor read operations are performed in non-burst mode. The setting of BREADE has no effect in this configuration. See Figure 25.

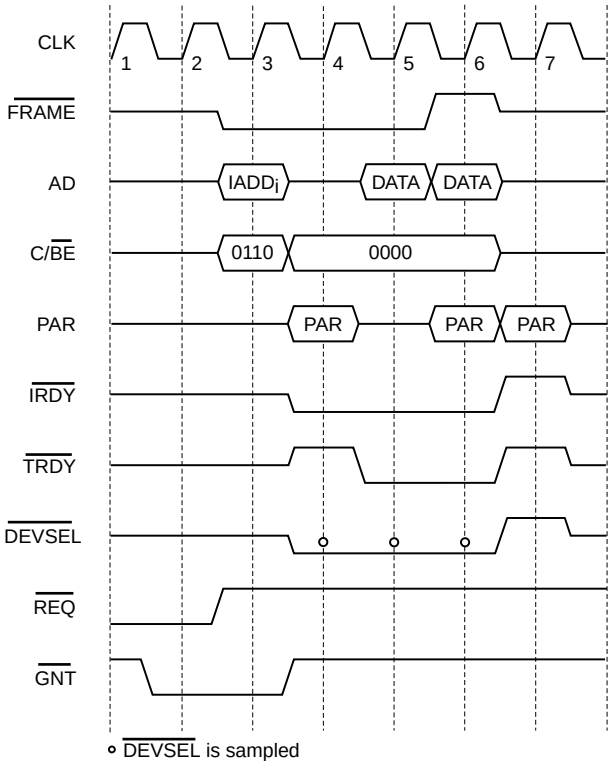
When SWSTYLE is set to 3, the descriptor entries are ordered to allow burst transfers. The Am79C971 controller will perform all descriptor read operations in burst mode, if BREADE is set to 1. See Figure 26.

Table 4 shows the descriptor read sequence.

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

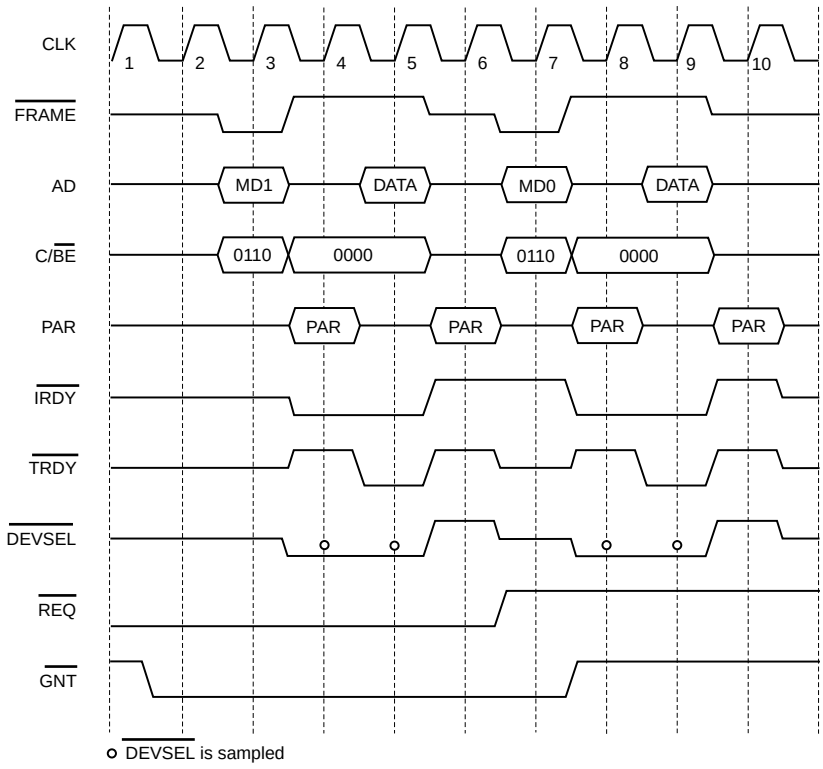
Table 4. Descriptor Read Sequence

SWSTYLE BCR20[7:0]	BREADE BCR18[6]	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]
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]



20550D-27

Figure 24. Initialization Block Read In Burst Mode



20550C-28

Figure 25. Descriptor Ring Read In Non-Burst Mode

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 0 (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 2 or 3 (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 data transfer writes a DWord containing status information. The second data transfer writes a byte (SWSTYLE cleared to 0), or otherwise a word containing additional status and the ownership bit (i.e., MD1[31]).

The settings of SWSTYLE (BCR20, bits 7-0) and BWRITE (BCR18, bit 5) affect the way the Am79C971 controller performs descriptor write operations.

When SWSTYLE is set to 0 or 2, 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 3, the descriptor entries are ordered to allow burst transfers. The Am79C971 controller will perform all descriptor write operations in burst mode, if BWRITE is set to 1. See Table 5 for the descriptor write sequence.

A write transaction to the descriptor ring entries is the only case where the Am79C971 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 $\overline{\text{IRDY}}$ is deasserted.

Note that Figure 26 assumes that the Am79C971 controller is programmed to use 32-bit software structures (SWSTYLE = 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).

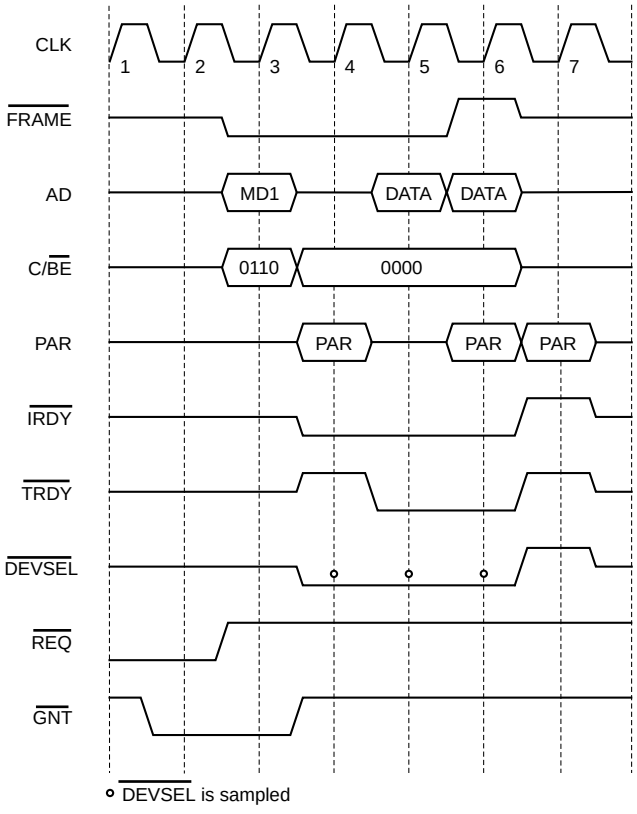


Figure 26. Descriptor Ring Read In Burst Mode

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]
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]

FIFO DMA Transfers

Am79C971 microcode will determine when a FIFO DMA transfer is required. This transfer mode will be used for transfers of data to and from the Am79C971 FIFOs. Once the Am79C971 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.

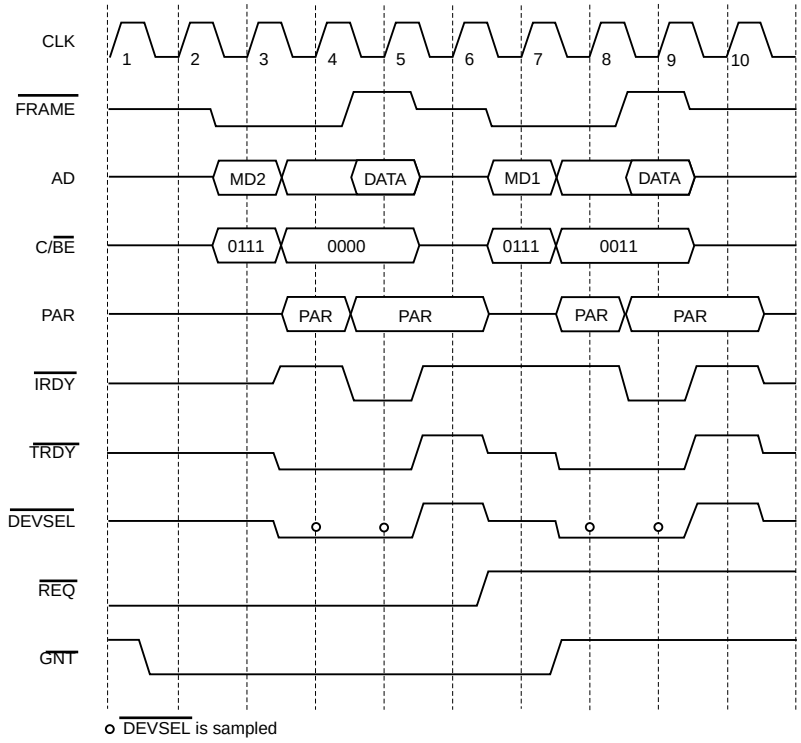
Non-Burst FIFO DMA Transfers

In the default mode, the Am79C971 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.

$\overline{\text{FRAME}}$ will be deasserted after every address phase. Several factors will affect the length of the bus mastership period. The possibilities are as follows:

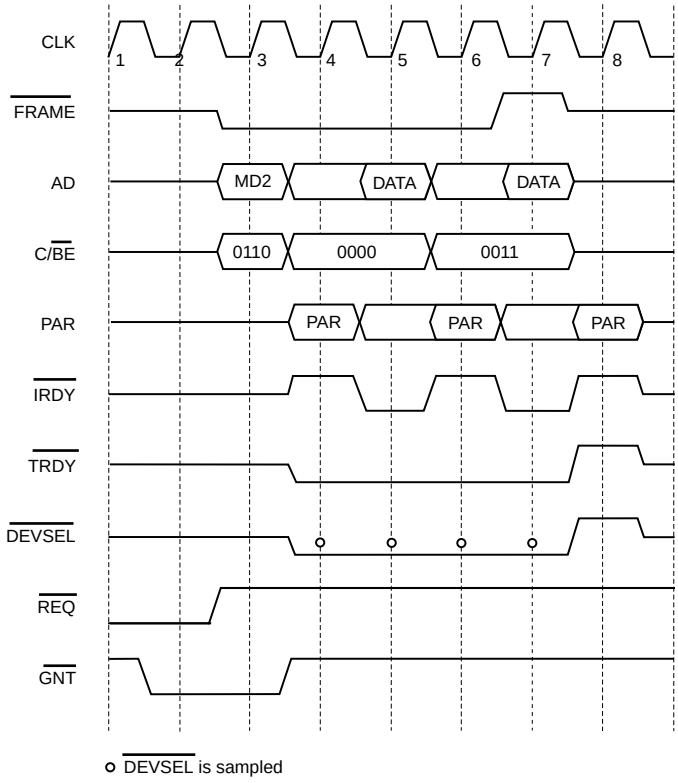
Bus cycles will continue until the transmit FIFO is filled to its high threshold (read transfers) or the receive FIFO is emptied to its low threshold (write transfers). The exact number of total transfer cycles in the bus mastership period is dependent on all of the following variables: the settings of the FIFO watermarks, the conditions of the FIFOs, the latency of the system bus to the Am79C971 controller's bus request, the speed of bus operation and bus preemption events. 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 higher the receive watermark, the longer the bus mastership period will be.

Note: *The PCI Latency Timer is not significant during non-burst transfers.*



20550D-30

Figure 27. Descriptor Ring Write In Non-Burst Mode



20550D-31

Figure 28. Descriptor Ring Write In Burst Mode

Burst FIFO DMA Transfers

Bursting is only performed by the Am79C971 controller if the BREADE and/or BWRITE bits of BCR18 are set. These bits individually enable/disable the ability of the Am79C971 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 0 during the address phase indicating a linear burst order.

During FIFO DMA read operations, all byte lanes will always be active. The Am79C971 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.

Figure 29 shows the beginning of a FIFO DMA write with the beginning of the buffer not aligned to a DWord boundary. The Am79C971 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 Am79C971 controller can continue bursting full DWords.

If a receive buffer does not end on a DWord boundary, the Am79C971 controller will perform a non-DWord write on the last transfer to the buffer. Figure 30 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 Am79C971 controller bursts three data phases. The first two data phases write a full DWord, the last one only writes a single byte.

Note that the Am79C971 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 Am79C971 controller will write a full DWord, containing the last byte of the receive frame in the least significant byte position (BSWP is cleared to 0, 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.

If the end of a receive buffer is not aligned to a DWord boundary, IWAIT (BCR18, bit 10) must stay at its default value of 0. This will result in one wait state added to every data phase in a burst write transaction. When the software ensures that all receive buffers end on a DWord boundary, IWAIT can be set to 1. In this mode, the Am79C971 controller will only insert a wait state in the first data phase of the burst write transaction.

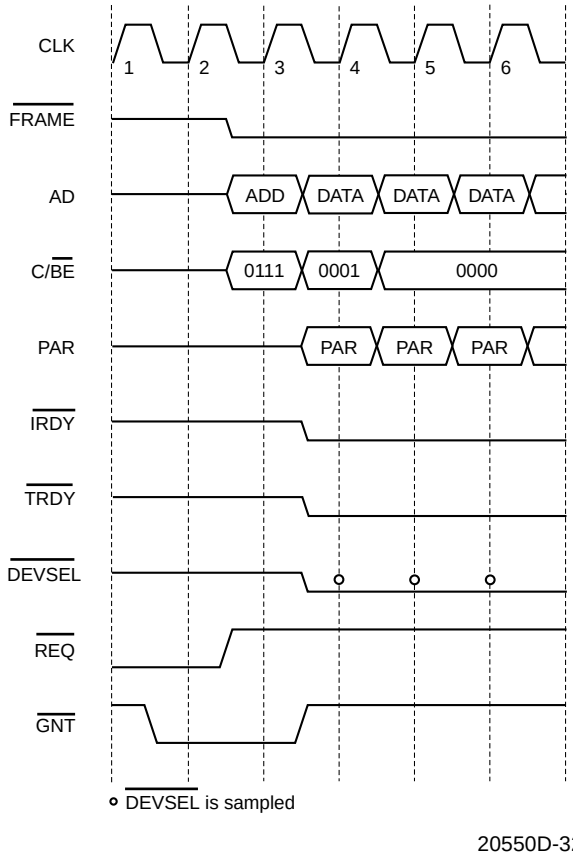


Figure 29. FIFO Burst Write At Start Of Unaligned Buffer

In a PCI bus application, the Am79C971 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 0. In this mode, the Am79C971 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 Am79C971 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 do not use the PCI Latency Timer or that do not support preemption, the following rules apply to limit the time the Am79C971 controller takes on the bus:

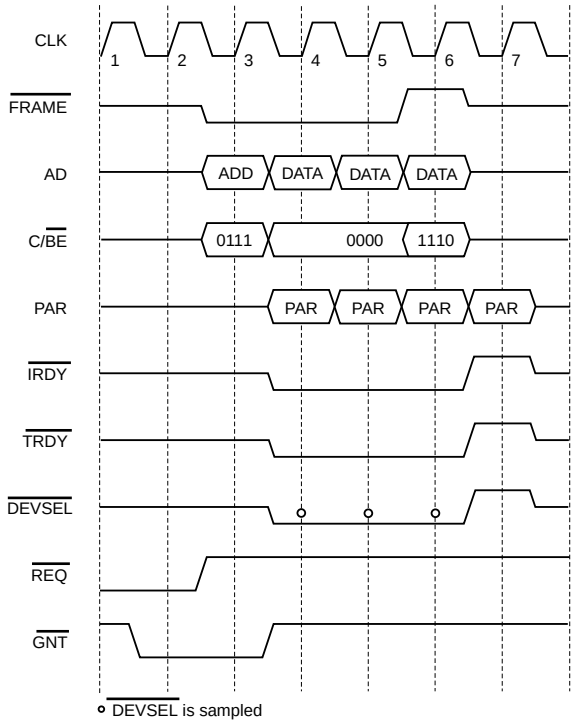


Figure 30. FIFO Burst Write At End Of Unaligned Buffer

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 Am79C971 controller's bus request, and the speed of bus operation. The 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 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 Am79C971 controller will not relinquish bus ownership until the PCI Latency Timer expires.

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

Am79C971 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 0, 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 1, all initialization block entries are logically 32-bits wide. Note that the Am79C971 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 Am79C971 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 Am79C971 operation, together with the base addresses and length information of the transmit and receive descriptor rings.

There is an alternate method to initialize the Am79C971 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, Alternative Method for Initialization* for details on this alternate method.

Re-Initialization

The transmitter and receiver sections of the Am79C971 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 Am79C971 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 Am79C971 controller as in the C-LANCE device. In particular, upon restart, the Am79C971 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 Am79C971 controller. The reload of descriptor base addresses is performed in the C-LANCE device only after initialization, so that a restart of the C-LANCE without initialization leaves the C-LANCE pointing at the same descriptor locations as before the restart.

Suspend

The Am79C971 controller offers two suspend modes that allows easy updating of the CSR registers without going through a full re-initialization of the device. The suspend modes also allow stopping the device with orderly termination of all network activity.

The host requests the Am79C971 controller to enter the suspend mode by setting SPND (CSR5, bit 0) to 1. The host must poll SPND until it reads back 1 to determine that the Am79C971 controller has entered the suspend mode. When the host sets SPND to 1, the procedure taken by the Am79C971 controller to enter the suspend mode depends on the setting of the fast suspend enable bit (FASTSPND, CSR7, bit 15).

When a fast suspend is requested (FASTSPND is set to 1), the Am79C971 controller performs a quick entry into the suspend mode. At the time the SPND bit is set, the Am79C971 controller will continue the DMA process of any transmit and/or receive packets that have already begun DMA activity until the network activity has been completed. In addition, any transmit packet that had started transmission will be fully transmitted and any receive packet that had begun reception will be fully received. However, no additional packets will be transmitted or received and no additional transmit or receive DMA activity will begin after network activity has ceased. Hence, the Am79C971 controller may enter the suspend mode with transmit and/or receive packets still in the FIFOs or external SRAM. This offers a worst case suspend time of a maximum length packet over the possibility of completely emptying the external SRAM. Care must be exercised in this mode, because the entire memory subsystem of the Am79C971 controller is suspended. Any changes to either the descriptor rings or the external SRAM can cause the Am79C971 controller to start up in an unknown condition and could cause data corruption.

When FASTSPNDE is 0 and the SPND bit is set, the Am79C971 controller may take longer before entering the suspend mode. At the time the SPND bit is set, the Am79C971 controller will complete the DMA process of a transmit packet if it had already begun and the Am79C971 controller will completely receive a receive packet if it had already begun. The Am79C971 controller will not receive any new packets after the completion of the current reception. Additionally, all transmit packets stored in the transmit FIFOs and the transmit buffer area in the external SRAM (if one is present) will be transmitted, and all receive packets stored in the receive FIFOs and the receive buffer area in the external SRAM (if one is present) will be transferred into system memory. Since the FIFO and external SRAM contents are flushed, it may take much longer before the Am79C971 controller enters the suspend mode. The amount of time that it takes depends on many factors including the size of the external SRAM, bus latency, and network traffic level.

Upon completion of the described operations, the Am79C971 controller sets the read-version of SPND to 1 and enters the suspend mode. In suspend mode, all of the CSR and BCR registers are accessible. As long as the Am79C971 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 SPND is set to 0, the Am79C971 controller will leave the suspend mode and will continue at the transmit and receive descriptor ring locations, where it had been when it entered the suspend mode.

See the section on *Magic Packet*™ technology for details on how that affects suspension of the Am79C971 controller.

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 0, the descriptor rings are backwards compatible with the Am79C90 C-LANCE and the 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

(SSIZE32 (BCR20, bit 8) is set to 0). Note that even though the Am79C971 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 2 or 3, 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 1). The fourth DWord is reserved. When SWSTYLE is set to 3, 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.

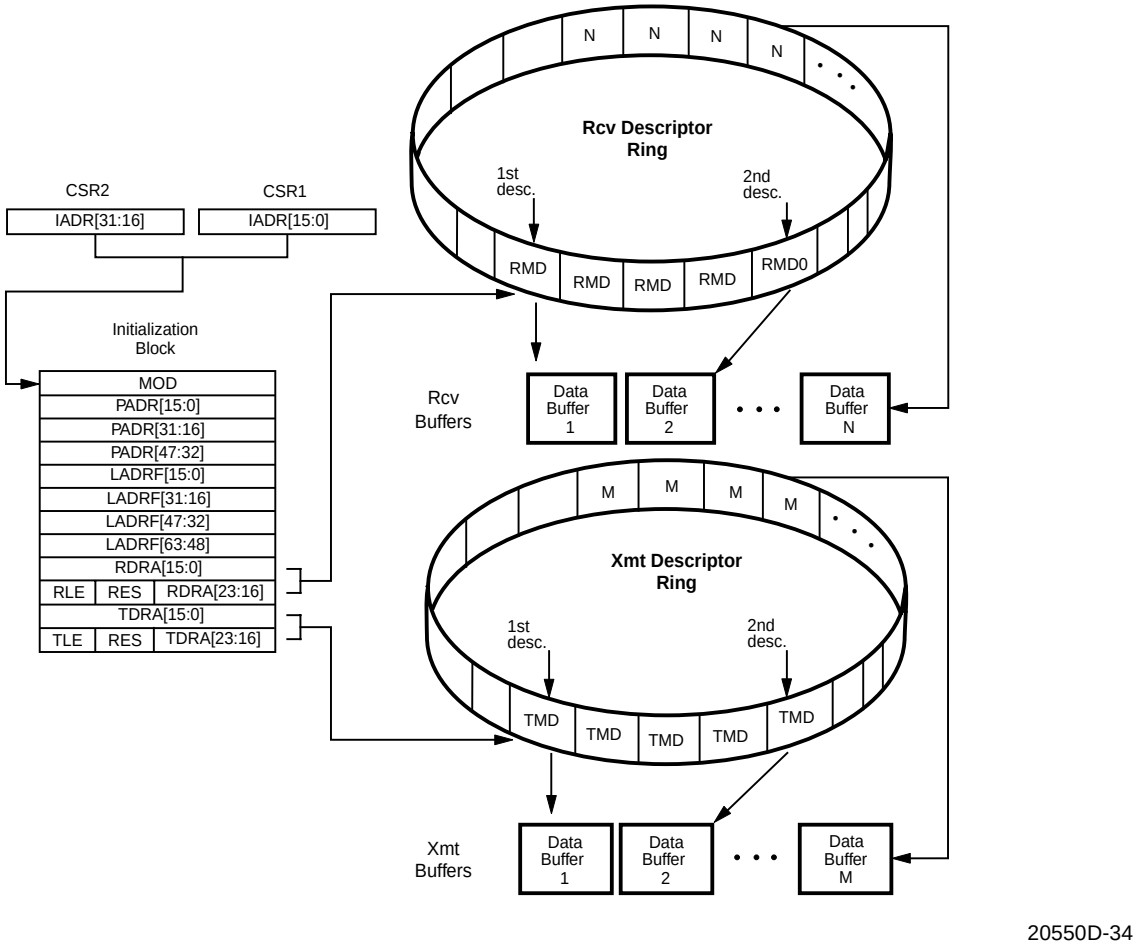
Each ring entry contains the following information:

- 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 Am79C971 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 1, it signifies that the Am79C971 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 Am79C971 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 Am79C971 controller sets OWN to 0 to release ownership to the software. (When LAPPEN (CSR3, bit 5) is set to 1, this rule is modified. See the LAPPEN description. At initialization, the Am79C971 controller reads the base address of both the transmit and receive descriptor rings into CSRs for use by the Am79C971 controller during subsequent operations.

Figure 31 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 0.



20550D-34

Figure 31. 16-Bit Software Model

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 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 1.

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 Am79C971 controller, then the Am79C971 controller will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership. If the TXDPOLL bit in CSR4 is set, then the transmit polling function is disabled.

A typical polling operation consists of the following sequence. The Am79C971 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. Am79C971 controller does not own the current RDTE *and* the poll time has elapsed *and* RXON = 1 (CSR0, bit 5), *or*
2. Am79C971 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.

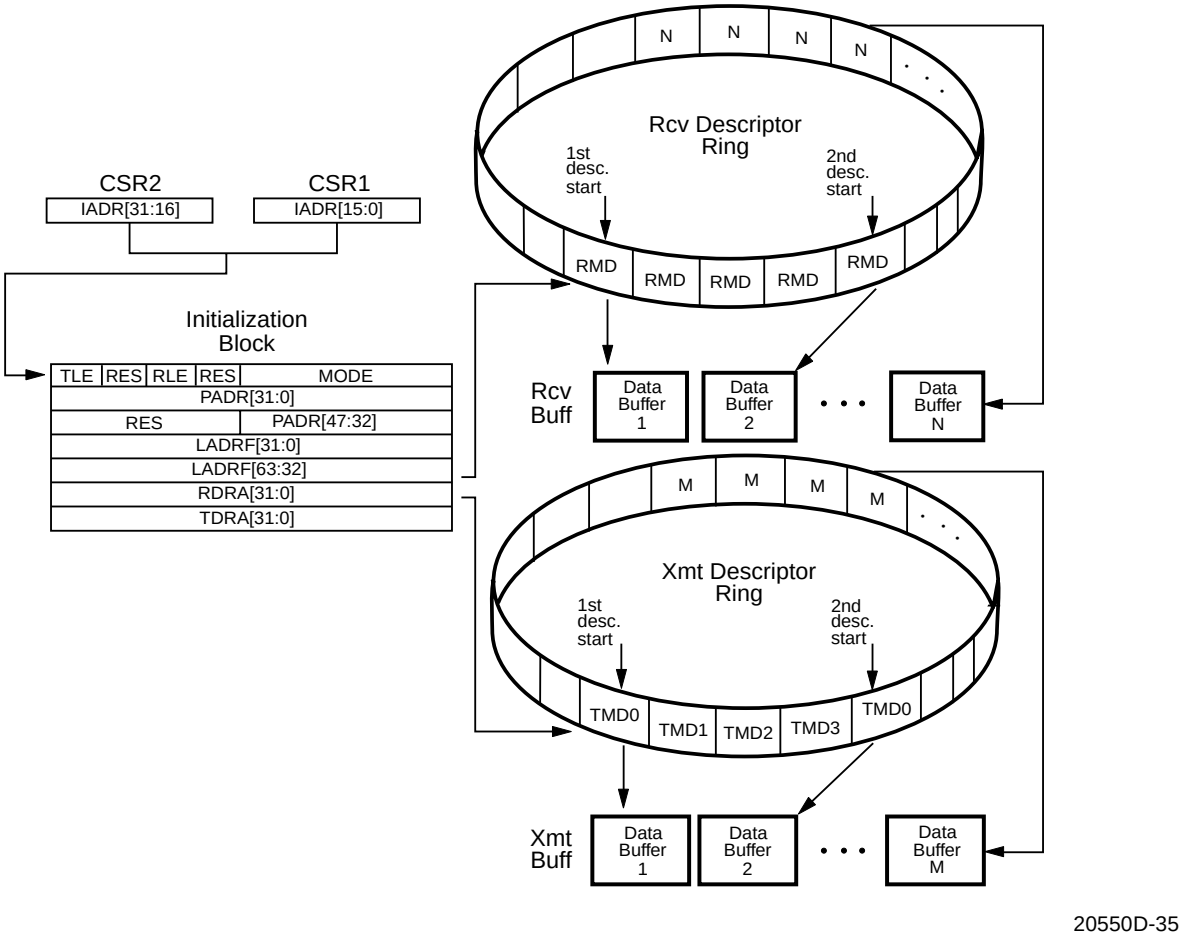


Figure 32. 32-Bit Software Model

If RXON is cleared to 0, the Am79C971 controller will never poll RDTE locations.

In order to avoid missing frames, the system should have at least one 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. Am79C971 controller does not own the current TDTE *and* TXDPOLL = 0 (CSR4, bit 12) *and* TXON = 1 (CSR0, bit 4) *and* the poll time has elapsed, *or*
2. Am79C971 controller does not own the current TDTE *and* TXDPOLL = 0 *and* TXON = 1 *and* a frame has just been received, or
3. Am79C971 controller does not own the current TDTE *and* TXDPOLL = 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 Am79C971 controller finds that the OWN bit of that TDTE is not set, the Am79C971 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 Am79C971 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 Am79C971 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 device, the buffer length of 0 is interpreted as a 4096-byte buffer. A zero length buffer 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 Am79C971 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 Am79C971 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 0, the underflow error will cause the transmitter to be disabled (CSR0, TXON = 0). The Am79C971 controller will have to be re-initialized to restore the transmit function. Setting DXSUFLO to 1 enables the Am79C971 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 Am79C971 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 Am79C971 controller normally clears OWN bits in strict FIFO order. However, the Am79C971 controller can queue up to two frames in the transmit FIFO. When the second frame uses buffer chaining, the Am79C971 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 Am79C971 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 Am79C971 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 Am79C971 controller will skip over the rest of the frame which experienced the error. This is done by returning to the polling microcode where the Am79C971 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 Am79C971 controller will always perform another polling operation. As described earlier, this polling operation will begin with a check of the current RDTE, unless the Am79C971 controller already owns that descriptor. Then the Am79C971 controller will poll the next TDTE. If the transmit descriptor OWN bit has a 0 value, the Am79C971 controller will resume incrementing the poll time counter. If the transmit descriptor OWN bit has a value of 1, the Am79C971 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 Am79C971 controller to avoid inserting poll time counts between successive transmit frames.

By default, whenever the Am79C971 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 Am79C971 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 1. Another mode, which is enabled by setting LTINTEN (CSR5, bit 14) to 1, allows suppression of interrupts for successful transmissions for all but the last frame in a sequence.

Receive Descriptor Table Entry

If the Am79C971 controller does not own both the current and the next Receive Descriptor Table Entry (RDTE), then the Am79C971 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 Am79C971 controller, then additional poll accesses are not necessary. Future poll operations will not include RDTE accesses as long as the Am79C971 controller retains ownership of the current and the next RDTE.

When receive activity is present on the channel, the Am79C971 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 Am79C971 controller checks the current receive buffer status register CRST (CSR41) to determine the ownership of the current buffer.

If ownership is lacking, the Am79C971 controller will immediately perform a final poll of the current RDTE. If ownership is still denied, the Am79C971 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. Another poll of the current RDTE will not occur until the frame has finished.

If the Am79C971 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 Am79C971 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 Am79C971 controller does not own the second buffer, ownership of the current descriptor will be passed back to the system by writing a 0 to the OWN bit of RMD1. 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 Am79C971 controller does own the second (next) buffer, ownership will be passed back to the system by writing a 0 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 Am79C971 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 Am79C971 controller recognizes the completion of the frame (the last byte of this receive message has been removed from the FIFO). The Am79C971 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.

Receive Frame Queuing

The Am79C971 controller supports the lack of RDTEs when external SRAM (SRAM SIZE in BCR 25, bits 7-0) is present through the Receive Frame Queuing mechanism. When the SRAM SIZE = 0, then the Am79C971 controller reverts back to the PCnet PCI II mode of operation. This operation is automatic and does not require any programming by the host. When SRAM is present, the Receive Frame Queuing mechanism allows a slow protocol to manage more frames without the high frame loss rate normally attributed to FIFO based network controllers.

The Am79C971 controller will store the incoming frames in the extended FIFOs until polling takes place; if enabled, it discovers it owns an RDTE. The stored frames are not altered in any way until written out into system buffers. When the receive FIFO overflows, further incoming receive frames will be missed during that time. As soon as the network receive FIFO is empty, incoming frames are processed as normal. Status on a per frame basis is not kept during the overflow process. Statistic counters are maintained and accurate during that time.

During the time that the Receive Frame Queuing mechanism is in operation, the Am79C971 controller relies on the Receive Poll Time Counter (CSR 48) to control the worst case access to the RDTE. The Receive Poll Time Counter is programmed through the Receive Polling Interval (CSR49) register. The Received Polling Interval defaults to approximately 2 ms. The Am79C971 controller will also try to access the RDTE during normal descriptor accesses whether they are transmit or receive accesses. The host can force the Am79C971 controller to immediately access the RDTE by setting the RDMD (CSR 7, bit 13) to 1. Its operation is similar to the transmit one. The polling process can be disabled by setting the RXDPOLL (CSR7, bit 12) bit. This will stop the automatic polling process and the host must set the RDMD bit to initiate the receive process into host memory. Receive frames are still stored even when the receive polling process is disabled.

Software Interrupt Timer

The Am79C971 controller is equipped with a software programmable free-running interrupt timer. The timer is constantly running and will generate an interrupt STINT (CSR 7, bit 11) when STINITE (CSR 7, bit 10) is set to 1. After generating the interrupt, the software timer will load the value stored in STVAL and restart. The timer value STVAL (BCR31, bits 15-0) is interpreted as an unsigned number with a resolution of 12.8 µs. For instance, a value of 122 ms would be programmed with

a value of 9531 (253Bh). The default value of STVAL is FFFFh which yields the approximate maximum 838 ms timer duration. A write to STVAL restarts the timer with the new contents of STVAL.

Media Access Control

The Media Access Control (MAC) engine incorporates the essential protocol requirements for operation of an Ethernet/IEEE 802.3-compliant node, and provides the interface between the FIFO subsystem and the Manchester Encoder/Decoder (MENDEC) or the MII. The MAC engine has been changed from a single-bit wide engine into a 4-bit (nibble) wide engine. This was done to accommodate the nibble wide MII.

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 features 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, except in full-duplex operation)
  - — Contention resolution (collision handling, except in full-duplex operation)

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 1, 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 1, the receiver will automatically strip pad bytes from the received message by observing the value in the length field and by 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 MAC does not use the content in the length/type field unless APAD_XMT (CSR4, bit 11) is set and the data portion of the frame is shorter than 60 bytes.

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. During MII operation, the MAC engine will detect the incoming preamble sequence when the RX_DV signal is activated by the external PHY. The MAC will discard the preamble and begin searching for the SFD except in the case of 100BASE-T4. In that case, the SFD will be the first nibble across the MII interface. Once the SFD is detected, all subsequent nibbles 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 Am79C971 controller has the ability to accept runt packets for diagnostic 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 (1, MORE, RTRY, and TRC).
- Whether the MAC engine had to Defer (DEF) due to channel activity.
- Excessive deferral (EXDEF), indicating that the transmitter experienced Excessive Deferral on this transmit frame, where Excessive Deferral is defined in the ISO 8802-3 (IEEE/ANSI 802.3) standard.
- 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 because the transceiver does not support this feature (or it is disabled). SQE Test is only valid for 10-Mbps networks.

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 as 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 is a non-integral number of bytes in the message.

During the reception, the FCS is generated on every nibble (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 0, 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 indicate one extra byte.

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 IEEE 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 optionally a two-part deferral after a receive message.

See ANSI/IEEE Std 802.3-1993 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 inter-Frame 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-SpacingPart1 is other than 0:*

1. *Upon completing a transmission, start timing the interrupted gap, as soon as transmitting and carrier sense are both false.*
2. *When timing an inter-frame gap following reception, reset the inter-frame gap timing if carrier sense becomes true during the first 2/3 of the inter-frame 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 0.*

The MAC engine implements the optional receive two part deferral algorithm, with an InterFrameSpacingPart1 time of 6.0 µs. The InterFrameSpacingPart 2 interval is, therefore, 3.4 µs.

The Am79C971 controller will perform the two-part deferral algorithm as specified in Section 4.2.8 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 9.6 µs InterFrameSpacing after the receive carrier is deasserted. During the first part deferral (InterFrameSpacingPart1 - IFS1), the Am79C971 controller will defer any pending transmit frame and respond to the receive message. The IPG counter will be cleared to 0 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 Am79C971 controller will begin timing the second part deferral (InterFrameSpacingPart2 - IFS2) of 3.4 µs. Once IFS1 has completed and IFS2 has commenced, the Am79C971 controller will not defer to a receive frame if a transmit frame is pending. This means that the Am79C971 controller will not attempt to receive the receive frame, since it will start to transmit and generate a collision at 9.6 µs. The Am79C971 controller will complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

The Am79C971 controller allows the user to program the IPG and the first part deferral (InterFrame-SpacingPart1 - IFS1) through CSR125. By changing the IPG default value of 96 bit times (60h), the user can adjust the fairness or aggressiveness of the Am79C971 MAC on the network. By programming a lower number of bit times than the ISO/IEC 8802-3 standard requires, the Am79C971 MAC engine will become more aggressive on the network. This aggressive nature will give rise to the Am79C971 controller possibly *capturing the network* at times by forcing other less aggressive compliant nodes to defer. By programming a larger number of bit times, the Am79C971 MAC will become less aggressive on the network and may defer more often than normal. The performance of the Am79C971 controller may decrease as the IPG value is increased from the default value, but the resulting behavior may improve network performance by reducing collisions. The Am79C971 controller uses the same IPG for back-to-back transmits and receive-to-transmit accesses. Changing IFS1 will alter the period for which the Am79C971 MAC engine will defer to incoming receive frames.

CAUTION: Care must be exercised when altering these parameters. Adverse network activity could result!

This transmit two-part deferral algorithm is implemented as an option which can be disabled using the DXMT2PD bit in CSR3. The IFS1 programming will have no effect when DXMT2PD is set to 1, but the IPG programming value is still valid. 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 to 15 bit times duration) on the CI± pair (within 0.6 to 1.6 µs after the transmission ceases). During the time period in which the SQE Test message is expected, the Am79C971 controller will not respond to receive carrier sense.

See ANSI/IEEE Std 802.3-1993 Edition, 7.2.4.6 (1):

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

The Am79C971 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 Am79C971 controller will defer its transmission. If carrier is detected within the 4.0 to 6.0 μs IFS1 period, the Am79C971 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) and through the MII via the COL input pin. Both are functionally equivalent in operation.

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 rescheduled to a time determined by the random backoff algorithm. If a single retry was required, the 1 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 (1 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 retransmission is attempted.

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

> *"At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to retransmit the frame. The delay is an integer multiple of slot time. The number of slot times to delay before the nth retransmission attempt is chosen as a uniformly distributed random integer r in the range:*
>
> $$0 \le r < 2^k \text{ where } k = \min(n,10).$$"

The Am79C971 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, while the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time-out their slot time counters as normal.

This modified backoff algorithm is enabled when EMBA (CSR3, bit 3) is set to 1.

Transmit Operation

The transmit operation and features of the Am79C971 controller are controlled by programmable options. The Am79C971 controller offers a large transmit FIFO to provide frame buffering for increased system latency, automatic retransmission 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 before 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 IEEE 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 IEEE 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 Am79C971 controller to compute the actual number of pad bytes to be inserted. The Am79C971 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 Am79C971 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. See Figure 33.

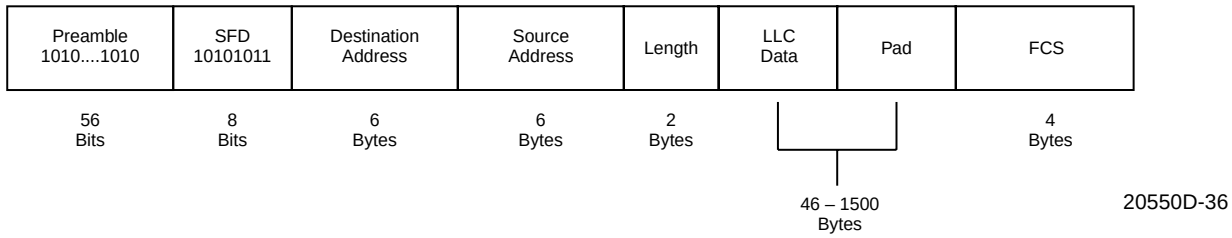


Figure 33. ISO 8802-3 (IEEE/ANSI 802.3) Data Frame

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 Am79C971 controller, therefore, will 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 0, the transmitter will generate and append the FCS to the transmitted frame. If the automatic padding feature is invoked (APAD_XMT is set in CSR4), the FCS will be appended to frames shorter than 64 bytes by the Am79C971 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 set to 1 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 1). Note that bit 29 of TMD1 has the function of ADD_FCS if SWSTYLE (BCR20, bits 7-0) is programmed to 0, 2, or 3.

Transmit Exception Conditions

Exception conditions for frame transmission fall into two distinct categories: those conditions 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 Am79C971 controller include collisions within the slot time with automatic retry. The Am79C971 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 1 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 Am79C971 controller sets the RTRY bit in the current transmit TDTE in host memory (TMD2), gives up ownership (resets the OWN bit to 0) 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 or 100-Mbps networks.)

These conditions should not occur on a correctly configured IEEE 802.3 network operating in half-duplex mode. If they do, they will be reported. 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 Am79C971 controller cannot observe receive activity while it is transmitting on the AUI or GPSI port. In AUI mode, after the Am79C971 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. In GPSI mode, LCAR will be asserted if RXEN does not go active during the transmission.

When the internal 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 Am79C971 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 IPG time following the completion of a transmitted message, the AUI CI± pair is asserted by some transceivers as a self-test. The integral MENDEC will expect the SQE Test Message (nominal 10-MHz sequence) to be returned via the CI± pair within a 40-network bit-time period after DI± 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 Am79C971 controller will set the CERR bit in CSR0. In GPSI mode, CLSN must be asserted after the transmission or otherwise CERR will be set. CERR will be asserted in 10BASE-T mode, or in the 10BASE-T mode through the MII after transmit, if the network port 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 Am79C971 controller are controlled by programmable options. The Am79C971 controller offers a large 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 IEEE 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 that the Am79C971 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 Am79C971 controller can be programmed to accept runt packets by setting RPA in CSR124.

Address Matching

The Am79C971 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 SFD (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 1s, 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 Am79C971 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 1, the Am79C971 controller will not accept unicast frames.

If the incoming frame is multicast, the Am79C971 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 Am79C971 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 Am79C971 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 Am79C971 controller operates in promiscuous mode when PROM (CSR15, bit 15) is set.

In addition, the Am79C971 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 Am79C971 controller to accept the frame. Note that these indicator bits are only available when the Am79C971 controller is programmed to use 32-bit structures for the descriptor entries (BCR20, bit 7-0, SWSTYLE is set to 2 or 3).

PAM (RMD1, bit 22) is set by the Am79C971 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 Am79C971 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 Am79C971 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 0, 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 1 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 Am79C971 controller operates in promiscuous mode and none of the three match bits is set, it is an indication that the Am79C971 controller only accepted the frame because it was in promiscuous mode.

When the Am79C971 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 Am79C971 controller only accepted the frame because it was not rejected by driving the $\overline{EAR}$ pin LOW within 64 bytes after SFD.

See Table 6 for receive address matches.

Automatic Pad Stripping

During reception of an IEEE 802.3 frame, the pad field can be stripped automatically. Setting ASTRP_RCV (CSR4, bit 0) to 1 enables the automatic pad stripping feature. The pad field will be stripped before the frame is passed to the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has had the pad characters stripped.

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

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.

Figure 34 shows the byte/bit ordering of the received length field for an IEEE 802.3-compatible frame format.

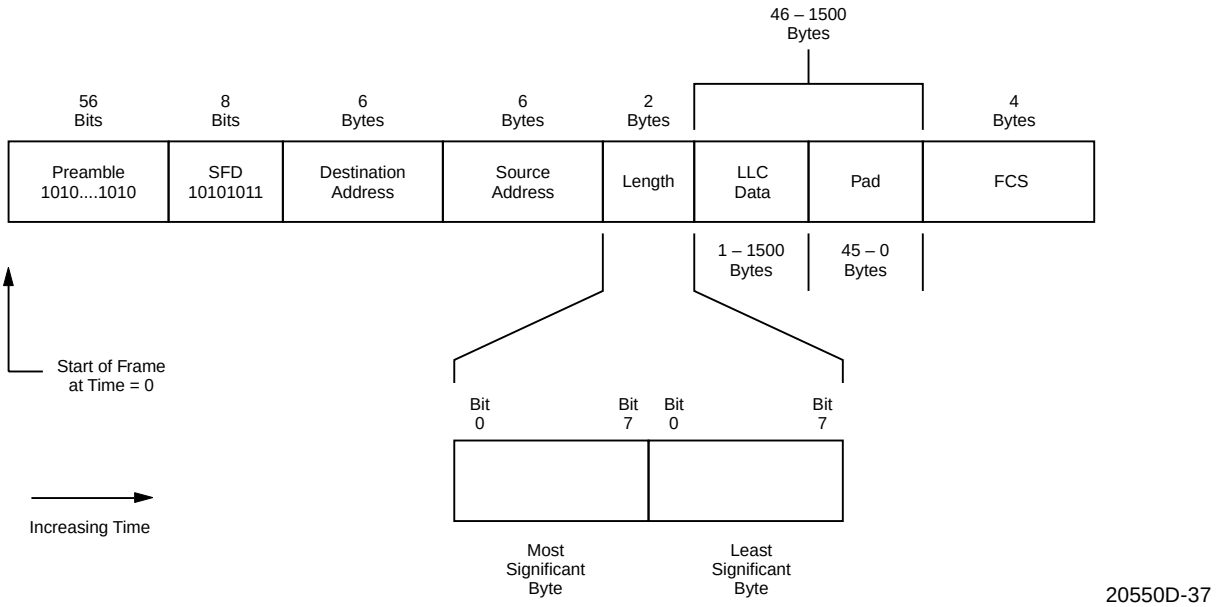


Figure 34. IEEE 802.3 Frame And Length Field Transmission Order

Since any valid Ethernet Type field value will always be greater than a normal IEEE 802.3 Length field (≥46), the Am79C971 controller will not attempt to strip valid Ethernet frames. *Note that for some network protocols, the value passed in the Ethernet Type and/or IEEE 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 Am79C971 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, i.e., those conditions 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 Am79C971 controller are basically collisions within the slot time and automatic runt packet rejection. The Am79C971 controller will ensure that collisions that 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 that 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 1. 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.

GPSI Loopback Modes

When GPSI is the active network port, there are only two modes of loopback operation: internal and external loopback. The settings of MENDECL and TMAULOOP have no effect for this port.

When INTL is set to 1, internal loopback is selected. Data coming out of the transmit FIFO is fed directly to the receive FIFO. All GPSI outputs are inactive; inputs are ignored.

External loopback operation is selected by setting INTL to 0. Data is transmitted to the network and is expected to be looped back to the GPSI receive pins outside the chip. Collision detection is active in this mode.

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 1, 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 1 and MENDECL is cleared to 0, 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 0. The programming of MENDECL has no effect in this mode. The AUI transmitter is enabled and data is transmitted to the network. The Am79C971 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 1, 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 1 and MENDECL is cleared to 0, 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 Am79C971 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 (with pin 1 jumpered to pin 3 and pin 2 jumpered to pin 6) that connects the transmit pins of the RJ-45 jack to its receive pins. When INTL is cleared to 0 and TMAULOOP is set to 1, 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 Am79C971 Am79C971 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 0. 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.

Media Independent Interface Loopback Features

Loopback through the MII can be handled in two ways. The Am79C971 controller supports an internal MII loopback and an external MII loopback. The MII loopback is completely separate from other network port loopback and requires that the other loopback modes be disengaged while the MII loopback is running. Further, the MII loopback requires that the MII port be manually configured through software using ASEL (BCR 2, bit 1) and PORTSEL (CSR 15, bits 8-7).

The external loopback through the MII requires a two-step operation. The external PHY must be placed into a loopback mode by writing to the MII Control Register (BCR33, BCR34). Then the Am79C971 controller must be placed into an external loopback mode. All other loopback modes have no meaning during MII operation.

The internal loopback through the MII is controlled by MIIILP (BCR32, bit 1). When set to 1, this bit will cause the internal portion of the MII data port to loopback on itself. The MII management port (MDC, MDIO) is unaffected by the MIILP bit. The internal MII interface is mapped in the following way:

- The TXD[3:0] nibble data path is looped back onto the RXD[3:0] nibble data path;
- TX_CLK is looped back as RX_CLK;
- TX_EN is looped back as RX_DV.
- CRS is correctly OR'd with TX_EN and RX_DV and always encompasses the transmit frame.
- TX_ER is not driven by the Am79C971 and therefore not looped back.

During the internal loopback, the TXD, TX_CLK, and TX_EN pins will toggle appropriately with the correct data.

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 except in the MII loopback mode. 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 Am79C971 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. On receive, the Am79C971 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 1. The FCS generator on the transmit side can still be disabled by setting DXMTFCS (CSR15, bit 3) to 1.

In internal loopback operation, the Am79C971 controller provides a special mode to test the collision logic. When FCOLL (CSR15, bit 4) is set to 1, a collision is forced during every transmission attempt. This will result in a Retry error.

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 Am79C971 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 (Table 7) 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. Internal Equivalent Series Resistance			35	ohm
7. Shunt Capacitance			7	pF

Note:
****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±. See Table 8.

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 Am79C971 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 0 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 Am79C971 controller that there is information on the receive pair and to separate the incoming Manchester encoded data stream into clock and NRZ data.

The receiver section consists of two parallel paths (see Figure 35). The receive data path is a zero threshold, wide bandwidth line receiver. The carrier path is an offset threshold, bandpass detecting line receiver. Both receivers share common bias networks to allow operation over a wide input common mode range.

Input Signal Conditioning

Transient noise pulses at the input data stream are rejected by the Noise Rejection Filter. Pulse width rejection is proportional to transmit data rate.

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 0) after IRXEN is asserted interrupts the receive oscillator. The oscillator is then restarted at the second Manchester 0 (bit time 4) and is phase locked to it. As a result, the MENDEC acquires the clock from the incoming Manchester bit pattern in 4 bit times with a 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 Am79C971 controller sees the first IRXCLK transition. This also strobes in the incoming fifth bit to the MENDEC as Manchester 1. IRXDAT may make a transition after the IRXCLK rising edge in bit cell 5, but its state is still undefined. The Manchester 1 at bit 5 is clocked to IRXDAT output at 1/4 bit time in bit cell 6.

PLL Tracking

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

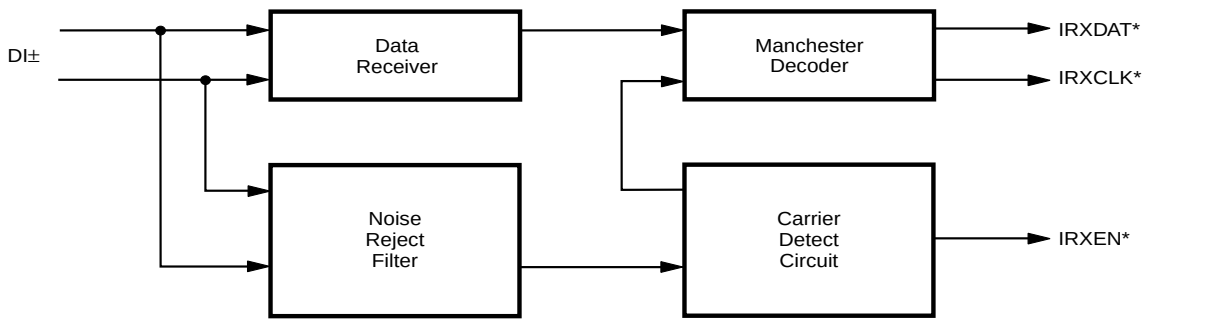


Figure 35. Receiver Block Diagram

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 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.*

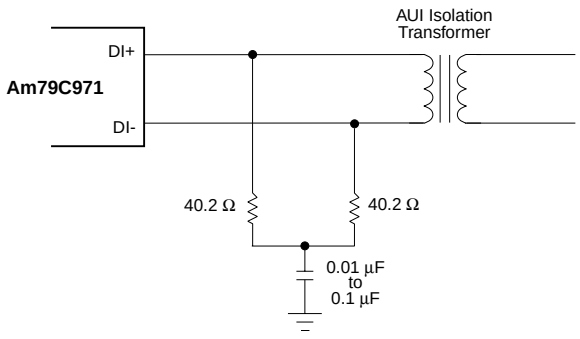
Attachment Unit Interface

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 Am79C971 controller is fully compliant to Section 7 of ISO 8802-3 (ANSI/IEEE 802.3) standard.

After the Am79C971 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 Figure 36. 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.



20550D-39

Figure 36. 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 MAU functions for the Twisted Pair Medium as specified by the supplement to the 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 0. When the LRT bit is set to 1, 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-meter (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, being 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 transmission, 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 $\overline{\text{LED0}}$ 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 it 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 to 6 consecutive link beat pulses of identical polarity. On entry to the Link Pass state, the polarity of the last five 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 to 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 to 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-MAU TXD± 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 *Power Savings Modes* section 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 ($\overline{\text{LED0}}$, 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 to 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

Figure 37 shows the proper 10BASE-T network interface design. Refer to the *PCnet Family Board Design and Layout Recommendations Application Note (PID #19595A)* for more design details. Also, refer to *Appendix A, Am79C971 Compatible Media Interface Modules* for a list of compatible 10BASE-T filter/transformer modules.

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

General Purpose Serial Interface

The General Purpose Serial Interface (GPSI) provides a direct interface to the MAC section of the Am79C971 controller. All signals are digital and data is non-encoded. The GPSI allows use of an external Manchester encoder/decoder such as the Am7992B Serial Interface Adapter (SIA). In addition, it allows the Am79C971 controller to be used as a MAC sublayer engine in repeater designs based on the IMR+ device (Am79C981).

GPSI mode is invoked by selecting the interface through the PORTSEL bits of the Mode register (CSR15, bits 8-7).

The GPSI interface uses some of the same pins as the interface to the MII. Simultaneous use of both functions is not possible.

After an H_RESET, all MII pins are internally configured to function as the MII interface. When the GPSI interface is selected by setting PORTSEL (CSR15, bits 8-7) to 10b, the Am79C971 controller will terminate all further accesses to the MII.

GPSI signal functions are described in the pin description section under the GPSI subheading.

Note that the XTAL1 input must always be driven with a clock source, even if GPSI mode is to be used. It is not necessary for the XTAL1 clock to meet the normal frequency and stability requirements in this case. Any frequency between 8 MHz and 20 MHz is acceptable. However, voltage drive requirements do not change. When GPSI mode is used, XTAL1 must be driven for several reasons:

The default H_RESET configuration for the Am79C971 controller is AUI port selected. Until GPSI mode is selected, the XTAL1 clock is needed for some internal operations (namely, RESET). The XTAL1 clock drives the EEPROM read operation, regardless of the network mode selected.

The XTAL1 clock determines the length of the internal S_RESET caused by the read of the Reset register, regardless of the network mode.

Note: *If a clock slower than 20 MHz is provided at the XTAL1 input, the time needed for EEPROM read and the internal S_RESET will increase.*

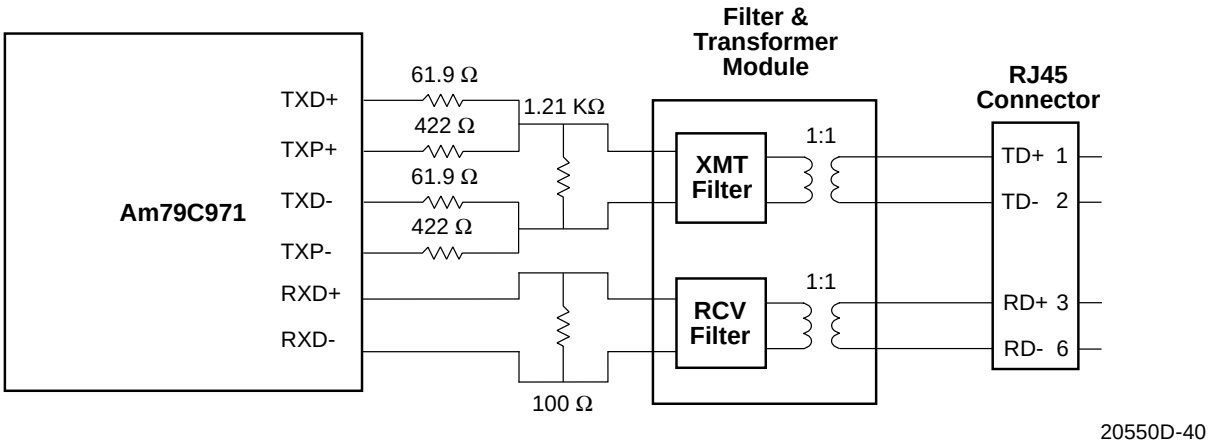


Figure 37. 10BASE-T Interface Connection

Full-Duplex Operation

The Am79C971 controller supports full-duplex operation on all four network interfaces: AUI, GPSI, 10BASE-T, and MII. 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, TXDAT and RXDAT pins of the GPSI port, and the TXD[3:0] and RXD[3:0] pins of the MII port. Full-duplex operation is enabled by the FDEN and AUIFD bits located in BCR9 for all ports. Full-duplex operation is enabled through Auto-Negotiation when DANAS (BCR 32, bit 7) is not enabled on the MII port or when Auto-Negotiation is running on the internal PHY.

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 preserved in the Transmit FIFO during transmission of the first 512 bits as described in the Transmit Exception Conditions section. Instead, when full-duplex mode is active and a frame is being

transmitted, the XMTFW bits (CSR80, bits 9-8) always govern when transmit DMA is requested.

- Successful reception of the first 64 bytes of every receive frame is not a requirement for Receive DMA to begin as described in the Receive Exception Conditions section. Instead, receive DMA will be requested as soon as either the RCVFW threshold (CSR80, bits 12-13) is reached or a complete valid receive frame is 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.

The MAC engine changes for full-duplex operation are as follows:

- Changes to the Transmit Deferral mechanism:
  - — Transmission is not deferred while receive is active.
  - — The 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 or MII 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.

The T-MAU changes for full-duplex operation are as follows:

- The transmit to receive loopback path in the T-MAU is disabled.
- The collision detect circuit is disabled.
- The SQE test function is disabled.

The MII changes for full-duplex operation are as follows:

- The collision detect (COL) pin is disabled.
- The SQE test function is disabled.

Full-Duplex Link Status LED Support

The Am79C971 controller provides bits 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 1 will be sent to the associated LEDOUT bit when the T-MAU is in the Full-Duplex Link Pass state only.

Media Independent Interface

The Am79C971 controller fully supports the MII according to the IEEE 802.3 standard. This Reconciliation Sublayer interface allows a variety of PHYs (100BASE-TX, 100BASE-FX, 100BASE-T4, 100BASE-T2, 10BASE-T, etc.) to be attached to the Am79C971 MAC engine without future upgrade problems. The MII interface is a 4-bit (nibble) wide data path interface that runs at 25 MHz for 100-Mbps networks and 2.5 MHz for 10-Mbps networks. The interface consists of two independent data paths, receive (RXD(3:0)) and transmit (TXD(3:0)), control signals for each data path (RX_ER, RX_DV, TX_ER, TX_EN), network status signals (COL, CRS), clocks (RX_CLK, TX_CLK) for each data path, and a two-wire management interface (MDC and MDIO). See Figure 38.

MII Transmit Interface

The MII transmit clock is generated by the external PHY and is sent to the Am79C971 controller on the TX_CLK input pin. The clock can run at 25 MHz or 2.5 MHz, depending on the speed of the network that the external PHY is attached to. The data is a nibble-wide (4 bits) data path, TXD(3:0), from the Am79C971 controller to the external PHY and is synchronous to the rising edge of TX_CLK. The transmit process starts when the Am79C971 controller asserts the TX_EN, which indicates to the external PHY that the data on TXD(3:0) is valid.

Normally, unrecoverable errors are signaled through the MII to the external PHY with the TX_ER output pin. The external PHY will respond to this error by generating a TX coding error on the current transmitted frame. The Am79C971 controller does not use this method of signaling errors on the transmit side. The Am79C971 controller will invert the FCS on the last byte generating an invalid FCS. The TX_ER pin is reserved for future use and is actively driven to 0.

MII Receive Interface

The MII receive clock is also generated by the external PHY and is sent to the Am79C971 controller on the RX_CLK input pin. The clock will be the same frequency as the TX_CLK but will be out of phase and can run at 25 MHz or 2.5 MHz, depending on the speed of the network the external PHY is attached to. The RX_CLK is a continuous clock during the reception of the frame, but can be stopped for up to two RX_CLK periods at the beginning and the end of frames, so that the external PHY can sync up to the network data traffic necessary to recover the receive clock. During this time, the external PHY may switch to the TX_CLK to maintain a stable clock on the receive interface. The Am79C971 controller will handle this situation with no loss of data. The data is a nibble-wide (4 bits) data

path, RXD(3:0), from the external PHY to the Am79C971 controller and is synchronous to the rising edge of RX_CLK.

The receive process starts when RX_DV is asserted. RX_DV will remain asserted until the end of the receive frame. The Am79C971 controller requires CRS (Carrier Sense) to toggle in between frames in order to receive them properly. Errors in the currently received frame are signaled across the MII by the RX_ER pin. RX_ER can be used to signal special conditions *out of band* when RX_DV is not asserted. Two defined out-of-band conditions for this are the 100BASE-TX signaling of *bad* Start of Frame Delimiter and the 100BASE-T4 indication of illegal code group before the receiver has *synched* to the incoming data. The Am79C971 controller will not respond to these conditions. All *out of band* conditions are currently treated as NULL events. Certain *in band* non-IEEE 802.3u-compliant flow control sequences may cause erratic behavior for the Am79C971 controller. Consult the switch/bridge/router/hub manual to disable the *in-band* flow control sequences if they are being used.

MII Network Status Interface

The MII also provides signals that are consistent and necessary for IEEE 802.3 and IEEE 802.3u operation. These signals are CRS (Carrier Sense) and COL (Collision Sense). Carrier Sense is used to detect non-idle activity on the network. Collision Sense is used to indicate that simultaneous transmission has occurred in a half-duplex network.

MII Management Interface

The MII provides a two-wire management interface so that the Am79C971 controller can control and receive status from external PHY devices.

The Am79C971 controller can support up to 31 external PHYs attached to the MII Management Interface with software support and only one such device without software support.

The Network Port Manager copies the PHYAD after the Am79C971 controller reads the EEPROM and uses it to communicate with the external PHY. The PHY address must be programmed into the EEPROM prior to starting the Am79C971 controller. This is necessary so that the internal management controller can work autonomously from the software driver and can always know where to access the external PHY. The Am79C971 controller is unique by offering direct hardware support of the external PHY device without software support. The internal PHY is addressed at the last available MII address of 1Fh. To access the 31 external PHYs, the software driver must have knowledge of the external PHY's address when multiple PHYs are present before attempting to address it.

The MII Management Interface uses the MII Control, Address, and Data registers (BCR32, 33, 34) to control and communicate to the external and internal 10BASE-T only PHYs. Am79C971 generates MII management frames to the external PHY through the MDIO pin synchronous to the rising edge of the Management Data Clock (MDC) based on a combination of writes and reads to these registers. To prevent problems on the exposed interface, MII management frames will not be generated when the internal PHY is the target. The MII only supports internal and external 10BASE-T or 100BASE-T as possible network connections. The internal AUI and GPSI are not considered part of the MII and cannot be selected through the MII.

.

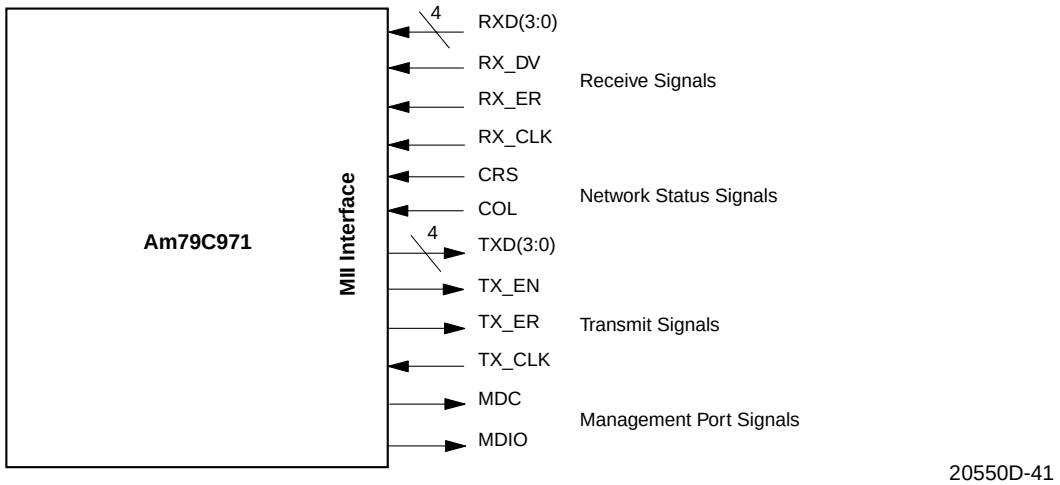


Figure 38. Media Independent Interface

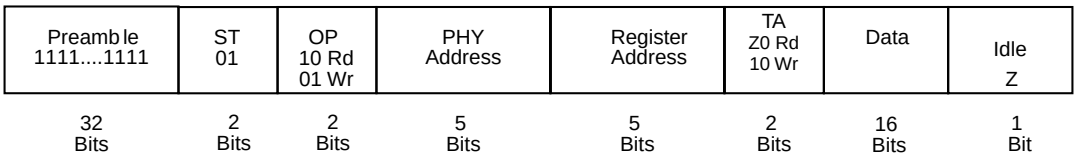
The MII Management Interface has a built-in detection system to allow the Am79C971 controller to determine if an external PHY is attached. The MDIO I/O pin has a resistor network between the Am79C971 controller and the external PHY that will assert a static 1 when connected. If there is no external PHY connected, the resistor network will drive a static zero. This information is signaled by the interrupt MPDTINT (CSR7, MIIPDTI, bit 1), and the status is provided by reading the Media Independent Interface PHY Detected (MIIPD) (BCR32, bit 14). This resistor network is only required on an exposed MII connector.

MII Management Frames

MII management frames are automatically generated by the Am79C971 controller and conform to the MII clause in the IEEE 802.3u standard.

The start of the frame is a preamble of 32 ones and guarantees that all of the external PHYs are synchronized on the same interface. (See Figure 39.) Loss of synchronization is possible due to the *hot-plugging* capability of the exposed MII.

The IEEE 802.3 specification allows you to drop the preamble, if after reading the MII Status Register from the external PHY you can determine that the external PHY will support Preamble Suppression (BCR34, bit 6). After having a valid MII Status Register read, the Am79C971 controller will then drop the creation of the preamble stream until a reset occurs, receives a read error, or the external PHY is disconnected.



20550D-42

Figure 39. Frame Format at the MII Interface Connection

This is followed by a start field (ST) and an operation field (OP). The operation field (OP) indicates whether the Am79C971 controller is initiating a read or write operation. This is followed by the external PHY address (PHYAD) and the register address (REGAD) programmed in BCR33. The internal PHYAD is at location 1Fh and the internal register address space REGAD is 00h - 08h. The external PHY may have a larger address space starting at 10h - 1Fh. This is the address range set aside by the IEEE as vendor usable address space and will vary from vendor to vendor. This field is followed by a bus turnaround field. During a read operation, the bus turnaround field is used to determine if the external PHY is responding correctly to the read request or not. The Am79C971 controller will tri-state the MDIO for both MDC cycles. During the second cycle, if the external PHY is synchronized to the Am79C971 controller, the external PHY will drive a 0. If the external PHY does not drive a 0, the Am79C971 controller will signal a MREINT (CSR7, bit 9) interrupt, if MREINTE (CSR7, bit 8) is set to a 1, indicating the Am79C971 controller had an MII management frame read error and that the data in BCR34 is not valid. The data field to/from the internal or external PHY is read or written into the BCR34 register. The last field is an IDLE field that is necessary to give ample time for drivers to turn off before the next access. The Am79C971 controller will drive the MDC to 0 and tri-state the MDIO anytime the MII Management Port is not active.

To help to speed up the reading and writing of the MII management frames to the external PHY, the MDC can be sped up to 10 MHz by setting the FMDC bits in BCR32. The IEEE 802.3 specification requires use of the 2.5-MHz clock rate, but 5 MHz and 10 MHz are available for the user. The intended applications are that the 10-MHz clock rate can be used for a single external PHY on an adapter card or motherboard. The 5-MHz clock rate can be used for an exposed MII with one external PHY attached. The 2.5-MHz clock rate is intended to be used when multiple external PHYs are connected to the MII Management Port or if compliance to the IEEE 802.3u standard is required.

Auto-Poll External PHY Status Polling

As defined in the IEEE 802.3 standard, the external PHY attached to the Am79C971 controller's MII has no way of communicating important timely status information back to Am79C971 controller. The Am79C971 controller has no way of knowing that an external PHY has undergone a change in status without polling the MII status register. To prevent problems from occurring with inadequate host or software polling, the Am79C971 controller will Auto-Poll when APEP (BCR32, bit 11) is set to 1 to insure that the most current information is available. See *Appendix E, Auto Ne-*

gotiation Registers, for the bit descriptions of the MII Status Register. The contents of the latest read from the external PHY will be stored in a shadow register in the Auto-Poll block. The first read of the MII Status Register will just be stored, but subsequent reads will be compared to the contents already stored in the shadow register. If there has been a change in the contents of the MII Status Register, a MAPINT (CSR7, bit 7) interrupt will be generated on INTA if the MAPINTE (CSR7, bit 6) is set to 1. The Auto-Poll features can be disabled if software driver polling is required.

The Auto-Poll's frequency of generating MII management frames can be adjusted by setting of the APDW bits (BCR32, bits 10-8). The delay can be adjusted from 0 MDC periods to 2048 MDC periods. Auto-Poll by default will only read the MII Status register in the external PHY.

Network Port Manager

The Am79C971 controller is unique in that it does not require software intervention to control and configure an external PHY attached to the MII. This was done to ensure backwards compatibility with existing software drivers. To the current software drivers, the Am79C971 controller will look and act like the PCnet-PCI II and will interoperate with existing PCnet drivers from revision 2.5 upward. The heart of this system is the Network Port Manager, which acts as an arbiter between all of the possible automatically controllable physical connections, including the external PHY and the internal 10BASE-T/AUI ports. See the section on *Automatic Network Port Selection* for more details.

If the external PHY is present and is active, the Network Port Manager will request status from the external PHY by generating MII management frames. These frames will be sent roughly every 900 ms. These frames are necessary so that the Network Port Manager can monitor the current active link and can select a different network port if the current link goes down.

Auto-Negotiation

The Am79C971 controller implements the Auto-Negotiation portion of the IEEE 802.3u specification for the 10BASE-T MAU. Auto-Negotiation attempts to automatically configure the link between two link partners.

To accomplish this, the 10BASE-T MAU can send a new link pulse train called Fast Link Pulses. These Fast Link Pulses replace the current 10BASE-T Link Pulse. The Fast Link Pulse are made up of a train of 17 clocks alternating with 16 data fields for a total of 33 pulses. The two link partners will send information in those 16 data positions between themselves. The primary information sent is called the Base Code Link Word. See *Appendix E, Auto Negotiation Registers,* for details on the Auto-Negotiation Registers. The Am79C971 controller will send in its Base Code Link Word the capabilities of the internal PHY. The internal PHY is capable of half- or full-duplex 10BASE-T. Through the external PHY, the following capabilities are possible: 100BASE-T4, 100BASE-TX Full-/Half-Duplex, and 10BASE-T Full-/Half-Duplex. The capabilities are then sent to a link partner that will also send its capabilities. Both sides look to see what is possible and then they will connect at the greatest possible speed and capability according to the following table as defined in the IEEE 802.3u standard.

Table 10. Auto-Negotiation Capabilities

Network Speed	Physical Network Type
200 Mbps	100BASE-X, Full Duplex
100 Mbps	100BASE-T4, Half Duplex
100 Mbps	100BASE-X, Half Duplex
20 Mbps	10BASE-T, Full Duplex
10 Mbps	10BASE-T, Half Duplex

By default, the link partner must be at least 10BASE-T half-duplex capable. The Am79C971 controller can automatically negotiate either internally or externally with the network and yield the highest performance possible without software support. See the section on *Network Port Manager* for more details.

Auto-Negotiation goes further by providing a message-based communication scheme called, *Next Pages*, before connecting to the Link Partner. *This feature is not supported in Am79C971 unless the DANAS (BCR32, bit 10) is selected and the software driver is capable of controlling the internal or external PHY.*

A complete bit description of the MII and Auto-Negotiation registers can be found in Appendix E.

Automatic Network Port Selection

The Am79C971 controller extends the PCnet-PCI II device's automatic network port selection by adding the MII port to the already existing 10BASE-T, and AUI ports. If ASEL (BCR2, bit 0) is set to 1 and DANAS (BCR 32, bit 7) is set to 0, then the Network Port Manager will start to configure the external PHY if it detects the external PHY on the MII Interface. If the external PHY is not responding, the Network Port Manager will try to resolve problems and to fail non-responding links in a graceful manner, utilizing a large timer on the Am79C971 controller to time-out links.

Automatic Network Selection: Exceptions

If ASEL (BCR2, bit 0) is set to 0 or DANAS (BCR 32, bit 7) is set to 1, then the Network Port Manager will discontinue actively trying to establish the connections. It is assumed that the software driver is attempting to

configure the network port and the Am79C971 controller will always defer to the software driver. When The ASEL is set to 0, the software driver should then configure the ports with PORTSEL (CSR15, bits 7-8). The GPSI does not participate in the automatic selection process and should be manually configured with the PORTSEL bits. If FDEN (BCR9, bit 0) is set to 1 or DLNKST (CSR15, bit 12) is set to 1, the Network Port Manager will continue to select the active network port, but the internal T-MAU will not auto-negotiate the network port. Instead, if FDEN (BCR9, bit 0) is set to 1, the internal T-MAU will come up as a full-duplex T-MAU port if link beats are found. If DLNKST (CSR15, bit 12) is set to 1 and the internal T-MAU is active, the T-MAU will be in a link pass state regardless of link beat.

Note: *It is highly recommended that ASEL and PORTSEL be used when trying to manually configure a specific network port.*

In order to manually configure the External PHY, the recommended procedure is to force the PHY configurations when Auto-Negotiation is *not* enabled. Set the DANAS bit (BCR32, bit 7) to turn off the Network Port Manager. Then write again to BCR32 with the DANAS and XPHYANE (BCR32, bit 5) bits cleared, together with the XPHYFD (BCR32, bit 4) and XPHYSP (BCR32, bit 3) bits set to the desired configuration. The Network Port Manager will send a few frames to validate the configuration.

If FCON (BCR32, bit 0) is set to 1, this bit will force the internal Network Port Manager into Fast Configuration Mode. During this mode, the Network Port Manager will not attempt to start Auto-Negotiation on the internal as well as the external PHY. Instead, it will rely on link integrity tests for link pass state. This will accelerate the automatic port selection on the Am79C971 controller. The Network Port Manager in Fast Configuration Mode will start with the external PHY if one is detected. If the link does not come up, the Network Port Manager will enable the internal 10BASE-T MAU. If the internal port also does not come up, the Network Port Manager will continue to search the MII and the internal T-MAU ports while enabling the internal AUI port. The FCON (BCR32, bit 0) should only be used if the network is experiencing difficulty and is not stable.

CAUTION: *The Network Port Manager utilizes the PHYADD (BCR33, bits 9-5) to communicate with the external PHY during the automatic port selection process. The PHYADD is copied into a shadow register after the Am79C971 controller has read the configuration information from the EEPROM. Extreme care must be exercised by the host software not to access BCR33 during this time. A read of PVALID (BCR19, bit 15) before accessing BCR33 will guarantee that the PHYADD has been shadowed.*

Am79C972's Automatic Network Port selection mechanism falls within the following three general categories:

- External PHY Not Present
- External PHY Present but *Not* Auto-Negotiable
- External PHY Present and Auto-Negotiable

Automatic Network Selection: External PHY Not Present

The first case occurs when the MIIPD (BCR32, bit 14) bit is 0. This indicates that there is no external PHY attached to Am79C971 controller's MII. The Am79C971 controller's Network Port Manager will start the internal Auto-Negotiation 10BASE-T MAU. If the Auto-Negotiation 10BASE-T MAU fails to respond within a specific time frame, then the Am79C971 controller will enable the AUI. Auto-Negotiation FAST Link Pulses are still being sent and the Auto-Negotiation 10BASE-T MAU is still running. At that point, the active link can switch back to the 10BASE-T MAU when Link Pulses or FAST Link Pulses are detected. The only way to disable the Auto-Negotiation process without using FDEN or DLNKST is to enable the DANAS (BCR32, bit 7) bit or to write to the internal/external PHYs MII control register and disable Auto-Negotiation when the internal Network Port Manager is disabled.

Automatic Network Selection: External PHY Present but *Not* Auto-Negotiable

The second case occurs when the MIIPD (BCR32, bit 14) bit is 1. This indicates that there is an external PHY attached to Am79C971 controller's MII. If more than one external PHY is attached to the MII Management Interface, then the DANAS (BCR32, bit 7) bit must be set to 1 and then all configuration control should revert to software. The Am79C971 controller will read the register of the external PHY to determine its status and network capabilities. See *Appendix E, Auto Negotiation Registers,* for the bit descriptions of the MII Status register. If the external PHY is not Auto-Negotiation capable and/or the XPHYANE (BCR32, bit 5) bit is set to 0, then the Network Port Manager will match up the external PHY capabilities with the XPHYFD (BCR 32, bit 4) and the XPHYSP (BCR32, bit 3) bits programmed from the EEPROM. The Am79C971 controller will then program the external PHY with those values. A new read of the external PHYs MII Status register will be made to see if the link is up. If the link does not come up as programmed after a specific time, the Am79C971 controller will fail the external PHY link and start the internal PHY process as described above. The Am79C971 controller will only start the external PHY link if the internal link has failed. If both links have failed, the AUI is enabled but the Network Port Manager will still query the internal and external PHYs for active links.

Automatic Network Selection: External PHY Present and Auto-Negotiable

The third case occurs when the MIIPD (BCR32, bit 14) bit is 1. This indicates that there is an external PHY attached to Am79C971 controller's MII. If more than one

external PHY is attached to the MII Management Interface, then the DANAS (BCR32, bit 7) bit must be set to 1 and then all configuration control should revert to software. The Am79C971 controller will read the MII Status register of the external PHY to determine its status and network capabilities. See Appendix E for the bit descriptions of the MII Status register. If the external PHY is Auto-Negotiation capable and/or the XPHYANE (BCR32, bit 5) bit is set to 1, then the Am79C971 controller will start the external PHY's Auto-Negotiation process. The Am79C971 controller will write to the external PHY's Advertisement register with the following conditions set: turn off the Next Pages support, set the Technology Ability Field (See Appendix E for the Auto-Negotiation register bit descriptions) from the external PHY MII Status register read, and set the Type Selector field to the IEEE 802.3 standard. The Am79C971 controller will then write to the external PHY's MII Control register instructing the external PHY to negotiate the link. The Am79C971 controller will poll the external PHY's MII Status register until the Auto-Negotiation Complete bit is set to 1and the Link Status bit is set to 1. The Am79C971 controller will then wait a specific time and then again read the external PHY's MII Status register. If the Am79C971 controller sees that the external PHY's link is down, it will try to bring up the external PHY's link manually as described above. A new read of the external PHY's MII Status register will be made to see if the link is up. If the link does not come up as programmed after a specific time, the Am79C971 controller will fail the external PHY link and start the process again for the internal PHY. If the link has failed, the AUI is enabled, but the Network Port Manager will still query the external PHY for an active link.

Automatic Network Selection: Working with the Micro Linear 6692

The final case that occurs is the hybrid condition that does not fit neither the Auto-Negotiable case nor the Non-Auto-Negotiable case. An example of this case is the Micro Linear 6692 PHY. The Micro Linear 6692 PHY masquerades as an Auto-Negotiable PHY by providing Auto-Negotiation capabilities, but does not provide the 10BASE-T MAU. It relies on the MAC controller, the Am79C971 controller in this case, to provide the 10BASE-T MAU for it. The Network Port Manager handles this condition virtually the same way as the Auto-Negotiable case, except for the final handshake that enables the internal 10BASE-T MAU. After the 6692 negotiates for the 10BASE-T MAU, it monitors the link for Normal LInk Pulses (NLPs). If it sees the NLPs, then it will report that it completed the Auto-Negotiation process. The Am79C971 controller will read the MII and Auto-Negotiation registers to figure out which port has been negotiated. At this point, the Network Port Manager will enable the internal 10BASE-T MAU, if that port has been negotiated, and complete the first part of the handshake. The final part of the handshake is to prevent the 6692 from renegotiating the link without the Network Port Manager's knowledge. Connecting the $\overline{\text{LED0}}$ pin to the 10BTRCV pin of the 6692 will accomplish this. The $\overline{\text{LED0}}$ reports the link status from the internal TMAU. The Network Port Manager monitors the internal link status, and knowing when the 6692 will start to renegotiate the link, it will stay in synchronization with the 6692.

Automatic Network Selection: Force External Reset

If the XPHYRST bit (BCR32, bit 6) is set to 1, then the external case flow changes slightly. The Am79C971 controller will write to the external PHY's MII Control register with the RESET bit set to 1 (See *Appendix E, Auto Negotiation Registers,* for the MII register bit descriptions). This will force a complete reset of the external PHY. The Am79C971 controller after a specific time will poll the external PHY's MII Control register to see if the RESET bit is 0. After the RESET bit is cleared, then the normal flow continues.

External Address Detection Interface (EADI)

The EADI is provided to allow external address filtering and to provide a Receive Frame Tag word for proprietary routing information. It is selected by setting the EADISEL bit in BCR2 to 1. This feature is typically utilized by 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 Am79C971 controller, to compare the captured address with a table of stored addresses or identifiers, and then to determine whether or not the Am79C971 controller should accept the packet.

External Address Detection Interface: Internal PHY

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 Am79C971 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 1s 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 Am79C971 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 or GPSI port is selected, transitions on SRD will occur during receive activity.

The $\overline{EAR}$ 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 Am79C971 controller or the frame is of the type 'Broadcast', then the frame will be accepted regardless of the condition of $\overline{EAR}$. When the EADISEL bit of BCR2 is set to 1 and the Am79C971 controller is programmed to promiscuous mode (PROM bit of the Mode Register is set to 1), then all incoming frames will be accepted, regardless of any activity on the $\overline{EAR}$ pin.

Internal address match is disabled when PROM (CSR15, bit 15) is cleared to 0, DRCVBC (CSR15, bit 14) and DRCVPA (CSR15, bit 13) are set to 1, and the Logical Address Filter registers (CSR8 to CSR11) are programmed to all zeros.

When the EADISEL bit of BCR2 is set to 1 and internal address match is disabled, then all incoming frames will be accepted by the Am79C971 controller, unless the $\overline{EAR}$ 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 $\overline{EAR}$ signal, assuming that the Am79C971 controller is not configured to accept runt packets. The EADI logic only samples $\overline{EAR}$ from 2 bit times after SFD until 512 bit times (64 bytes) after SFD. The frame will be accepted if $\overline{EAR}$ has not been asserted during this window. If Runt Packet Accept (CSR124, bit 3) is enabled, then the $\overline{EAR}$ signal must be generated prior to the 8 bytes received, 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. $\overline{EAR}$ must have a pulse width of at least 110 ns.

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, internetworking 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 Am79C971 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. Table 11 summarizes the operation of the EADI interface.

Table 11. 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 0 bits after SFD to 512 bits after SFD	Am79C971 controller internal physical address and logical address filter matches and broadcast frames

External Address Detection Interface: External PHY

When using the MII, the EADI interface changes to reflect the changes on that interface. Except for the notations below the interface conforms to the previous functionality. The data arrives in nibbles and can be at a rate of 25 MHz or 2.5 MHz.

The MII provides all necessary data and clock signals needed for the EADI interface. Consequently, SRDCLK and SRD are not used and are driven to 0. Data for the EADI is the RXD(3:0) receive data provided to the MII. Instead of deserializing the network data, the user will receive the data as 4 bit nibbles. RX_CLK is provided to allow clocking of the RXD(3:0) receive nibble stream into the external address detection logic. The RXD(3:0) data is synchronous to the rising edge of the RX_CLK.

The assertion of SFBD is a signal to the external address detection logic that the SFD has been detected and that the first valid data nibble is on the RXD(3:0) data bus. The SFBD signal is delayed one RX_CLK cycle from the above definition and actually signals the start of valid data. In order to reduce the amount of logic external to the Am79C971 controller for multiple

address decoding systems, the SFBD signal will go HIGH at each new byte boundary within the packet, subsequent to the SFD. This eliminates the need for externally supplying byte framing logic.

The $\overline{\text{EAR}}$ pin function is the same and should be driven LOW by the external address comparison logic to reject a frame.

External Address Detection Interface: Receive Frame Tagging

The Am79C971 controller supports receive frame tagging in both internal PHY mode or in the MII mode. The method remains constant, but the chip interface pins will change between the MII and the internal PHY modes. The receive frame tagging implementation will be a two- and three-wire chip interface, respectively, added to the existing EADI.

The Am79C971 controller supports up to 15 bits of receive frame tagging per frame in the receive frame status (RFRTAG). The RFRTAG bits are in the receive frame status field in RMD2 (bits 30-16) in 32-bit software mode. The receive frame tagging is not supported in the 16-bit software mode. The RFRTAG field are all zeros when either the EADISEL (BCR2, bit3) or the RXFRTAG (CSR7, bit 14) are set to 0. When EADISEL (BCR2, bit 3) and RXFRTAG (CSR7, bit 14) are set to 1, then the RFRTAG reflects the tag word shifted in during that receive frame.

In the MII mode, the two-wire interface will use the MIIRXFRTGD and MIIRXFRTGE pins from the EADI interface. These pins will provide the data input and data input enable for the receive frame tagging, respectively. These pins are normally not used during the MII operation.

In the internal PHY mode, the three-wire interface will use the RXFRTGD, SRDCLK, and the RXFRTGE pins from the EADI and MII. These pins will provide the data input, data input clock, and the data input for the receive frame tagging enable, respectively.

The receive frame tag register is a shift register that shifts data in MSB first, so that less than the 15 bits allocated may be utilized by the user. The upper bits not utilized will return zeros. The receive frame tag register is set to 0 in between reception of frames. After receiving SFBD indication on the EADI, the user can start shifting data into the receive tag register until one network clock period before the Am79C971 controller receives the end of the current receive frame.

In the MII mode, the user must see the RX_CLK to drive the synchronous receive frame tag data interface. After receiving the SFBD indication, sampled by the rising edge of the RX_CLK, the user will drive the data input and the data input enable synchronous with the rising edge of the RX_CLK. The user has until one network clock period before the deassertion of the RX_DV to input the data into the receive frame tag register. At the deassertion of the RX_DV, the receive frame tag register will no longer accept data from the two-wire interface. If the user is still driving the data input enable pin, erroneous or corrupted data may reside in the receive frame tag register. See Figure 40.

In the internal PHY mode, the user must use the recovered receive data clock driven on the SRDCLK pin to drive the synchronous receive frame tag data interface. After receiving the SFBD indication, sampled by the rising edge of the recovered receive data clock, the user will drive the data input and the data input enable synchronous with the rising edge of the recovered receive data clock. The user has until one network clock period before the deassertion of the data from the network to input the data into the receive frame tag register. At the completion of received network data, the receive frame tag register will no longer accept data from the two-wire interface. If the user is still driving the data input enable pin, erroneous or corrupted data may reside in the receive frame tag register. See Figure 41.

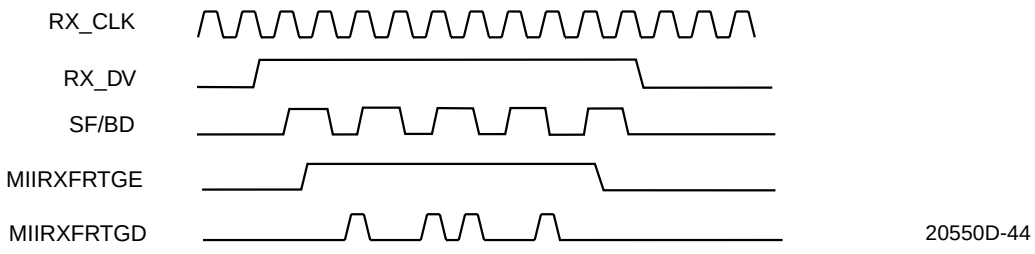
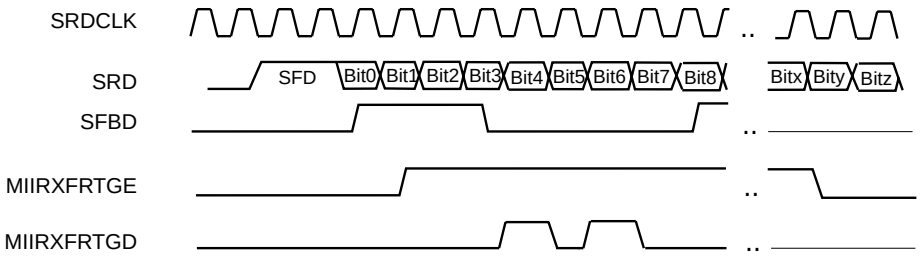


Figure 40. MII Receive Frame Tagging



20550D-44

Figure 41. Internal PHY Receive Frame Tagging

Expansion Bus Interface

The Am79C971 controller contains an Expansion Bus Interface that supports two different boot devices, EPROM and Flash, as well as SRAM used as an extension to the internal FIFOs to buffer packets. The Am79C971 controller supports Flash and EPROM devices as boot devices as well as providing read/write access to Flash or EPROM while the Am79C971 controller is in STOP or in SPND or when the SRAM SIZE bits (BCR25, bits 7-0) are set to 0. While in STOP, the Am79C971 controller provides read/write diagnostic access to SRAM (when present). This limitation on the SRAM diagnostic is necessary to prevent data corruption.

The signal AS_$\overline{\text{EBOE}}$ is provided to strobe the upper 8 bits of the address into an external ‘374 (D flip-flop) address latch. AS_$\overline{\text{EBOE}}$ is asserted LOW during EPROM/Flash read operations to control the $\overline{\text{OE}}$ input of the EPROM/Flash.

The Expansion Bus Address is split into two different buses, EBUA_EBA[7:0] and EBDA[15:8]. The EBUA_EBA[7:0] provides the least and the most significant address byte. When accessing SRAM and EPROM/Flash the EBUA_EBA[7:0] is strobed into an external ‘374 (D flip-flop) address latch. This constitutes the most significant portion of the Expansion Bus Address. For SRAM/EPROM/Flash accesses, EBUA_EBA[7:0] constitutes the remaining least significant address byte. For byte oriented EPROM/Flash accesses, EBDA[15:8] constitutes the upper or middle address byte. EBADDRU (BCR29, bits 3-0) should be set to 0 even when not used, since EBADDRU constitutes the EBUA portion of the EBUA EBA address byte and is strobed into the external’374 address latch.

The signal $\overline{\text{EROMCS}}$ is connected to the $\overline{\text{CS}}$/$\overline{\text{CE}}$ input of the EPROM/Flash. The signal $\overline{\text{ERAMCS}}$ is connected to the $\overline{\text{CE}}$/$\overline{\text{CS}}$ input of the SRAM. The signal $\overline{\text{EBWE}}$ is connected to the $\overline{\text{WE}}$ of the SRAM and Flash devices.

The Expansion Data Bus is configured for 16-bit word access during SRAM accesses and 8-bit byte access during EPROM/Flash accesses. During SRAM accesses, EBD[7:0] provides the lower data byte while EBDA[15:8] provides the upper data byte. During EPROM/Flash accesses, EBD[7:0] provides the data byte. See Figure 42.

Expansion ROM - Boot Device Access

The Am79C971 controller supports EPROM or Flash as an Expansion ROM boot device. Both are configured using the same methods and operate the same. See the previous section on Expansion ROM transfers to get the PCI timing and functional description of the transfer method. The Am79C971 controller is functionally equivalent to the PCnet-PCI II controller with Expansion ROM. See Figure 43 and Figure 44.

The Am79C971 controller will always read four bytes for every host Expansion ROM read access. The interface to the Expansion Bus runs synchronous to the PCI bus interface clock. The Am79C971 controller will start the read operation to the Expansion ROM by driving the upper 8 bits of the Expansion ROM address on EBUA_EBA[7:0]. One-half clock later, AS_$\overline{\text{EBOE}}$ goes high to allow registering of the upper address bits externally. The upper portion of the Expansion ROM address will be the same for all four byte read cycles. AS_$\overline{\text{EBOE}}$ is driven high for one-half clock, EBUA_EBA[7:0] are driven with the upper 8 bits of the Expansion ROM address for one more clock cycle after AS_$\overline{\text{EBOE}}$ goes low. Next, the Am79C971 controller starts driving the lower 8 bits of the Expansion ROM address on EBUA_EBA[7:0].

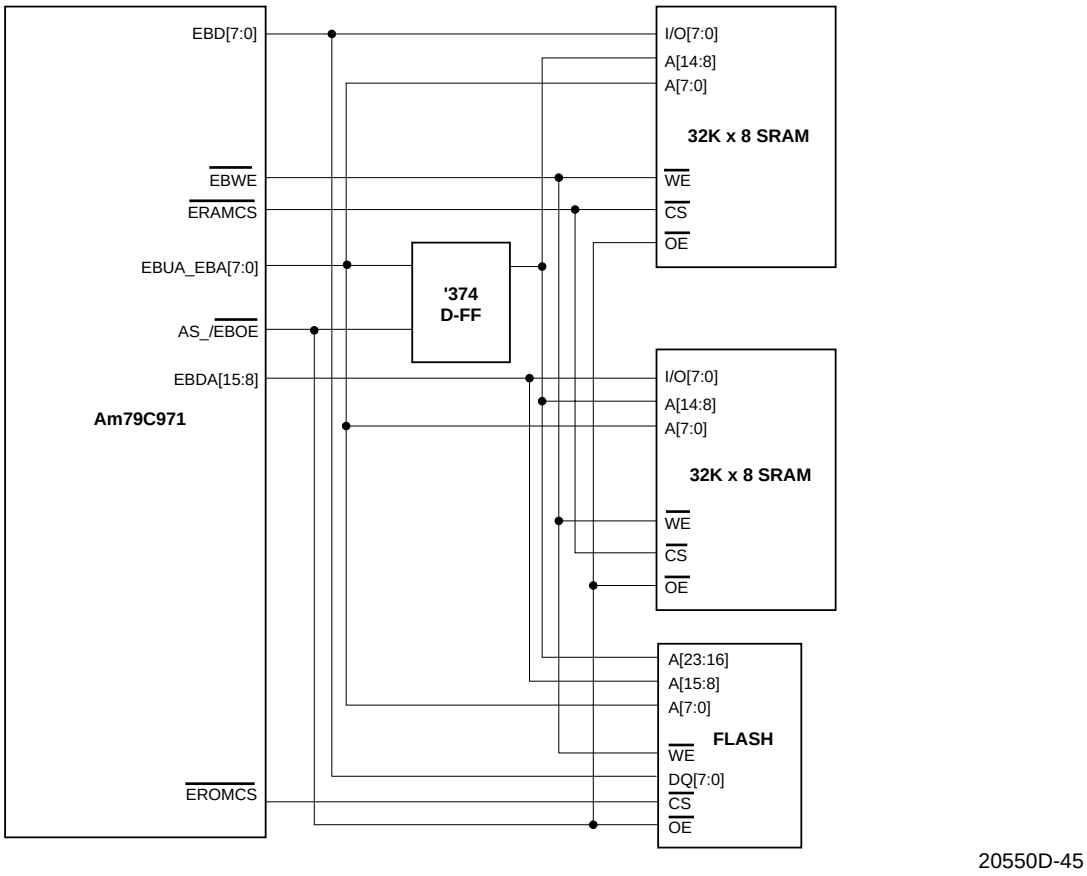


Figure 42. SRAM and Flash Configuration for the Expansion Bus

The time that the Am79C971 controller waits for data to be valid is programmable. ROMTMG (BCR18, bits 15-12) defines the time from when the Am79C971 controller drives EBUA_EBA[7:0] with the lower 8 bits of the Expansion ROM address to when the Am79C971 controller latches in the data on the EBD[7:0] inputs. The register value specifies the time in number of clock cycles. When ROMTMG is set to nine (the default value), EBD[7:0] is sampled with the next rising edge of CLK ten clock cycles after EBUA_EBA[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. All four bytes of Expansion ROM data are stored in holding registers. One clock cycle after the last data byte is available, the Am79C971 controller asserts TRDY.

The access time for the Expansion ROM device (tACC) can be calculated by subtracting the clock-to-output delay for the EBUA_EBA[7:0] outputs (tv_A_D) and the input-to-clock setup time for the EBD[7:0] inputs (ts_D) from the time defined by ROMTMG:

tACC $\leq$ ROMTMG* clock period - tv_A_D - ts_D

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 in Figure 45 assumes the default programming of ROMTMG (1001b = 9 CLK). After reading the first byte, the Am79C971 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 50. When the host tries to perform a burst read of the Expansion ROM, the Am79C971 controller will disconnect the access at the second data phase.

The host must program the Expansion ROM Base Address register in the PCI configuration space before the first access to the Expansion ROM. The Am79C971 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 1.

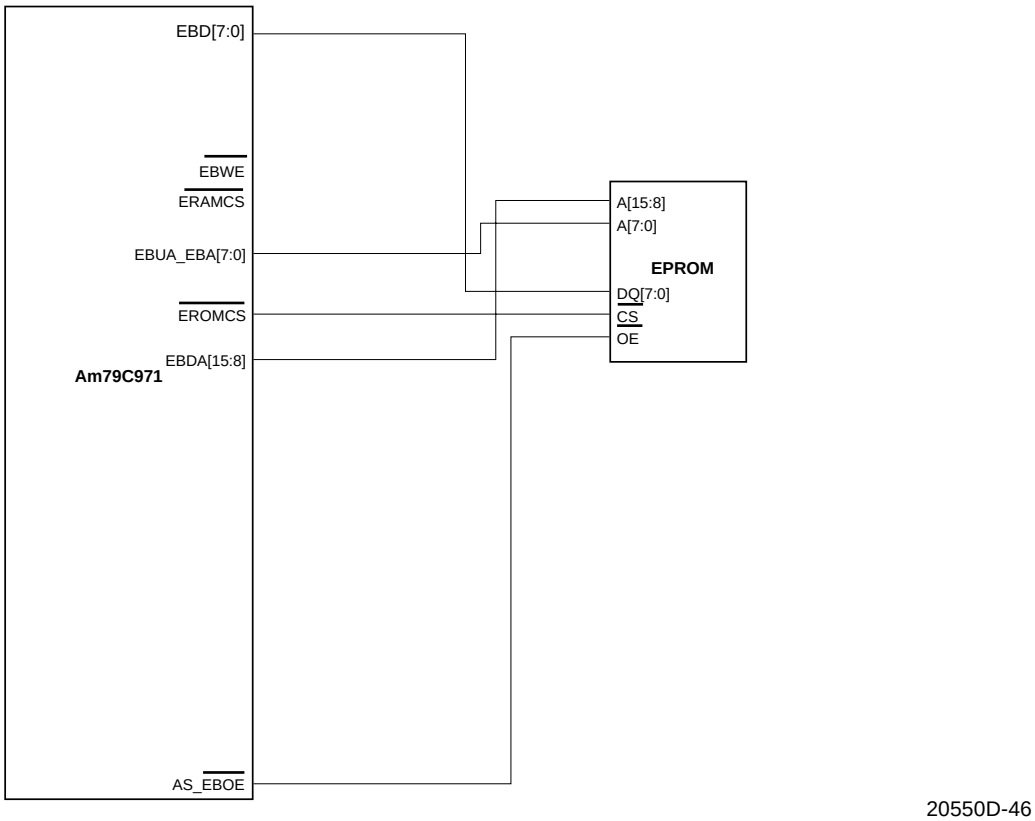


Figure 43. EPROM Only Configuration for the Expansion Bus (64K EPROM)

After the Expansion ROM is enabled, the Am79C971 controller will claim all memory read accesses with an address between ROMBASE and ROMBASE + 1M - 4 (ROMBASE, PCI Expansion ROM Base Address register, bits 31-20). The address output to the Expansion ROM is the offset from the address on the PCI bus to ROMBASE. The Am79C971 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 Am79C971 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 EBD pins together and tying them high or low.

Direct Flash Access

Am79C971 controller supports Flash as an Expansion ROM device, as well as providing a read/write data path to the Flash. The Am79C971 controller will support up to 1 Mbyte of Flash on the Expansion Bus. The Flash is accessed by a read or write to the Expansion Bus Data port (BCR30). The user must load the upper address EPADDRU (BCR 29, bits 3-0) and then set the FLASH (BCR29, bit 15) bit to a 1. The Flash read/write utilizes the PCI clock instead of the EBCLK during all accesses. EPADDRU is not needed if the Flash size is 64K or less, but still must be programmed. The user will then load the lower 16 bits of address, EPADDRL (BCR 28, bits 15-0).

Flash/EPROM Read

A read to the Expansion Bus Data Port (BCR30) will start a read cycle on the Expansion Bus Interface. The Am79C971 controller will drive EBUA_EBA[7:0] with the most significant address byte at the same time the Am79C971 controller will drive AS_$\overline{\text{EBOE}}$ high to strobe the address in the external ‘374 (D flip-flop). On the next clock, the Am79C971 controller will drive EBDA[15:8] and EBUA_EBA[7:0] with the middle and least significant address bytes.

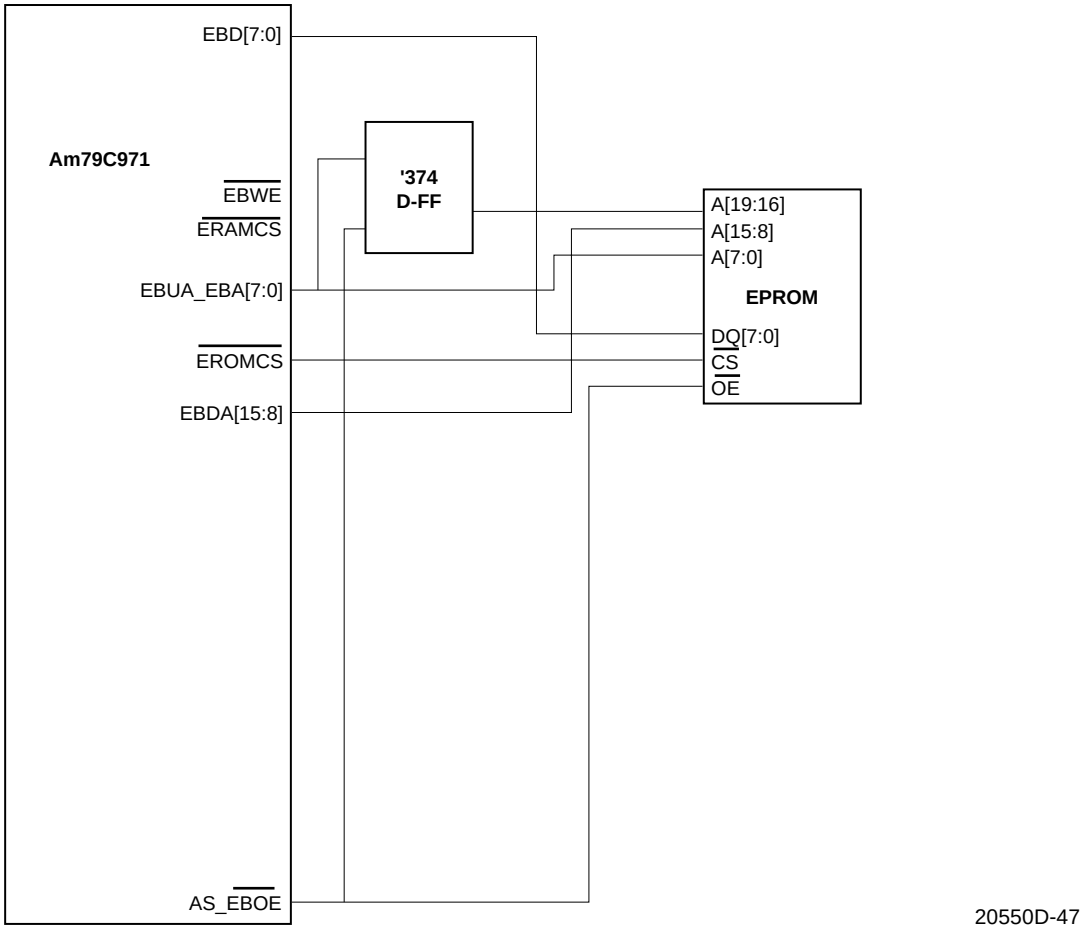


Figure 44. EPROM Only Configuration for the Expansion Bus (>64K EPROM)

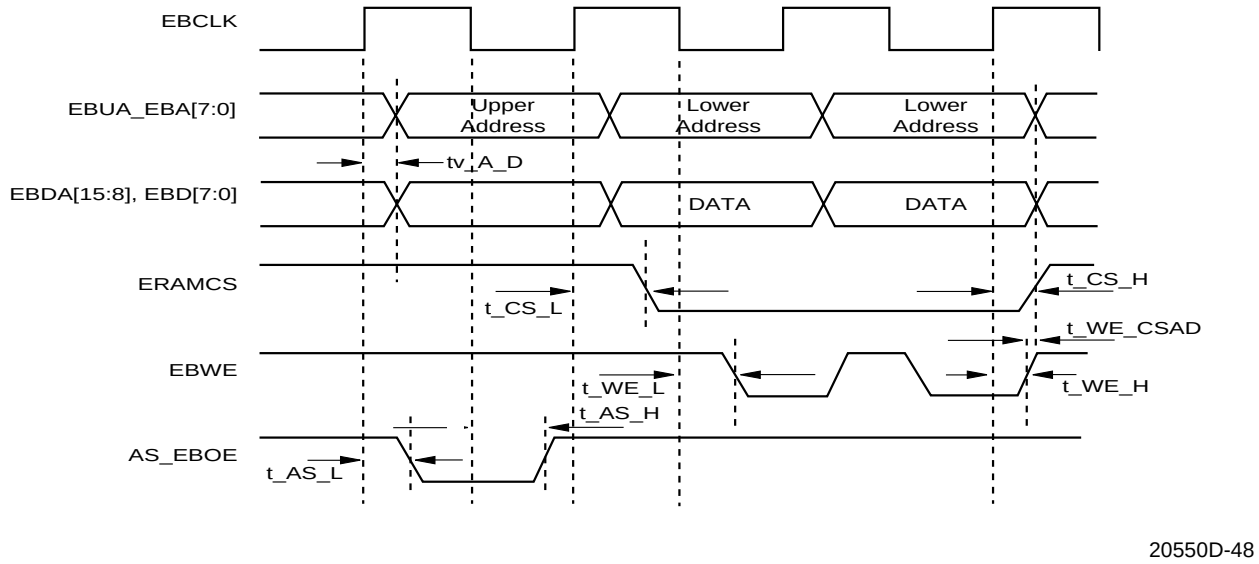


Figure 45. Expansion ROM Bus Read Sequence

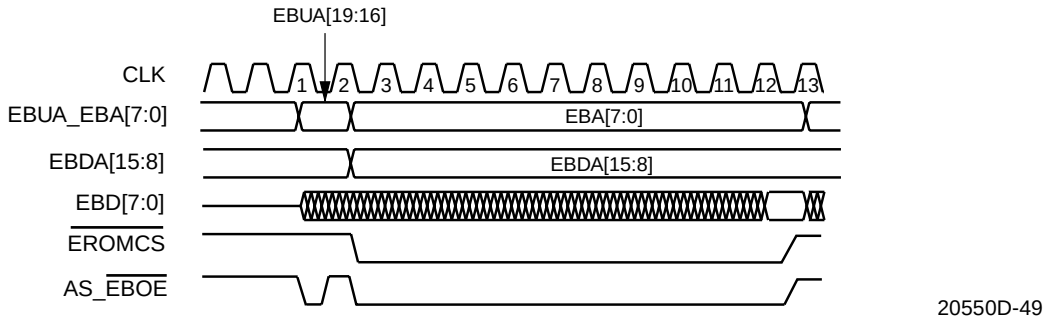


Figure 46. Flash Read from Expansion Bus Data Port

The EROMCS is driven low for the value ROMTMG + 1. Figure 46 assumes that ROMTMG is set to nine. EBD[7:0] is sampled with the next rising edge of CLK ten clock cycles after EBUA_EBA[7:0] was driven with a new address value. This PCI slave access to the Flash/EPROM will result in a retry for the very first access. Subsequent accesses may give a retry or not, depending on whether or not the data is present and valid. The access time is dependent on the ROMTMG bits (BCR18, bits 15-12) and the Flash/EPROM. This access mechanism differs from the Expansion ROM access mechanism since only one byte is read in this manner, instead of the 4 bytes in an Expansion ROM access. The PCI bus will not be held during accesses through the Expansion Bus Data Port. If the LAAINC (BCR29, bit 15) is set, the EBADDRL address will be incremented and a continuous series of reads from the Expansion Data Port (EBDATA, BCR30) is possible. The address incrementor will roll over without warning and without incrementing the upper address EBADDRU.

The Flash write is almost the same procedure as the read access, except that the Am79C971 controller will not drive AS_EBOE low. The EROMCS and EBWE are driven low for the value ROMTMG again. The write to the FLASH port is a posted write and will not result in a retry to the PCI unless the host tries to write a new value before the previous write is complete, then the host will experience a retry. The FLASH can only be accessed while in STOP or when the SRAM_SIZE = 0 (BCR25, bits 7-0). See Figure 47.

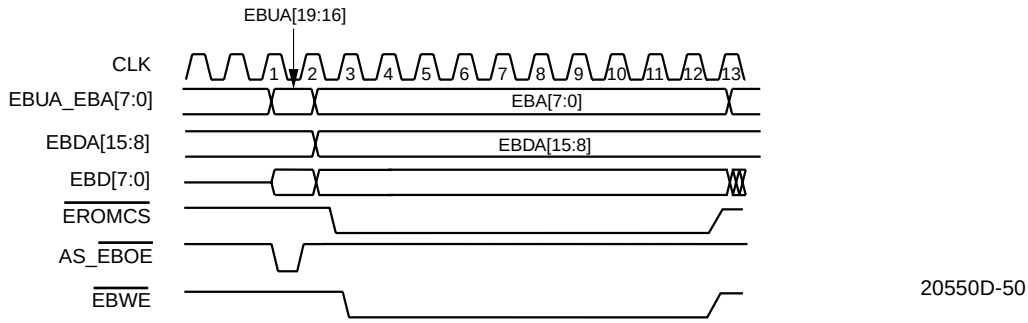


Figure 47. Flash Write from Expansion Bus Data Port

AMD Flash Programming

AMD's Flash products are programmed on a byte-by-byte basis. Programming is a four bus cycle operation. There are two "unlock" write cycles. These are followed by the program set-up command and data write cycles. Addresses are latched on the falling edge of EBWE and the data is latched on the rising edge of EBWE. The rising edge of EBWE begins programming.

Upon executing the AMD Flash Embedded Program Algorithm command sequence, the Am79C971 controller is not required to provide further controls or timing. The AMD Flash product will compliment EBD[7] during a read of the programmed location until the programming is complete. The host software should poll the programmed address until EBD[7] matches the programmed value.

AMD Flash byte programming is allowed in any sequence and across sector boundaries. *Note that a data 0 cannot be programmed back to a 1. Only erase operations can convert zeros to ones.* AMD Flash chip erase is a six-bus cycle operation. There are two *unlock* write cycles, followed by writing the set-up command. Two more *unlock* cycles are then followed by the chip erase command. Chip erase does *not* require the user to program the device prior to erasure. Upon executing the AMD Flash Embedded Erase Algorithm command sequence, the Flash device will program and verify the entire memory for an all zero data pattern prior to electrical erase. The Am79C971 controller is not required to provide any controls or timings during these operations. The automatic erase begins on the rising edge of the last EBWE pulse in the command sequence and terminates when the data on EBD[7] is 1, at which time the Flash device returns to the read mode. Polling by the Am79C971 controller is not required during the erase sequence. The following FLASH programming-table excerpt (Table 12) shows the command sequence for byte programming and sector/chip erasure on an AMD Flash device. In the following table, PA and PD stand for programmed address and programmed data, and SA stands for sector address.

The Am79C971 controller will support only a single sector erase per command and not concurrent sector erasures. The Am79C971 controller will support most FLASH devices as long as there is no timing requirement between the completion of commands. The FLASH access time cannot be guaranteed with the Am79C971 controller access mechanism. The Am79C971 controller will also support only Flash devices that do not require data hold times after write operations.

Table 12. Am29Fxxx Flash Command

Command Sequence	Bus Write Cycles Req'd	First Bus Write Cycle		Second Bus Write Cycle		Third Bus Write Cycle		Fourth Bus Write Cycle		Fifth Bus Write Cycle		Sixth Bus Write Cycle	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Byte Program	4	5555h	AAh	2AAAh	55H	5555h	A0h	PA	PD				
Chip Erase	6	5555h	AAh	2AAAh	55H	5555h	80h	5555h	AAh	2AAAh	55h	5555h	10h
Sector Erase	6	5555h	AAh	2AAAh	55H	5555h	80h	5555h	AAh	2AAAh	55h	SA	3h

SRAM Configuration

The Am79C971 controller supports SRAM as a FIFO extension as well as providing a read/write data path to the SRAM. See Figure 48. The Am79C971 controller will support up to 128K of SRAM on the Expansion Bus. See Figure 49.

External SRAM Configuration

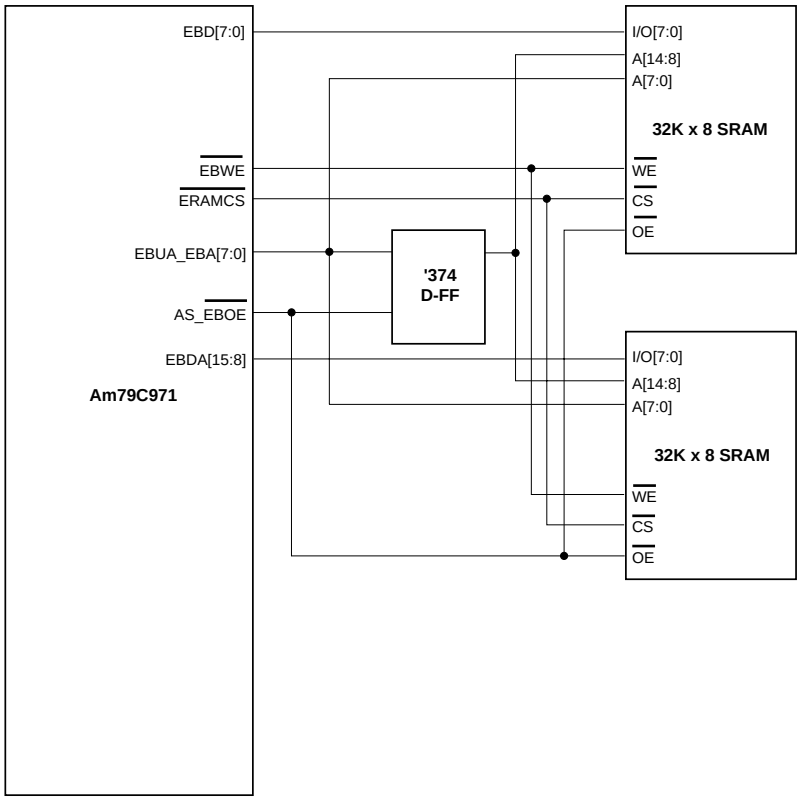
The SRAM_SIZE (BCR25, bits 7-0) programs the size of the external SRAM. SRAM_SIZE can also be programmed to a smaller value than what is present on the Expansion Bus.

The external SRAM should be programmed on a 512-byte boundary. However, there should be no accesses to the RAM space while the Am79C971 controller is running. The Am79C971 controller assumes that it completely owns the SRAM while it is in operation. To specify how much of the SRAM is allocated to transmit and how much is allocated to receive, the user should program SRAM_BND (BCR26, bits 7-0) with the page boundary where the receive buffer begins. The SRAM_BND also should be programmed on a 512-byte boundary. The transmit buffer space starts at 0000h. It is up to the user or the software driver to split up the memory for transmit or receive; there is no defaulted value. The minimum SRAM size required is four 512-byte pages for each transmit and receive queue, which limits the SRAM size to be at least 4 Kbytes.

The SRAM_BND upon H_RESET will be reset to 0000h. The Am79C971 controller will not have any transmit buffer space unless SRAM_BND is programmed. The last configuration parameter necessary is the clock source used to control the Expansion Bus interface. This is programmed through the SRAM Interface Control register. The externally driven Expansion Bus Clock (EBCLK) can be used by specifying a value of 010h in EBCS (BCR27, bits 5-3). This allows the user to utilize any clock that may be available.

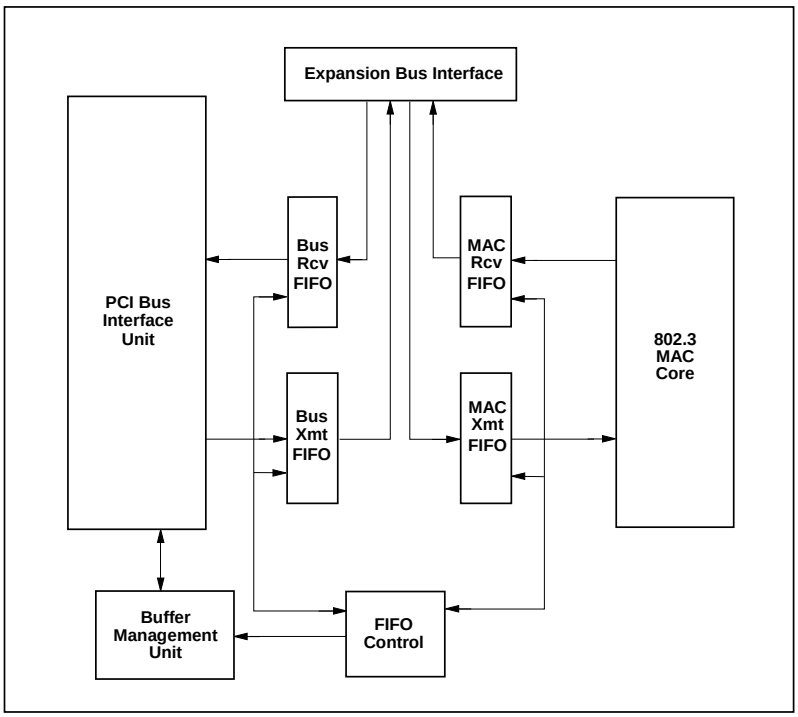
There are two standard clocks that can be chosen as well, the PCI clock or the crystal clock used to power the network MAUs. When the PCI or the crystal clock is used, the EBCLK does not have to be driven, but it must be tied to VDD through a resistor. The user must specify an SRAM clock (BCR27, bits 5-3) that will not stop unless the Am79C971 controller is stopped. Otherwise, the Am79C971 controller will report buffer overflows, underflows, corrupt data, and will hang eventually.

The user can decide to use a fast clock and then divide down the frequency to get a better duty-cycle if required. The choices are a divide by 2 or 4 and is programmed by the CLK_FAC bits (BCR27, bits 2-0). Note that the Am79C971 controller does not support an SRAM frequency above 33 MHz regardless of the clock and clock factor used.



20550D-51

Figure 48. SRAM Only Configuration for the Expansion Bus



20550D-52

Figure 49. Block Diagram With External SRAM

No SRAM Configuration

If the SRAM_SIZE (BCR25, bits 7-0) value is 0 in the SRAM size register, the Am79C971 controller will assume that there is no SRAM present and will reconfigure the four internal FIFOs into two FIFOs, one for transmit and one for receive. The FIFOs will operate the same as in the PCnet-PCI II controller. When the SRAM SIZE (BCR25, bits 7-0) value is 0, the SRAM BND (BCR26, bits 7-0) are ignored by the Am79C971 controller. See Figure 50.

NOTE: A "No SRAM configuration" is only valid for 10Mb mode. In 100Mb mode, SRAM is mandatory and must always be used.

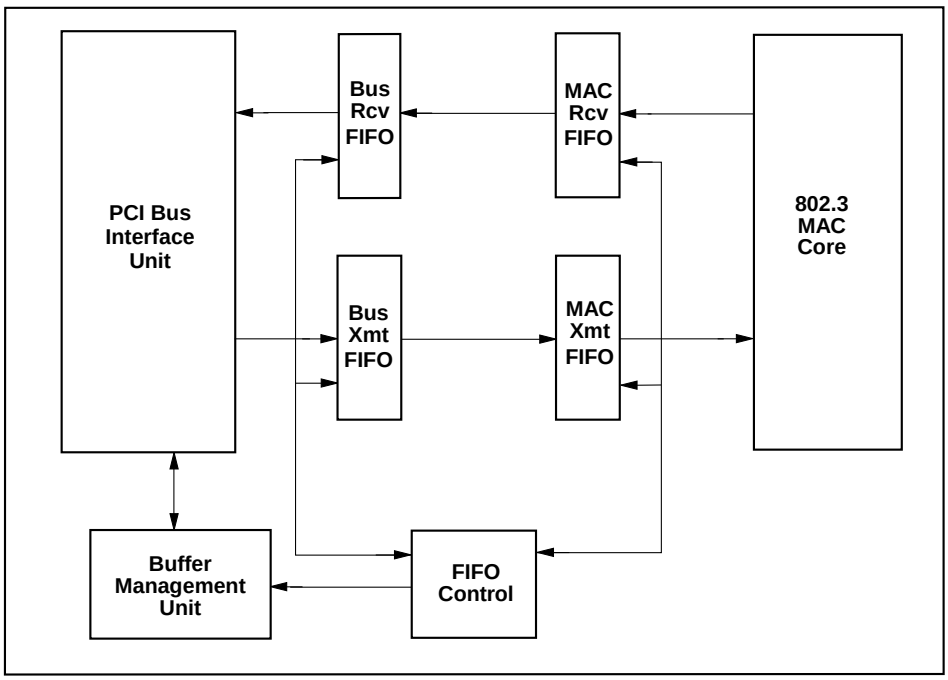
Low Latency Receive Configuration

If the LOLATRX (BCR27, bit 4) bit is set to 1, then the Am79C971 controller will configure itself for a low latency receive configuration. In this mode, external SRAM is required at all times. If the SRAM_SIZE (BCR25, bits 7-0) value is 0, the Am79C971 controller will not configure for low latency receive mode. The Am79C971 controller will provide a fast path on the receive side bypassing the external RAM. All transmit traffic will go to the SRAM, so SRAM_BND (BCR26, bits 7-0) has no meaning in low latency receive mode. When the Am79C971 controller has received 16 bytes from the network, it will start a DMA request to the PCI Bus Interface Unit. The Am79C971 controller will not wait for the first 64 bytes to pass to check for collisions in Low Latency Receive mode. The Am79C971 controller must be in STOP before switching to this mode. See Figure 51.

CAUTION: To provide data integrity when switching into and out of the low latency mode, DO NOT SET the FASTSPNDE bit when setting the SPND bit. Receive frames WILL be overwritten and the Am79C971 controller may give erratic behavior when it is enabled again.

Direct SRAM Access

The SRAM can be accessed through the Expansion Bus Data port (BCR30). To access this data port, the user must load the upper address EPADDRU (BCR29, bits 3-0) and set FLASH (BCR29, bit 15) to 0. Then the user will load the lower 16 bits of address EPADDRL (BCR28, bits 15-0). To initiate a read, the user reads the Expansion Bus Data Port (BCR30). This slave access from the PCI will result in a retry for the very first access. Subsequent accesses may give a retry or not, depending on whether or not the data is present and valid. The direct SRAM access uses the same FLASH/EPROM access except for accessing the SRAM in word format instead of byte format. This access is meant to be a diagnostic access only. The SRAM can only be accessed while the Am79C971 controller is in STOP or SPND (FASTSPNDE is set to 0) mode.



20550D-53

Figure 50. Block Diagram No SRAM Configuration

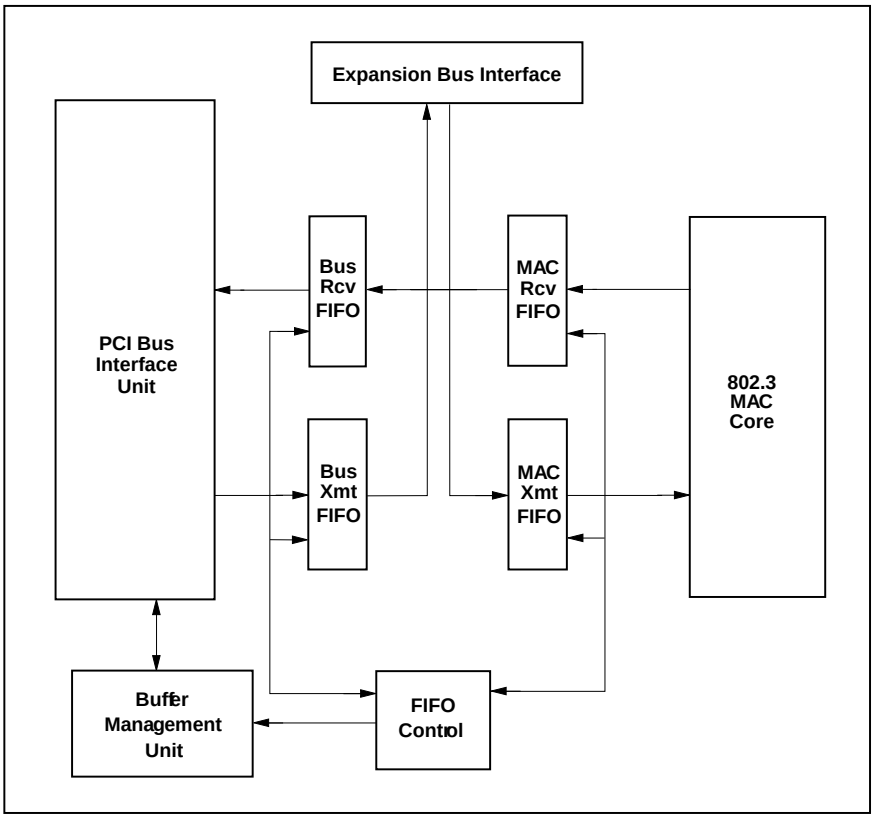


Figure 51. Block Diagram Low Latency Receive Configuration

SRAM Accesses

The SRAM access during normal operations is a single cycle address load to fill the upper bits into the '374 followed by 17 subsequent accesses. This results in the best utilization for the 4-FIFO arbiter in the Am79C971 controller. If the FIFO does not have enough data to complete the full 18 cycles, the arbiter will switch after all of the data has been written or read. This under utilization occurs only at the end of a packet.

The most significant address byte EBUA_EBA[7:0] is registered into the external '374 by assertion of AS_$\overline{\text{EBOE}}$ The least significant address byte is then toggled on the EBUA_EBA[7:0] throughout the remainder of the read/write access. The data word is made up of the most significant data byte EBDA[15:8], and the least significant data byte EBD[7:0]. $\overline{\text{ERAMCS}}$ is connected to the $\overline{\text{CE}}$/$\overline{\text{CS}}$ chip select of the external SRAM. AS_$\overline{\text{EBOE}}$ provides the output enable signal to the SRAM during read operations. During write operations, the AS_$\overline{\text{EBOE}}$ is driven high during the remainder of the accesses. $\overline{\text{EBWE}}$ is toggled during SRAM writes. See Figure 52 and Figure 53.

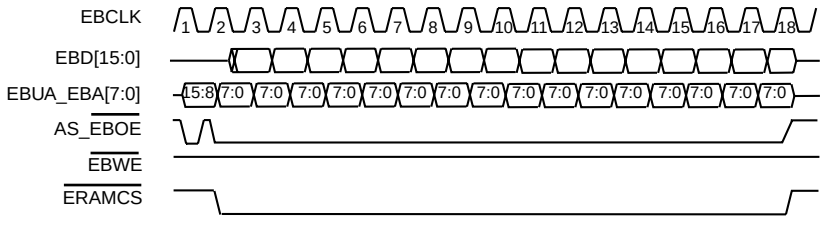
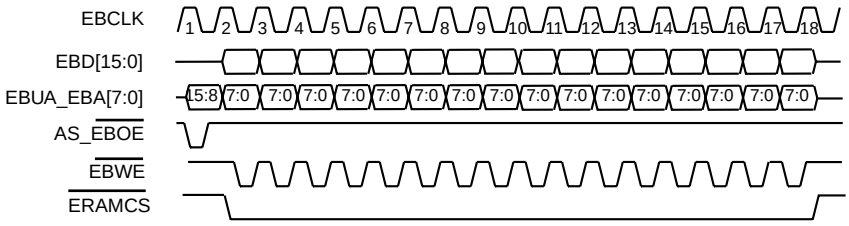


Figure 52. Typical SRAM Read Operation



Note:
EBD[15:0] = EBDA[15:8]+EBD[7:0]

20550D-56

Figure 53. Typical SRAM Write Operation

SRAM Interface Bandwidth Requirements

When the EBCLK pin is used to drive the Expansion Bus cycles and external SRAMs are present, the CLK_FAC (BCR27, bits 2-0) selects the clock factor for the Expansion Bus Clock (EBCLK). The Expansion Bus Clock can be divided down by factors of 2 or 4. For maximum throughput capability to support maximum wire rates in a full-duplex 100-Mbps network, a 33-MHz clock should be supplied to the EBCLK input pin and 15-ns SRAM devices must be used. For systems with lower throughput requirements, a lower clock frequency, along with slower speed SRAM devices, may be used. In a half-duplex 10-Mbps design, an EBCLK frequency as low as 2.5 MHz may be used, while still providing sufficient bandwidth on the SRAM interface to keep up with maximum wire data rates.

Frequency Demands for Network Operation

The minimum supported clock frequency on the Expansion Bus for normal network operations is 10 MHz. The minimum supported clock frequency on the PCI Bus for normal network operations is 15 MHz. The PCI clock pin can be stopped or run at any frequency, but may give underflows and overflows due to reduced bandwidth. These minimum requirements apply only to 10-Mbps half-duplex operation. Details of the clock frequency and SRAM depth requirements for typical network can be found in the *PCnet Fast Buffer Memory Performance White Paper, PID #20898A*.

EEPROM Interface

The Am79C971 controller contains a built-in capability for reading and writing to an external serial 93C46 EEPROM. This built-in capability consists of an interface for direct connection to a 93C46 compatible EEPROM, an automatic EEPROM read feature, and a user-programmable register that allows direct access to the interface pins.

Automatic EEPROM Read Operation

Shortly after the deassertion of the $\overline{\text{RST}}$ pin, the Am79C971 controller will read the contents of the EEPROM that is attached to the interface. Because of this automatic-read capability of the Am79C971 controller, an EEPROM can be used to program many of the features of the Am79C971 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 interface, the Am79C971 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 Am79C971 controller. Access to the Am79C971 configuration space, the Expansion ROM or any I/O resource is not possible during the EEPROM read operation. The Am79C971 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 1. If the checksum verification of the EEPROM data fails, PVALID will be cleared to 0, and the Am79C971 controller will force all EEPROM-programmable BCR registers back to their H_RESET default values. However, the content of the Address PROM locations (offsets 0h - Fh from the I/O or memory mapped I/O base address) will not be cleared. The 8-bit checksum for the entire 64 bytes of the EEPROM should be FFh.

If no EEPROM is present at the time of the automatic read operation, the Am79C971 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 an Am79C971 controller reread of the EEPROM so that the new EEPROM contents will be loaded into the EEPROM-programmable registers on board the Am79C971 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 Am79C971 controller to ignore further accesses to the Am79C971 configuration space, the Expansion ROM, or any I/O resource until the completion of the EEPROM read operation. The Am79C971 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 Am79C971 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 Am79C971 controller will sample the value of the EESK/$\overline{\text{LED1}}$/SFBD pin. If the sampled value is a 1, then the Am79C971 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 0, the Am79C971 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 Am79C971 controller, since the checksum verification will fail.

Direct Access to the Interface

The user may directly access the port through the EEPROM register, BCR19. This register contains bits that can be used to control the 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
- BCR4 LED0 Status
- BCR5 LED1 Status
- BCR6 LED2 Status
- BCR7 LED3 Status
- BCR9 Full-Duplex Control
- BCR18 Burst and Bus Control
- BCR22 PCI Latency
- BCR23 PCI Subsystem Vendor ID
- BCR24 PCI Subsystem ID
- BCR25 SRAM Size
- BCR26 SRAM Boundary
- BCR27 SRAM Interface Control
- BCR32 MII Control and Status
- BCR33 MII Address
- BCR35 PCI Vendor ID

If PREAD (BCR19, bit 14) and PVALID (BCR19, bit 15) are cleared to 0, 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 Am79C971 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 32 words (i.e., 64 bytes) of the EEPROM. The format of the EEPROM contents is shown in Table 14, beginning with the byte that resides at the lowest EEPROM address.

Note that the first bit out of any word location in the EEPROM is treated as the MSB of the register being programmed. For example, the first bit out of EEPROM word location 09h will be written into BCR4, bit 15; the second bit out of EEPROM word location 09h will be written into BCR4, bit 14, etc.

Table 13. EEPROM Content

Word Address	Byte Addr.	Most Significant Byte	Byte Addr.	Least Significant Byte
00h*	01h	2nd 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	4th byte of the node address	02h	3rd byte of the node address
02h	05h	6th byte of the node address	04h	5th 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	MSB of two-byte checksum, which is the sum of bytes 00h-0Bh and bytes 0Eh and 0Fh	0Ch	LSB 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	BCR2[15:8] (Miscellaneous Configuration)	10h	BCR2[7:0] (Miscellaneous Configuration)
09h	13h	BCR4[15:8] (Link Status LED)	12h	BCR4[7:0] (Link Status LED)
0Ah	15h	BCR5[15:8] (LED1 Status)	14h	BCR5[7:0] (LED1 Status)
0Bh	17h	BCR6[15:8] (LED2 Status)	16h	BCR6[7:0] (LED2 Status)
0Ch	19h	BCR7[15:8] (LED3 Status)	18h	BCR7[7:0] (LED3 Status)
0Dh	1Bh	BCR9[15:8] (Full-Duplex Control)	1Ah	BCR9[7:0] (Full-Duplex Control)
0Eh	1Dh	BCR18[15:8] (Burst and Bus Control)	1Ch	BCR18[7:0] (Burst and Bus Control)
0Fh	1Fh	BCR22[15:8] (PCI Latency)	1Eh	BCR22[7:0] (PCI Latency)
10h	21h	BCR23[15:8] (PCI Subsystem Vendor ID)	20h	BCR23[7:0] (PCI Subsystem Vendor ID)
11h	23h	BCR24[15:8] (PCI Subsystem ID)	22h	BCR24[7:0] (PCI Subsystem ID)
12h	25h	BCR25[15:8] (SRAM Size)	24h	BCR25[7:0] (SRAM Size)
13h	27h	BCR26[15:8] (SRAM Boundary)	26h	BCR26[7:0] (SRAM Boundary)
14h	29h	BCR27[15:8] (SRAM Interface Control)	28h	BCR27[7:0] (SRAM Interface Control)
15h	2Bh	BCR32[15:8] (MII Control and Status)	2Ah	BCR32[7:0] (MII Control and Status)
16h	2Dh	BCR33[15:8] (MII Address)	2Ch	BCR33[7:0] (MII Address)
17h	2Fh	BCR35[15:8] (PCI Vendor ID)	2Eh	BCR35[7:0] (PCI Vendor ID)
18h	31h	Reserved location must be 00h	30h	Reserved location must be 00h
19h	33h	Reserved location must be 00h	32h	Reserved location must be 00h
1Ah	35h	Reserved location must be 00h	34h	Reserved location must be 00h
1Bh	37h	Reserved location must be 00h	36h	Reserved location must be 00h
1Ch	39h	Reserved location must be 00h	38h	Reserved location must be 00h
1Dh	3Bh	Reserved location must be 00h	3Ah	Reserved location must be 00h
1Eh	3Dh	Reserved location must be 00h	3Ch	Reserved location must be 00h
1Fh	3Fh	Checksum adjust byte for the 64 bytes of the EEPROM contents, checksum of the 64 bytes of the EEPROM should total to FFh	3Eh	Reserved location must be 00h

Note:
**Lowest EEPROM address.*

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 3Fh) 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 64 bytes of EEPROM data equals the value FFh. The checksum adjust byte is needed by the Am79C971 controller in order to verify that the EEPROM content has not been corrupted.

LED Support

The Am79C971 controller can support up to four LEDs. LED outputs $\overline{\text{LED0}}$, $\overline{\text{LED1}}$, and $\overline{\text{LED2}}$ allow for direct connection of an LED and its supporting pullup device.

In applications that want to use the pin to drive an LED and also have an EEPROM, it might be necessary to buffer the $\overline{\text{LED3}}$ circuit from the EEPROM connection. When an LED circuit is directly connected to the EEDO/$\overline{\text{LED3}}$/SRD pin, then it is not possible for most EEPROM devices to sink enough I_{OL} to maintain a valid low level on the EEDO input to the Am79C971 controller.

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, Receive Match, Receive Polarity, Receive Status, Magic Packet, Disable Transceiver, MII Enable 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 as shown in Table 14:

Table 14. LED Default Configuration

LED Output	Indication	Driver Mode	Pulse Stretch
$\overline{\text{LED0}}$	Link Status	Open Drain - Active Low	Enabled
$\overline{\text{LED1}}$	Receive Status	Open Drain - Active Low	Enabled
$\overline{\text{LED2}}$	Receive Polarity	Open Drain - Active Low	Enabled
$\overline{\text{LED3}}$	Transmit Status	Open Drain - Active Low	Enabled

For each LED register, each of the status signals is AND'd with its enable signal, and these signals are all OR'd together to form a combined status signal. Each LED pin 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 to 3 clocks of stretched LED output, or 52 ms to 78 ms. See Figure 54.

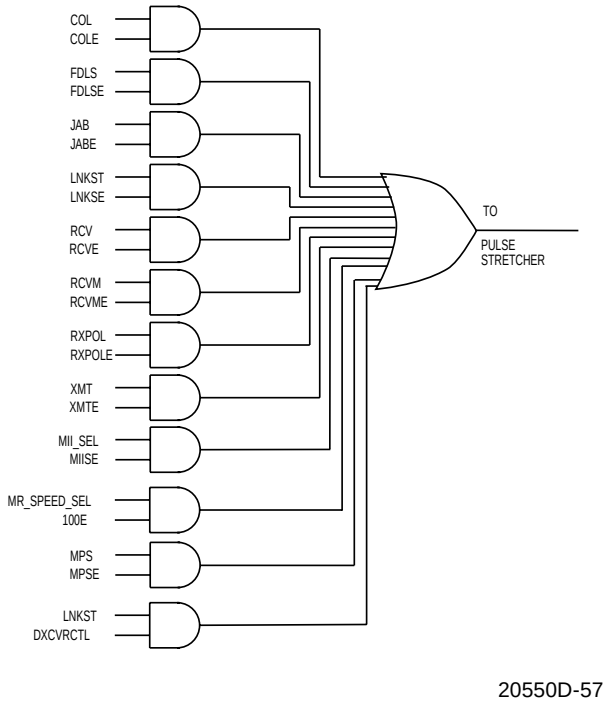


Figure 54. LED Control Logic

Power Savings Modes

SLEEP Mode

The Am79C971 controller supports two hardware power savings modes. Both are entered by driving the $\overline{\text{SLEEP}}$ pin LOW and by leaving the MPMODE (CSR 5, bit 1) bit at its default value of 0.

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 $\overline{\text{SLEEP}}$ pin itself. Coma mode is enabled when AWAKE (BCR2, bit 2) is at its default value of 0 and $\overline{\text{SLEEP}}$ is asserted.

The second power saving mode is called, *snooze mode*. In snooze mode, enabled by setting AWAKE to 1 and driving the $\overline{\text{SLEEP}}$ pin LOW, the T-MAU receive

circuitry will remain active even while the SLEEP pin is driven LOW. All other sections of the device are shut down except the LED0 pin, the only LED pin that continues to function, just as in normal operation. 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. Once the T-MAU has a good link, LED0 will be active. This LED0 pin can be used to drive an LED and/or external hardware that directly controls the SLEEP pin of the Am79C971 controller. In the case of driving external hardware, it can be used to tell an external SLEEP control logic to drive the SLEEP pin HIGH to bring the Am79C971 controller out of the snooze mode. 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.

SLEEP must not be asserted while the Am79C971 controller is requesting the bus or while a bus cycle is active. It is recommended to set the Am79C971 controller into suspend mode (SPND (CSR5, bit 0) set to 1) or to stop the device (STOP (CSR0, bit 2) set to 1) before asserting the SLEEP pin.

Before the sleep mode is invoked, the Am79C971 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 Am79C971 controller can be programmed to generate an interrupt and inform the driver about the wake-up. The Am79C971 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 1. 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 SLEEP pin should not be asserted during power supply ramp-up. If it is desired that SLEEP be asserted at power up time, then the system must delay the assertion of SLEEP until three clock cycles after completion of a hardware reset operation.

Magic Packet Mode

Magic Packet mode is enabled by performing three steps. First, the Am79C971 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 1 if it has not been set already. Finally, either SLEEP must be asserted (hardware control) or MPEN (CSR5, bit 2) must be set to 1 (software control). Note that FASTSPNDE (CSR7, bit 15) has no meaning in Magic Packet mode.

In Magic Packet mode, the Am79C971 controller remains fully powered up (all VDD and VDDB 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 Am79C971 controller are still possible. The 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 Am79C971 MAC and contains a data sequence in its data field made up of 16 repetitions of the physical addresses (PADR[47:0]). The Am79C971 controller will search incoming frames until it finds a Magic Packet frame. It 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 Am79C971 controller reaches the frame's FCS field. Any deviation of the incoming frame's 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 Am79C971 controller supports two different modes of address detection for a Magic Packet frame. If MPPLBA (CSR5, bit 5) is at its default value of 0, the Am79C971 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 1, 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 frame's data sequence must be made up of 16 repetitions of the physical addresses (PADR[47:0]), regardless of what kind of destination address it has.

When the Am79C971 controller detects a Magic Packet frame, it sets MPINT (CSR5, bit 4) to 1. If INEA (CSR0, bit 6) and MPINTE (CSR5, bit 3) are set to 1, 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 indicated that a Magic Packet frame has been received. MPSE (BCR4-7, bit 9) must be set to 1 and the RCVE (BCR4-7, bit 2) must be set to 0 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 1.

Once a Magic Packet frame is detected, the Am79C971 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. Re-initialization should not be performed. If the part is re-initialized, then the descriptor locations will be reset also, and the Am79C971 controller will not start where it left off.

If Magic Packet mode is disabled by the deassertion of SLEEP, then in order to immediately re-enable 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 re-enabled by setting MPEN back to 1.

The PCI bus interface clock (CLK) is not required to be running. Both INTA and the LED pins may be used to indicate the receipt of a Magic Packet frame when the CLK is stopped. If the system wishes to stop the CLK, it should do so after enabling the Magic Packet mode. 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 SLEEP is being deasserted. Otherwise, the receive FIFO may overflow if new frames arrive. The network clock (XTAL) must continue running at all times while in Magic Packet mode.

CAUTION: *To prevent unwanted interrupts from other active parts of the Am79C971 controller, care must be taken to mask all likely interruptible events during Magic Packet mode. An example would be the interrupts from the MII which operate while in Magic Packet mode.*

IEEE 1149.1 (1990) Test Access Port Interface

An IEEE 1149.1-compatible boundary scan Test Access Port is provided for board-level continuity test and diagnostics. All digital input, output, and input/output pins are tested. Analog pins, including the AUI differential driver (DO±) and receivers (DI±, CI±), and the crystal input (XTAL1/XTAL2) pins are tested. The T-MAU drivers TXD±, TXP±, and receiver RXD± are also tested. The following is a brief summary of the IEEE 1149.1-compatible test functions implemented in the Am79C971 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 that the FSM is in the TEST_LOGIC_RESET state at power-up. Therefore, the TRST is not provided. 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 Table 15 for a summary of supported instructions.

Table 15. 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 shown in Table 16.

Table 16. BSR Mode Of Operation

1	Capture
2	Shift
3	Update
4	System Function

Other Data Registers

Other data registers are the following:

1. Bypass Register (1 bit)
2. Device ID register (32 bits) (Table 17).

Table 17. Device ID Register

Bits 31-28	Version
Bits 27-12	Part Number (0010 0110 0010 0011)
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

The contents of the Device ID register is the same as the contents of CSR88.

NAND Tree Testing

The Am79C971 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, MII, and LED signals.

NAND tree testing is enabled by asserting $\overline{RST}$. The result of the NAND tree test can be observed on the $\overline{INTA}$ pin. See Figure 55.

Pin 143 ($\overline{RST}$) is the first input to the NAND tree. Pin 145 (CLK) is the second input to the NAND tree, followed by pin 147 ($\overline{GNT}$). All other PCI bus, Expansion Bus, MII, LED signals follow, counterclockwise, with pin 136 (EECS) being the last. Pins labeled NC, analog interfaces, and all power supply pins are not part of the NAND tree. Table 18 shows the complete list of pins connected to the NAND tree.

$\overline{RST}$ must be asserted low to start a NAND tree test sequence. Initially, all NAND tree inputs except $\overline{RST}$ should be driven high. This will result in a high output at the $\overline{INTA}$ 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, $\overline{INTA}$ will toggle every time an additional input is driven low. $\overline{INTA}$ will change to low, when CLK is driven low and all other NAND tree inputs stay high. $\overline{INTA}$ will toggle back to high, when GNT is additionally driven low. The square wave will continue until all NAND tree inputs are driven low. $\overline{INTA}$ will be high, when all NAND tree inputs are driven low. See Figure 56.

Note: *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 Am79C971 controller is configured for NAND tree testing.*

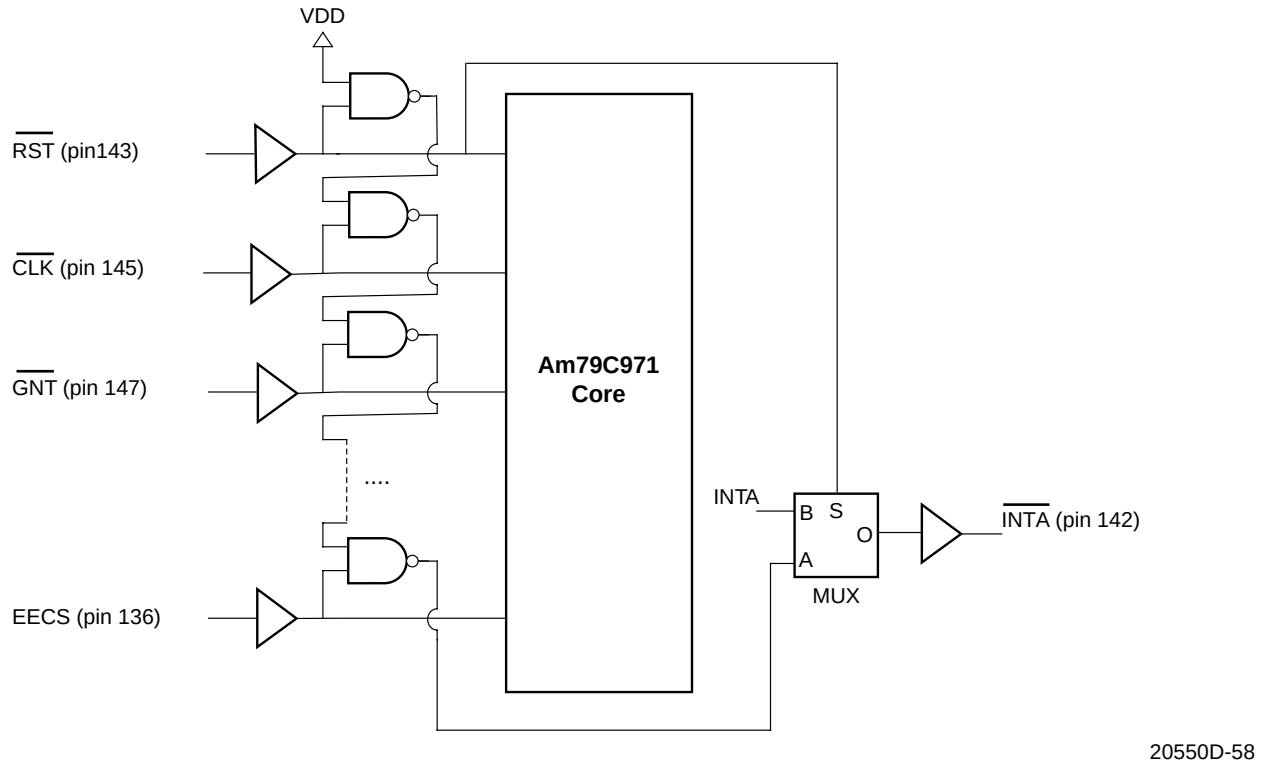


Figure 55. NAND Tree Circuitry

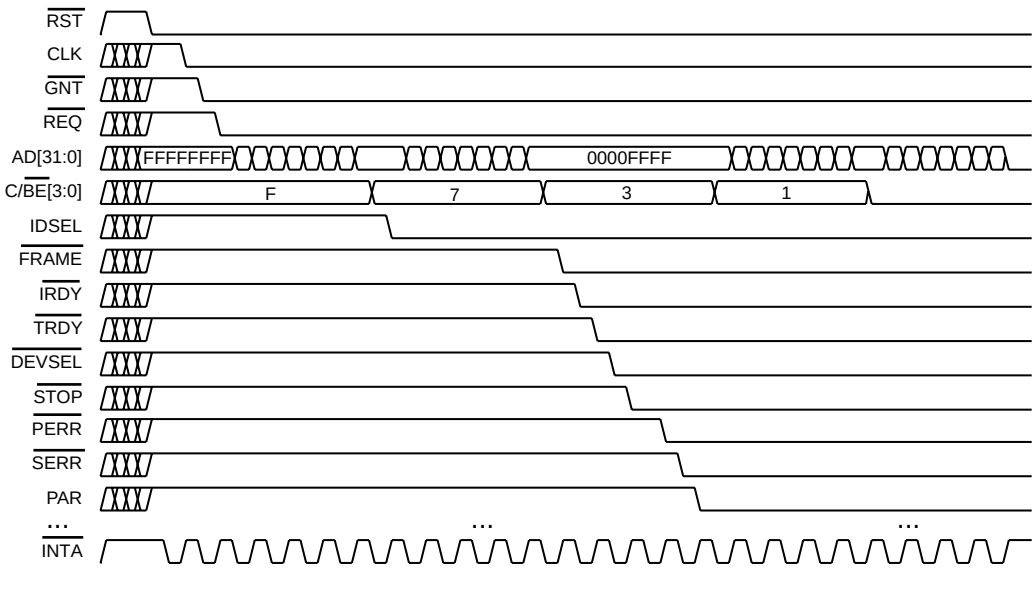


Figure 56. NAND Tree Waveform

Table 18. 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	143	RST	25	16	IRDY	49	47	AD0
2	145	CLK	26	17	TRDY	50	79	CRS
3	147	GNT	27	18	DEVSEL	51	80	COL
4	148	REQ	28	19	STOP	52	81	TXD3
5	149	AD31	29	21	PERR	53	90	TXD2
6	151	AD30	30	22	SERR	54	91	TXD1
7	152	AD29	31	24	PAR	55	92	TXD0
8	153	AD28	32	25	C/$\overline{BE}$1	56	94	TX_EN
9	154	AD27	33	26	AD15	57	95	TX_CLK
10	155	AD26	34	28	AD14	58	96	TX_ER
11	157	AD25	35	29	AD13	59	98	RX_ER
12	159	C/$\overline{BE}$3	36	29	AD12	60	99	RX_CLK
13	160	AD24	37	31	AD11	61	100	RX_DV
14	1	IDSEL	38	32	AD10	62	102	RXD0
15	3	AD23	39	34	AD9	63	103	RXD1
16	4	AD22	40	35	AD8	64	104	RXD2
17	6	AD21	41	37	C/$\overline{BE}$0	65	105	RXD3
18	7	AD20	42	38	AD7	66	106	$\overline{SLEEP}$/$\overline{EAR}$
19	9	AD19	43	39	AD6	67	108	MDIO
20	10	AD18	44	41	AD5	68	131	EEDO/$\overline{LED3}$/SRD
21	12	AD17	45	42	AD4	69	132	EEDI/$\overline{LED0}$
22	13	AD16	46	43	AD3	70	133	$\overline{LED2}$/SRDCLK
23	14	C/$\overline{BE}$2	47	44	AD2	71	134	EESK/$\overline{LED1}$/SFBD
24	15	FRAME	48	46	AD1	72	136	EECS

Reset

There are three different types of RESET operations that may be performed on the Am79C971 device, H_RESET, S_RESET, and STOP. The following is a description of each type of RESET operation.

H_RESET

Hardware Reset (H_RESET) is an Am79C971 reset operation that has been created by the proper assertion of the $\overline{RST}$ pin of the Am79C971 device. When the minimum pulse width timing as specified in the $\overline{RST}$ 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 Am79C971 controller will attempt to read the EEPROM device through the EEPROM interface. H_RESET resets the T-MAU into the Link Fail state.

H_RESET will clear DWIO (BCR18, bit 7) and the Am79C971 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

Software Reset (S_RESET) is an Am79C971 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 Am79C971 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 location. 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 Am79C971 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 Am79C971 controller before starting network activity. S_RESET will cause $\overline{REQ}$ 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 1 to the STOP bit of CSR0, when the stop bit currently has a value of 0, will initiate a STOP reset. If the STOP bit is already a 1, then writing a 1 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 Am79C971 controller will not attempt to read the EEPROM device. Setting the STOP bit does not affect the T-MAU.

Note: *STOP will not cause a deassertion of the $\overline{REQ}$ signal, if it happens to be active at the time of the write to CSR0. The Am79C971 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 Am79C971 controller implements a 256-byte configuration space as defined by the PCI specification revision 2.1. The 64-byte header includes all registers required to identify the Am79C971 controller and its function. Additional registers are used to setup the configuration of the Am79C971 controller in a system. None of the device specific registers located at offsets 40h through FCh are implemented. The layout of the Am79C971 PCI configuration space is shown in Table 19.

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 0.

Table 19. 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
Subsystem ID		Subsystem Vendor ID		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 Am79C971 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 Am79C971 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 Am79C971 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 Am79C971 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 Am79C971 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 used 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 point 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 point 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 0s when an H_RESET or S_RESET occurs. RAP is unaffected by setting the STOP bit.

Address PROM Space

The Am79C971 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 and 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 six bytes of the IEEE station address occupy the first six 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. APROMWE bit (BCR2, bit 8) must be set to 1 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 Am79C971 controller. The internal S_RESET pulse that is generated by this access is different from both the assertion of the hardware $\overline{RST}$ pin (H_RESET) and from the assertion of the software STOP bit. Specifically, S_RESET is the equivalent of the assertion of the $\overline{RST}$ 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 Am79C971 controller does not have a similar requirement. The write access is not required and does not have any effect.

Note: *The Am79C971 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 Am79C971 controller will terminate all slave accesses with the assertion of $\overline{DEVSEL}$ and $\overline{STOP}$ while $\overline{TRDY}$ 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 Am79C971 controller is programmed to operate in Word I/O mode. DWIO (BCR18, bit 7) will be cleared to 0. It will then be loaded by the value in the EEPROM. Table 20 shows how the 32 bytes of address space are used in Word I/O mode.

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 Am79C971 control registers. Table 20 shows legal I/O accesses in Word I/O mode.

Table 20. 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

Double Word I/O Mode

After H_RESET, the Am79C971 controller is programmed to operate in Word I/O mode. DWIO (BCR18, bit 7) will be cleared to 0. It will then be loaded by the value in the EEPROM. Table 20 shows how the 32 bytes of address space are used in Word I/O mode.

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 Am79C971 control registers. Table 23 shows legal I/O accesses in Word I/O mode.

Table 21. 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

Double Word I/O Mode

The Am79C971 controller can be configured to operate in DWord (32-bit) 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 Am79C971 controller. Setting the device into 32-bit I/O mode is usually the first operation after H_RESET or S_RESET. The RAP register will point to CSR0 at that time. Writing a value of 0 to CSR0 is a safe operation. DWIO (BCR18, bit 7) will be set to 1 as an indication that the Am79C971 controller operates in 32-bit I/O mode.

The DWIO mode can be configured from the EEPROM or programmed by the software.

Note: *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 1, only H_RESET or a read of EEPROM can clear it to 0. The DWIO mode setting is unaffected by setting the STOP bit. Table 24 shows how the 32 bytes of address space are used in DWord I/O mode.*

All I/O resources must be accessed in DWord quantities and on DWord addresses. A read access other than listed in Table 25 will yield undefined data, a write operation may cause unexpected reprogramming of the Am79C971 control registers

After H_RESET, the Am79C971 controller is programmed to operate in Word I/O mode. DWIO (BCR18, bit 7) will be cleared to 0. It will then be loaded by the value in the EEPROM. Table 20 shows how the 32 bytes of address space are used in Word I/O mode.

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 Am79C971 control registers. Table 23 shows legal I/O accesses in Word I/O mode

.

Table 22. 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

Double Word I/O Mode

The Am79C971 controller can be configured to operate in DWord (32-bit) 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 Am79C971 controller. Setting the device into 32-bit I/O mode is usually the first operation after H_RESET or S_RESET. The RAP register will point to CSR0 at that time. Writing a value of 0 to CSR0 is a safe operation. DWIO (BCR18, bit 7) will be set to 1 as an indication that the Am79C971 controller operates in 32-bit I/O mode.

The DWIO mode can be configured from the EEPROM or programmed by the software.

Note: *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 1, only H_RESET or a read of EEPROM can clear it to 0. The DWIO mode setting is unaffected by setting the STOP bit. Table 24 shows how the 32 bytes of address space are used in DWord I/O mode.*

All I/O resources must be accessed in DWord quantities and on DWord addresses. A read access other than listed in Table 25 will yield undefined data, a write operation may cause unexpected reprogramming of the Am79C971 control registers

Table 23. 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 locations 3h (MSB) and 2h (LSB), 7h and 6h, Bh and Ah or Fh and 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

Table 24. 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

Table 25. Legal I/O Accesses in Double Word I/O Mode (DWIO =1)

AD[4:0]	$\overline{BE}$[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
11100	0000	RD	DWord read of BDP
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
11100	0000	WR	DWord write of BDP

USER ACCESSIBLE REGISTERS

The Am79C971 controller has three types of user registers: the PCI configuration registers, the Control and Status registers (CSR), and the Bus Control registers (BCR).

The Am79C971 controller implements all PCnet-ISA (Am79C960) registers, all C-LANCE (Am79C90) registers, plus a number of additional registers. The Am79C971 controller CSRs are compatible upon power up with both the PCnet-ISA CSRs and all of the C-LANCE 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.

The Am79C971 registers can be divided into four groups: PCI Configuration, Setup, Running, and Test. Registers not included in any of these categories can be assumed to be intended for diagnostic purposes.

■ PCI Configuration Registers

These registers are intended to be initialized by the system initialization procedure (e.g., BIOS device initialization routine) to program the operation of the Am79C971 controller PCI bus interface.

The following is a list of the registers that would typically need to be programmed once during the initialization of the Am79C971 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

These registers are intended to be initialized by the device driver to program the operation of various Am79C971 controller features.

The following is a list of the registers that would typically need to be programmed once during the setup of the Am79C971 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.

Note: *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
CSR7	Extended Control and Interrupt2
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 Ring Lower
CSR25*	Base Address of Receive Ring Upper
CSR30*	Base Address of Transmit Ring Lower
CSR31*	Base Address of Transmit Ring Upper
CSR47*	Transmit Polling Interval
CSR49*	Receive Polling Interval
CSR76*	Receive Ring Length
CSR78*	Transmit Ring Length
CSR80	DMA Transfer Counter and FIFO Threshold Control
CSR82	Bus Activity Timer
CSR100	Memory Error Timeout
CSR122	Receiver Packet Alignment Control
CSR125^	MAC Enhanced Configuration Control
BCR2^	Miscellaneous Configuration
BCR4^	LED0 Status
BCR5^	LED1 Status
BCR6^	LED2 Status
BCR7^	LED3 Status
BCR9^	Full-Duplex Control
BCR18^	Bus and Burst Control
BCR19	EEPROM Control and Status
BCR20	Software Style
BCR22^	PCI Latency
BCR23^	PCI Subsystem Vendor ID
BCR24^	PCI Subsystem ID
BCR25^	SRAM Size
BCR26^	SRAM Boundary
BCR27^	SRAM Interface Control
BCR32^	MII Control and Status
BCR33^	MII Address
BCR35^	PCI Vendor ID

■ Running Registers

These registers are intended to be used by the device driver software after the Am79C971 controller is running to access status information and to pass control information.

The following is a list of the registers that would typically need to be periodically read and perhaps written during the normal running operation of the Am79C971 controller within a system. Each of these registers contains control bits, or status bits, or both.

RAP	Register Address Port
CSR0	Am79C971 Controller Status
CSR3	Interrupt Masks and Deferral Control
CSR4	Test and Features Control
CSR5	Extended Control and Interrupt
CSR7	Extended Control and Interrupt2
CSR112	Missed Frame Count
CSR114	PCI Status register

The following registers are only necessary if an external PHY device is being used and accessed.

BCR32	MII Control and Status
BCR33	MII Address
BCR34	MII Management Data

■ **Test Registers**

These registers are intended to be used only for testing and diagnostic purposes. Those registers not included in any of the above lists can be assumed to be intended for diagnostic purposes.

PCI Configuration Registers

PCI Vendor ID Register

Offset 00h

The PCI Vendor ID register is a 16-bit register that identifies the manufacturer of the Am79C971 controller. AMD's Vendor ID is 1022h. Note that this vendor ID is not the same as the Manufacturer ID in CSR88 and CSR89. 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 Am79C971 controller within AMD's product line. The Am79C971 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 AMD. The Device ID is the same as the PCnet-PCI II (Am79C970A) device.

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

This register is the same as BCR35 and can be written by the EEPROM.

PCI Command Register

Offset 04h

The PCI Command register is a 16-bit register used to control the gross functionality of the Am79C971 controller. It controls the Am79C971 controller's ability to generate and respond to PCI bus cycles. To logically disconnect the Am79C971 device from all PCI bus cycles except configuration cycles, a value of 0 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 Am79C971 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	RES	Reserved location. Read as zeros; write operations have no effect.
6	PERREN	Parity Error Response Enable. Enables the parity error response functions. When PERREN is 0 and the Am79C971 controller detects a parity error, it only sets the Detected Parity Error bit in the PCI Status register. When PERREN is 1, the Am79C971 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 Am79C971 controller only generates Memory Write cycles.

3 SCYCEN Special Cycle Enable. Read as zero; write operations have no effect. The Am79C971 controller ignores all Special Cycle operations.

2 BMEN Bus Master Enable. Setting BMEN enables the Am79C971 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 Am79C971 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 Am79C971 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 Am79C971 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 1. Since MEMEN also enables the memory mapped access to the Am79C971 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 Am79C971 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 Am79C971 controller detects a parity error. The Am79C971 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 Am79C971 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 Am79C971 controller sets the PERR bit if the target reports a data parity error by asserting the 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 Am79C971 controller and cleared by writing a 1. Writing a 0 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 Am79C971 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 Am79C971 controller and cleared by writing a 1. Writing a 0 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 Am79C971 controller terminates a master cycle with a master abort sequence.

RMABORT is set by the Am79C971 controller and cleared by writing a 1. Writing a 0 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 an Am79C971 master cycle with a target abort sequence.

RTABORT is set by the Am79C971 controller and cleared by writing a 1. Writing a 0 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 Am79C971 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 Am79C971 controller will assert DEVSEL two clock periods after FRAME is asserted.

DEVSEL is read only.

8 DATAPERR — Data Parity Error Detected. DATAPERR is set when the Am79C971 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 Am79C971 controller checks for parity error by sampling the AD[31:0] and C/BE[3:0] and the PAR lines. During the data phase of all memory write commands, the Am79C971 controller checks the PERR input to detect whether the target has reported a parity error.

DATAPERR is set by the Am79C971 controller and cleared by writing a 1. Writing a 0 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 Am79C971 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 Am79C971 controller revision number. The value of this register is 2Xh, with the lower four bits being silicon-revision dependent. The initial revision value will be 21h.

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 Am79C971 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 Am79C971 controller. The value of this register is 00h which identifies the Am79C971 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 Am79C971 controller. The value of this register is 02h which classifies the Am79C971 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 Am79C971 controller will control the bus once it starts its bus mastership period. The time is measured in clock cycles. Every time the Am79C971 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 0. When the system arbiter removes $\overline{\text{GNT}}$ while the counter is non-zero, the Am79C971 controller will continue with its data transfers. It will only release the bus when the counter has reached 0.

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 eight bits of the PCI Latency Timer register are programmable. The host should read the Am79C971 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 Am79C971 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 Am79C971 I/O resources in all of I/O space. It is located at offset 10h in the PCI Configuration Space.

Bit	Name	Description
31-5	IOBASE	I/O base address most significant 27 bits. These bits are written by the host to specify the location of the Am79C971 I/O resources in

		all of I/O space. IOBASE must be written with a valid address before the Am79C971 controller slave I/O mode is turned on by setting the IOEN bit (PCI Command register, bit 0). When the Am79C971 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 Am79C971 controller will drive 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 Am79C971 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 0 in bits 4-2. That indicates an Am79C971 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 Am79C971 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 Am79C971 I/O resources in all of memory space. MEMBASE must be written with a valid address before the Am79C971 controller slave memory mapped I/O mode is turned on by setting the MEMEN bit (PCI Command register, bit 1). When the Am79C971 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 Am79C971 controller will drive 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 Am79C971 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 0 in bit 4. That indicates a Am79C971 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 the 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 Subsystem Vendor ID Register

Offset 2Ch

The PCI Subsystem Vendor ID register is a 16-bit register that together with the PCI Subsystem ID uniquely identifies the add-in card or subsystem the Am79C971 controller is used in. Subsystem Vendor IDs can be obtained from the PCI SIG. A value of 0 (the default) indicates that the Am79C971 controller does not support subsystem identification. The PCI Subsystem Vendor ID is an alias of BCR23, bits 15-0. It is programmable through the EEPROM.

The PCI Subsystem Vendor ID register is located at offset 2Ch in the PCI Configuration Space. It is read only.

PCI Subsystem ID Register

Offset 2Eh

The PCI Subsystem ID register is a 16-bit register that together with the PCI Subsystem Vendor ID uniquely identifies the add-in card or subsystem the Am79C971 controller is used in. The value of the Subsystem ID is up to the system vendor. A value of 0 (the default) indicates that the Am79C971 controller does not support subsystem identification. The PCI Subsystem ID is an alias of BCR24, bits 15-0. It is programmable through the EEPROM.

The PCI Subsystem ID register is located at offset 2Eh in the PCI Configuration Space. It is read only.

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
31-20	ROMBASE	Expansion ROM base address most significant 12 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 Am79C971 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 12 most significant bits of the base address are programmable, the host can map the Expansion ROM on any 1M boundary. When the Am79C971 controller is enabled for Expansion ROM access (ROMEN and MEMEN are set to 1), 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 + 1M - 4, the Am79C971 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.
19-1	ROMSIZE	ROM size. Read as zeros; write operation have no effect. ROMSIZE indicates the maximum size of the Expansion ROM the Am79C971 controller can support. The host can determine the Expansion ROM size by writing FFFF FFFFh to the Expansion ROM Base Address register. It will read back a value of 0 in bit 19-1, indicating an Expansion ROM size of 1M. Note that ROMSIZE only specifies the maximum size of Expansion ROM the Am79C971 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.
0	ROMEN	Expansion ROM Enable. Written by the host to enable access to the Expansion ROM. The Am79C971 controller will only respond to accesses to the Expansion ROM when both ROMEN and MEMEN (PCI Command register, bit 1) are set to 1.

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 Am79C971 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 Am79C971 controller. The PCI Interrupt Line register is not modified by the Am79C971 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 Am79C971 controller is using. The value for the Am79C971 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 Am79C971 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 µs. 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 and which is the time it takes the Am79C971 controller to read/write half of the FIFO. (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 PCI MIN_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 Am79C971 controller can sustain without causing problems to the network activity. The register value specifies the time in units of 1/4 µs. The MAX_LAT register is an alias of BCR22, bits 15-8. It is recommended that BCR22 be programmed to the value of 1818H.

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

The PCI MAX_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 Am79C971 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 Am79C971 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: Am79C971 Controller Status and Control Register

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.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	ERR	Error is set by the OR 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 Am79C971 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, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit BABLM (CSR3, bit 14) is 0. 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 1. Writing a 0 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 Am79C971 controller when the device operates in half-duplex mode and the collision inputs to the AUI or to the GPSI 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 or the GPSI port is active and the Am79C971 controller operates in full-duplex mode. When the 10BASE-T port is selected, for both half-duplex and full-duplex operation, CERR will be set after a transmission if the T-MAU is in Link Fail state. When the MII port is selected, CERR is only reported when the external PHY is operating as a 10BASE-T PHY and if the external 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 1. Writing a 0 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 Am79C971 controller when it has lost an incoming receive frame resulting from a Receive Descriptor not being 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 1 and the mask bit MISSM (CSR3, bit 12) is 0. 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 1. Writing a 0 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 Am79C971 controller when it requests the use of the system interface bus by asserting $\overline{REQ}$ and has not received $\overline{GNT}$ assertion 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 give a MERR after 153.6 μs of bus latency.

When MERR is set, $\overline{INTA}$ is asserted if IENA is 1 and the mask bit MERRM (CSR3, bit 11) is 0. 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 1. Writing a 0 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 Am79C971 controller after the last descriptor of a receive frame has been updated by writing a 0 to the OWNership bit. RINT may also be set when the first descriptor of a receive frame has been updated by writing a 0 to the OWNership bit if the LAPPEN bit of CSR3 has been set to a 1.

When RINT is set, $\overline{INTA}$ is asserted if IENA is 1 and the mask bit RINTM (CSR3, bit 10) is 0.

Read/Write accessible always. RINT is cleared by the host by writing a 1. Writing a 0 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 Am79C971 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, $\overline{INTA}$ is asserted if IENA is 1 and the mask bit TINTM (CSR3, bit 9) is 0.

TINT will not be set if TINTOKD (CSR5, bit 15) is set to 1 and the transmission was successful.

Read/Write accessible always. TINT is cleared by the host by writing a 1. Writing a 0 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 Am79C971 controller after the initialization sequence has completed. When IDON is set, the Am79C971 controller has read the initialization block from memory.

When IDON is set, $\overline{INTA}$ is asserted if IENA is 1 and the mask bit IDONM (CSR3, bit 8) is 0.

Read/Write accessible always. IDON is cleared by the host by writing a 1. Writing a 0 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, RCVCCO, RINT, SINT, SLPINT, TINT, TXSTRT, UINT, STINT, MREINT, MCCINT, MCCIINT, MIIPDTINT, MAPINT and the associated mask or enable bit is programmed to allow the event to cause an interrupt. If IENA is set to 1 and INTR is set, $\overline{INTA}$ will be active. When INTR is set by SINT or SLPINT, $\overline{INTA}$ will be active independent of the state of IENA.

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 = 0, then $\overline{\text{INTA}}$ will be disabled regardless of the state of INTR.

Read/Write accessible always. IENA is set by writing a 1 and cleared by writing a 0. IENA is cleared by H_RESET or S_RESET and setting the STOP bit.

5 RXON — Receive On indicates that the receive function is enabled. RXON is set if DRX (CSR15, bit 0) is set to 0 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 or S_RESET and setting the STOP bit.

4 TXON — Transmit On indicates that the transmit function is enabled. TXON is set if DTX (CSR15, bit 1) is set to 0 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 or S_RESET and 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 reset and no Transmit Descriptor Ring access will occur.

TDMD is required to be set if the TXDPOLL bit in CSR4 is set. Setting TDMD while TXDPOLL = 0 merely hastens the Am79C971 controller's response to a Transmit Descriptor Ring Entry.

Read/Write accessible always. TDMD is set by writing a 1. Writing a 0 has no effect. TDMD will be cleared by the Buffer Management Unit when it fetches a Transmit Descriptor. TDMD is cleared by H_RESET or S_RESET and setting the STOP bit.

2 STOP — STOP assertion disables the chip from all DMA 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 1, by H_RESET or S_RESET. Writing a 0 has no effect. STOP is cleared by setting either STRT or INIT.

1 STRT — STRT assertion enables Am79C971 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 Am79C971 controller initialization will be performed first.

Read/Write accessible always. STRT is set by writing a 1. Writing a 0 has no effect. STRT is cleared by H_RESET, S_RESET, or by setting the STOP bit.

0 INIT — INIT assertion enables the Am79C971 controller to begin the initialization procedure which reads in the initialization block from memory. Setting INIT clears the STOP bit. If STRT and INIT are set together, the Am79C971 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 1. Writing a 0 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
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 0 to align the initialization block to a DWord boundary. This register is aliased with CSR16. 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
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	IADR[31:24]	If SSIZE32 is set (BCR20, bit 8), then the IADR[31:24] bits will be used strictly as the upper 8 bits of the initialization block address. However, if SSIZE32 is reset (BCR20, bit 8), 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 specified by the SSIZE32 = 0 setting will yield only 24 bits of address for the Am79C971 bus master accesses, while the 32-bit hardware for which the Am79C971 controller is intended will require 32 bits of address. Therefore, whenever SSIZE32 = 0, 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 Am79C971 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 = 1, then software will provide 32-bit pointer values for all of the shared software structures - i.e., descriptor bases and buffer addresses, and therefore, IADR[31:24] will not be written to the upper 8 bits of any of these resources, but it will be used as the upper 8 bits of the initialization address. This register is aliased with CSR17. 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. Whenever this register is written, CSR17 is updated with CSR2's contents. 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 (CSR3, bit 6) is set to 0, the transmitter is turned off when an UFLO error occurs (CSR0, TXON = 0).

When DXSUFLO is set to 1, the Am79C971 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 a 1, the LAPPEN bit will cause the Am79C971 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 a 1 also enables the Am79C971 controller to read the STP bit of receive descriptors. The Am79C971 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 Am79C971 controller can write intermediate packet data to buffers whose descriptors do not contain STP bits set to 1. Following the write to the last descriptor used by a packet, the Am79C971 controller will scan through the next descriptor entries to locate the next STP bit that is set to a 1. The Am79C971 controller will begin writing the next packets 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 contain

STP = 1, then some descriptors/ buffers may be skipped in the ring. While performing the search for the next STP bit that is set to 1, the Am79C971 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 Am79C971 controller ownership of the descriptor but also indicate STP = 0, then the Am79C971 controller will reset the OWN bit to 0 in these entries. If a scanned entry indicates host ownership with STP = 0, then the Am79C971 controller will not alter the entry, but will advance to the next entry.

When the STP bit is found to be true, but the descriptor that contains this setting is not owned by the Am79C971 controller, then the Am79C971 controller will stop advancing through the ring entries and begin periodic polling of this entry. When the STP bit is found to be true, and the descriptor that contains this setting is owned by the Am79C971 controller, then the Am79C971 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. The LAPPEN bit will be reset to 0 by H_RESET or S_RESET and will be unaffected by STOP.

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

4 DXMT2PD Disable Transmit Two Part Deferral (see Medium Allocation section in the *Media Access Management* section 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 Contention Resolution section in *Media Access Management* section 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 a 1, big Endian mode is selected. When BSWP is set to 0, little Endian mode is selected.

When big Endian mode is selected, the Am79C971 controller will swap the order of bytes on the AD bus during a data phase on accesses to the FIFOs only. Specifically, AD[31:24] becomes Byte 0, AD[23:16] becomes Byte 1, AD[15:8] becomes Byte 2, and AD[7:0] becomes Byte 3 when big Endian mode is selected. 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 are not affected by the setting of the

		BSWP bit. Descriptor transfers are not affected by the setting of the BSWP bit. RDP, RAP, BDP and PCI configuration space accesses are not affected by the setting of the BSWP bit. Address PROM transfers are not affected by the setting of the BSWP bit. 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 1 when the Am79C971 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 0. Writing a 1 to this bit has no effect on device function. If a 1 is written to this bit, then a 1 will be read back. Existing drivers may write a 1 to this bit for compatibility, but new drivers should write a 0 to this bit and should treat the read value as undefined.
0	RES	Reserved location. The default value of this bit is a 0. Writing a 1 to this bit has no effect on device function. If a 1 is written to this bit, then a 1 will be read back. Existing drivers may write a 1 to this bit for compatibility, but new drivers should write a 0 to this bit and should treat the read value as undefined.

CSR4: Test and Features Control

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.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	EN124	Enable CSR124 access. Setting EN124 to 1 allows the user to write to bits in CSR124, which enables the GPSI interface (GPSIEN, bit 4) and Runt Packet Accept mode (RPA, bit 3). Once these bits are accessed, EN124 must be cleared back to 0. Read/Write accessible always. ENTST is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.
14	DMAPLUS	Writing and reading from this bit has no effect. DMAPLUS is always set to 1.
13	RES	Reserved Location. Written as zero and read as undefined.
12	TXDPOLL	Transmit Disable Transmit Polling. If TXDPOLL is set, the Buffer Management Unit will disable transmit polling. Likewise, if TXDPOLL is cleared, automatic transmit polling is enabled. If TXDPOLL is set, 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 reset. Transmit polling will take place following Receive activities. Read/Write accessible always. TXDPOLL is cleared by H_RESET or S_RESET and is unaffected by 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 field. APAD_XMT will override the programming of the DXMTFCS bit (CSR15, bit 3) and of the ADD_FCS/NO_FCS bit (TMD1, bit 29) for frames shorter than 64 bytes. Read/Write accessible always. APAD_XMT is cleared by H_RESET or S_RESET and is unaffected by 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 the STOP bit.

9 MFCO Missed Frame Counter Overflow is set by the Am79C971 controller when the Missed Frame Counter (CSR112 and CSR114) has wrapped around.

When MFCO is set, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit MFCOM is 0.

Read/Write accessible always. MFCO is cleared by the host by writing a 1. Writing a 0 has no effect. MFCO is cleared by H_RESET, S_RESET, or by setting the STOP bit.

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 1 by H_RESET or S_RESET and is not affected by the STOP bit.

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 1. UINTCMD will be cleared internally after the Am79C971 controller has set UINT to 1.

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 Am79C971 controller after the host has issued a user interrupt command by setting UINTCMD (CSR4, bit 7) to 1.

Read/Write accessible always. UINT is cleared by the host by writing a 1. Writing a 0 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 Am79C971 controller when the Receive Collision Counter (CSR114 and CSR115) has wrapped around.

When RCVCCO is set, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit RCVCCOM is 0.

Read/Write accessible always. RCVCCO is cleared by the host by writing a 1. Writing a 0 has no effect. RCVCCO is cleared by H_RESET, S_RESET, or by setting the STOP bit.

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 1 by H_RESET or S_RESET and is not affected by the STOP bit.

3 TXSTRT Transmit Start status is set by the Am79C971 controller whenever it begins transmission of a frame.

When TXSTRT is set, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit TXSTRTM is 0.

Read/Write accessible always. TXSTRT is cleared by the host by writing a 1. Writing a 0 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 1 by H_RESET or S_RESET and is not affected by the STOP bit.

1	JAB	Jabber Error is set by the Am79C971 controller when the T-MAU exceeds the allowed transmission limit. Jabber can only be asserted in 10BASE-T mode. When JAB is set, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit JABM is 0. Read/Write accessible always. JAB is cleared by the host by writing a 1. Writing a 0 has no effect. JAB is cleared by H_RESET, S_RESET or by setting the STOP bit.
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 1 by H_RESET or S_RESET and is not affected by the STOP bit.

CSR5: Extended Control and Interrupt 1

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.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	TOKINTD	Transmit OK Interrupt Disable. If TOKINTD is set to 1, 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 1. A transmit descriptor with LTINT set to 1 will always cause TINT to be set to 1, independent of the success of the transmission. Read/Write accessible always. TOKINTD is cleared by H_RESET or S_RESET and is unaffected by STOP.
14	LTINTEN	Last Transmit Interrupt Enable. When set to 1, the LTINTEN bit will cause the Am79C971 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 STOP.
13-12	RES	Reserved locations. Written as zeros and read as undefined.
11	SINT	System Interrupt is set by the Am79C971 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 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 1. 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 1. Writing a 0 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 set to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.

9 SLPINT Sleep Interrupt is set by the Am79C971 controller when it comes out of sleep mode.

When SLPINT is set, $\overline{\text{INTA}}$ is asserted if the enable bit SLPINTE is 1. 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 the S_RESET reset generated when entering the sleep mode.

Read/Write accessible always. SLPINT is cleared by the host by writing a 1. Writing a 0 has no effect. SLPINT is cleared by H_RESET and is not affected by S_RESET or 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 set to 0 by H_RESET and is not affected by S_RESET or setting the STOP bit.

7 EXDINT Excessive Deferral Interrupt is set by the Am79C971 controller when the transmitter has experienced Excessive Deferral on a transmit frame, where Excessive Deferral is defined in the ISO 8802-3 (IEEE/ANSI 802.3) standard.

When EXDINT is set, $\overline{\text{INTA}}$ is asserted if the enable bit EXDINTE is 1.

Read/Write accessible always. EXDINT is cleared by the host by writing a 1. Writing a 0 has no effect. EXDINT is cleared by H_RESET and is not affected by S_RESET or 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 set to 0 by H_RESET and is not affected by S_RESET or setting the STOP bit.

5 MPPLBA Magic Packet Physical Logical Broadcast Accept. If MPPLBA is at its default value of 0, the Am79C971 controller will only detect a Magic Packet frame if the destination address of the packet matches the content of the physical address register (PADR). If MPPLBA is set to 1, the destination address of the Magic Packet frame can be unicast, multicast, or broadcast. Note that the setting of MPPLBA only affects the address detection of the Magic Packet frame. The Magic Packet frame's data sequence must be made up of 16 consecutive physical addresses (PADR[47:0]) regardless of what kind of destination address it has.

Read/Write accessible always. MPPLBA is set to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.

4 MPINT Magic Packet Interrupt. Magic Packet Interrupt is set by the Am79C971 controller when the device is in the Magic Packet mode and the Am79C971 controller receives a Magic Packet frame. When MPINT is set to 1, $\overline{\text{INTA}}$ is asserted if IENA (CSR0, bit 6) and the enable bit MPINTE are set to 1.

Read/Write accessible always. MPINT is cleared by the host by writing a 1. Writing a 0 has no affect. MPINT is cleared by H_RESET, S_RESET, or by setting the STOP bit.

3 MPINTE Magic Packet Interrupt Enable. If MPINTE is set to 1, the MPINT bit will be able to set the INTR bit.

Read/Write accessible always. MPINT is cleared to 0 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 the host. The Am79C971 controller will enter the Magic Packet mode when both MPEN and MPMODE are set to 1. Read/Write accessible always. MPEN is cleared to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.
1	MPMODE	Magic Packet Mode. Setting MPMODE to 1 will redefine the $\overline{\text{SLEEP}}$ pin to be a Magic Packet enable pin. The Am79C971 controller will enter the Magic Packet mode when MPMODE is set to 1 and either $\overline{\text{SLEEP}}$ is asserted or MPEN is set to 1. Read/Write accessible always. MPMODE is cleared to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit
0	SPND	Suspend. Setting SPND to 1 will cause the Am79C971 controller to start requesting entrance into suspend mode. The host must poll SPND until it reads back 1 to determine that the Am79C971 controller has entered the suspend mode. Setting SPND to 0 will get the Am79C971 controller out of suspend mode. SPND can only be set to 1 if STOP (CSR0, bit 2) is set to 0. H_RESET, S_RESET or setting the STOP bit will get the Am79C971 controller out of suspend mode. Requesting entrance into the suspend mode by the host depends on the setting of the FASTSPNDE bit (CSR7, bit 15). Refer to the bit description of the FASTSPNDE bit and the Suspend section in *Detailed Functions, Buffer Management Unit* for details. In suspend mode, all of the CSR and BCR registers are accessible. As long as the Am79C971 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 Am79C971 controller will continue at the transmit and receive descriptor ring locations, from where it had left, when it entered the suspend mode. 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 the Am79C971 controller initialization. This field is written during the Am79C971 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 STOP.
11-8	RLEN	Contains a copy of the receive encoded ring length (RLEN) read from the initialization block during Am79C971 controller initialization. This field is written during the Am79C971 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. RLEN is only defined after initialization. These bits are unaffected by H_RESET, S_RESET, or STOP.
7-0	RES	Reserved locations. Read as 0s. Write operations are ignored.

CSR7: Extended Control and Interrupt 2

Certain bits in CSR7 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 CSR7 and write back the value just read to clear the interrupt condition.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	FASTSPNDE	Fast Suspend Enable. When FASTSPNDE is set to 1, the Am79C971 controller performs a fast suspend whenever the SPND bit is set. When a fast suspend is requested, the Am79C971 controller performs a quick entry into the suspend mode. At the time the SPND bit is set, the Am79C971 controller will complete the DMA process of any transmit and/or receive packet that had already begun DMA activity. In addition, any transmit packet that had started transmission will be fully transmitted and any receive packet that had begun reception will be fully received. However, no additional packets will be transmitted or received and no additional transmit or receive DMA activity will begin. Hence, the Am79C971 controller may enter the suspend mode with transmit and/or receive packets still in the FIFOs or external SRAM. When FASTSPNDE is 0 and the SPND bit is set, the Am79C971 controller may take longer before entering the suspend mode. At the time the SPND bit is set, the Am79C971 controller will complete the DMA process of a transmit packet if it had already begun and the Am79C971 controller will completely receive a receive packet if it had already begun. Additionally, all transmit packets stored in the transmit FIFOs and the transmit buffer area in the external SRAM (if one is present) will be transmitted and all receive packets stored in the receive FIFOs, and the receive buffer area in the external SRAM (if one is present) will be transferred into system memory. Since the FIFO and external SRAM contents are flushed, it may take much longer before the Am79C971 controller enters the suspend mode. The amount of time that it takes depends on many factors including the size of the external SRAM, bus latency, and network traffic level. When a write to CSR5 is performed with bit 0 (SPND) set to 1, the value that is simultaneously written to FASTSPNDE is used to determine which approach is used to enter suspend mode. Read/Write accessible always. FASTSPNDE is cleared by H_RESET, S_RESET or by setting the STOP bit.
14	RXFRTG	Receive Frame Tag. When Receive Frame Tag is set to 1, a tag word is put into the receive descriptor supplied by the EADI. See the section *Receive Frame Tagging* for details. This bit is valid only when the EADISEL (BCR2, bit 3) is set to 1. Read/Write accessible always. RXFRTG is cleared by H_RESET. RXFRTG is unaffected by S_RESET or by setting the STOP bit.
13	RDMD	Receive Demand, when set, causes the Buffer Management Unit to access the Receive Descriptor Ring without waiting for the receive poll-time counter to elapse. If RXON is not enabled, RDMD has no meaning and no receive Descriptor Ring access will occur. RDMD is required to be set if the RXDPOLL bit in CSR7 is set. Setting RDMD while RXDPOLL = 0 merely hastens the Am79C971 controller's response to a receive Descriptor Ring Entry.

Read/Write accessible always. RDMD is set by writing a 1. Writing a 0 has no effect. RDMD will be cleared by the Buffer Management Unit when it fetches a receive Descriptor. RDMD is cleared by H_RESET. RDMD is unaffected by S_RESET or by setting the STOP bit.

12 RXDPOLL Receive Disable Polling. If RXDPOLL is set, the Buffer Management Unit will disable receive polling. Likewise, if RXDPOLL is cleared, automatic receive polling is enabled. If RXDPOLL is set, RDMD bit in CSR7 must be set in order to initiate a manual poll of a receive descriptor. Receive Descriptor Polling will not take place if RXON is reset.

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

11 STINT Software Timer Interrupt. The Software Timer interrupt is set by the Am79C971 controller when the Software Timer counts down to 0. The Software Timer will immediately load the STVAL (BCR 31, bits 5-0) into the Software Timer and begin counting down.

When STINT is set to 1, $\overline{\text{INTA}}$ is asserted if the enable bit STINTE is set to 1.

Read/Write accessible always. STINT is cleared by the host by writing a 1. Writing a 0 has no effect. STINT is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

10 STINTE Software Timer Interrupt Enable. If STINTE is set, the STINT bit will be able to set the INTR bit.

Read/Write accessible always. STINTE is set to 0 by H_RESET and is not affected by S_RESET or setting the STOP bit

9 MREINT MII Management Read Error Interrupt. The MII Read Error interrupt is set by the Am79C971 controller to indicate that the currently read register from the external PHY is invalid. The contents of BCR34 are incorrect and that the operation should be performed again. The indication of an incorrect read comes from the PHY. During the read turnaround time of the MII management frame the external PHY should drive the MDIO pin to a LOW state. If this does not happen, it indicates that the PHY and the Am79C971 controller have lost synchronization.

When MREINT is set to 1, $\overline{\text{INTA}}$ is asserted if the enable bit MREINTE is set to 1.

Read/Write accessible always. MREINT is cleared by the host by writing a 1. Writing a 0 has no effect. MREINT is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

8 MREINTE MII Management Read Error Interrupt Enable. If MREINTE is set, the MREINT bit will be able to set the INTR bit.

Read/Write accessible always. MREINTE is set to 0 by H_RESET and is not affected by S_RESET or setting the STOP bit

7 MAPINT MII Management Auto-Poll Interrupt. The MII Auto-Poll interrupt is set by the Am79C971 controller to indicate that the currently read status does not match the stored previous status indicating a change in state for the external PHY. A change in the Auto-Poll Access Method (BCR32, Bit 10) will reset the shadow register and will not cause an interrupt on the first access from the Auto-Poll section. Subsequent accesses will generate an interrupt if the shadow register and the read register produce differences.

When MAPINT is set to 1, $\overline{\text{INTA}}$ is asserted if the enable bit MAPINTE is set to 1.

Read/Write accessible always. MAPINT is cleared by the host by writing a 1. Writing a 0 has no effect. MAPINT is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

6 MAPINTE MII Auto-Poll Interrupt Enable. If MAPINTE is set, the MAPINT bit will be able to set the INTR bit.

Read/Write accessible always. MAPINTE is set to 0 by H_RESET and is not affected by S_RESET or setting the STOP bit

5 MCCINT MII Management Command Complete Interrupt. The MII Management Command Complete Interrupt is set by the Am79C971 controller when a read or write operation to the MII Data Port (BCR34) is complete.

When MCCINT is set to 1, $\overline{\text{INTA}}$ is asserted if the enable bit MCCINTE is set to 1.

Read/Write accessible always. MCCINT is cleared by the host by writing a 1. Writing a 0 has no effect. MCCINT is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

4 MCCINTE MII Management Command Complete Interrupt Enable. If MCCINTE is set to 1, the MCCINT bit will be able to set the INTR bit when the host reads or writes to the MII Data Port (BCR34) only. Internal MII Management Commands will not generate an interrupt. For instance Auto-Poll state machine generated MII management frames will not generate an interrupt upon completion unless there is a compare error which get reported through the MAPINT (CSR7, bit 6) interrupt or the MCCIINTE is set to 1.

Read/Write accessible always. MCCINTE is set to 0 by H_RESET and is not affected by S_RESET or setting the STOP bit.

3 MCCIINT MII Management Command Complete Internal Interrupt. The MII Management Command Complete Interrupt is set by the Am79C971 controller when a read or write operation on the MII management port is complete from an internal operation. Examples of internal operations are Auto-Poll or MII Management Port generated MII management frames. These are normally hidden to the host.

When MCCIINT is set to 1, $\overline{\text{INTA}}$ is asserted if the enable bit MCCINTE is set to 1.

Read/Write accessible always. MCCIINT is cleared by the host by writing a 1. Writing a 0 has no effect. MCCIINT is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

2 MCCIINTE MII Management Command Complete Internal Interrupt Enable. If MCCIINTE is set to 1, the MCCIINT bit will be able to set the INTR bit when the internal state machines generate MII management frames. For instance, when MCCIINTE is set to 1 and the Auto-Poll state machine generates a MII management frame, the MCCIINT will set the INTR bit upon completion of the MII management frame regardless of the comparison outcome.

Read/Write accessible always. MCCIINTE is set to 0 by H_RESET and is not affected by S_RESET or setting the STOP bit.

1	MIIPDTINT	MII PHY Detect Transition Interrupt. The MII PHY Detect Transition Interrupt is set by the Am79C971 controller whenever the MIIPD bit (BCR32, bit 14) transitions from 0 to 1 or vice versa. Read/Write accessible always. MIIPDTINT is cleared by the host by writing a 1. Writing a 0 has no effect. MIIPDTINT is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
0	MIIPDTINTE	MII PHY Detect Transition Interrupt Enable. If MIIPDTINTE is set to 1, the MIIPDTINT bit will be able to set the INTR bit. Read/Write accessible always. MIIPDTINTE is set to 0 by H_RESET and is not affected by S_RESET or setting the STOP bit.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

CSR15: Mode

This register's fields are loaded during the Am79C971 controller initialization routine with the corresponding Initialization Block values, or when a direct register write has been performed on this register.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	PROM	Promiscuous Mode. When PROM = 1, 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, disables the Am79C971 controller from receiving broadcast messages. Used for protocols that do not support broadcast addressing, except as a function of multicast. DRCVBC is cleared by activation of H_RESET or S_RESET (broadcast messages will be received) and is unaffected by STOP. Read/Write accessible only when either the STOP or the SPND bit is set.
13	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the Am79C971 controller will be disabled. Frames addressed to the nodes individual physical address will not be recognized. Read/Write accessible only when either the STOP or the SPND bit is set.
12	DLNKTST	Disable Link Status. When DLNKTST = 1, monitoring of Link Pulses is disabled. When DLNKTST = 0, 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 = 1, the

10BASE-T receive polarity reversal algorithm is disabled. Likewise, when DAPC = 0, 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, bit 2.

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

9 LRT (TMAU mode) Low Receive Threshold. When LRT = 1, 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 to 312 mV peak.

When LRT = 0, the unsquelch threshold for the RXD circuit will be the standard 10BASE-T value, 300 mV to 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 STOP.

TSEL (AUI mode) Transmit Mode Select. TSEL controls the levels at which the AUI drivers rest when the AUI transmit port is idle. When TSEL = 0, DO+ and DO- yield “zero” differential to operate transformer coupled loads (Ethernet 2 and IEEE 802.3). When TSEL = 1, 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.

8-7 PORTSEL[1:0] Port Select bits allow for software controlled selection of the network medium. See Table 26.

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

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 STOP.

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.

Table 26. Network Port Configuration.

PORTSEL [1:0]	ASEL (BCR2[1])	Link Status (10BASE-T)	MII Status (BCR32[14])	Network Port
XX	1	Fail	0	AUI
XX	1	Pass	0	10BASE-T
XX	1	Don't Care	1	MII
00	0	Don't Care	Don't Care	AUI
01	0	Don't Care	Don't Care	10BASE-T
10	0	Don't Care	Don't Care	GPSI
11	0	Don't Care	Don't Care	MII

5 DRTY Disable Retry. When DRTY is set to 1, the Am79C971 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 set to 0, the Am79C971 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 Am79C971 controller must be in internal loopback for FCOLL to be valid. If FCOLL = 1, a collision will be forced during loopback transmission attempts, which will result in a Retry Error. If FCOLL = 0, the Force Collision logic will be disabled. FCOLL is defined after the initialization block is read.

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

3 DXMTFCS Disable Transmit CRC (FCS). When DXMTFCS is set to 0, the transmitter will generate and append an FCS to the transmitted frame. When DXMTFCS is set to 1, no FCS is generated or sent with the transmitted frame. DXMTFCS is overridden when ADD_FCS is set in TMD1.

When APAD_XMT (CSR4, bit 11) is set to 1, the setting of DXMTFCS has no effect on frames shorter than 64 bytes.

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 was called DTCR in the LANCE (Am7990) device.

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

2 LOOP Loopback Enable allows the Am79C971 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 = 1, loopback is enabled. In combination with INTL and MENDECL, various loopback modes are defined as follows in Table 27. Refer to *Loop Back Operation* section for more details.

Table 27. Loopback Configuration for AUI

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. LOOP is cleared by H_RESET or S_RESET and is unaffected by STOP.
1	DTX	Disable Transmit results in Am79C971 controller not accessing the Transmit Descriptor Ring and, therefore, no transmissions are attempted. DTX = 0, will set TXON bit (CSR0 bit 4) if STRT (CSR0 bit 1) is asserted. Read/Write accessible only when either the STOP or the SPND bit is set.
0	DRX	Disable Receiver results in the Am79C971 controller not accessing the Receive Descriptor Ring and, therefore, all receive frame data are ignored. DRX = 0, will set RXON bit (CSR0 bit 5) if STRT (CSR0 bit 1) is asserted. 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 Am79C971 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 STOP.

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 Am79C971 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 STOP.

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 Am79C971 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 STOP.

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 Am79C971 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 STOP.

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 Am79C971 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 STOP.

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 Am79C971 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 STOP.

CSR24: Base Address of Receive 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 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 STOP.

CSR25: Base Address of Receive 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 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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

CSR30: Base Address of Transmit 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 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 STOP.

CSR31: Base Address of Transmit 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 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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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 STOP.

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

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 STOP.

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 STOP.

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 STOP.

CSR46: Transmit Poll Time Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TXPOLL	Transmit Poll Time Counter. This counter is incremented by the Am79C971 controller microcode and is used to trigger the transmit descriptor ring polling operation of the Am79C971 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 STOP.

CSR47: Transmit Polling Interval

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TXPOLLINT	Transmit Polling Interval. This register contains the time that the Am79C971 controller will wait between successive polling operations. The TXPOLLINT value is expressed as the two's complement of the desired interval, where each bit of TXPOLLINT represents 1 clock period of time. TXPOLLINT[3:0] are ignored. (TXPOLLINT[16] is implied to be a one, so TXPOLLINT[15] is significant and does not represent the sign of the two's complement TXPOLLINT value.)

The default value of this register is 0000b. This corresponds to a polling interval of 65,536 clock periods (1.966 ms when CLK = 33 MHz). The TXPOLLINT value of 0000b 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, then the correct procedure is to first set INIT only in CSR0. Then, when the initialization sequence is complete, the user must set STOP (CSR0, bit 2). 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, then it is imperative that the user also writes all zeros 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 STOP.

CSR48: Receive Poll Time Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RXPOLL	Receive Poll Time Counter. This counter is incremented by the Am79C971 controller microcode and is used to trigger the receive descriptor ring polling operation of the Am79C971 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 STOP.

CSR49: Receive Polling Interval

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RXPOLLINT	Receive Polling Interval. This register contains the time that the Am79C971 controller will wait between successive polling operations. The RXPOLLINT value is expressed as the two's complement of the desired interval, where each bit of RXPOLLINT approximately represents one clock time period. RXPOLLINT[3:0] are ignored. (RXPOLLINT[16] is implied to be a 1, so RXPOLLINT[15] is significant and does not represent the sign of the two's complement RXPOLLINT value.)

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 RXPOLLINT value of 0000h is created during the microcode initialization routine and, therefore, might not be seen when reading CSR49 after H_RESET or S_RESET.

If the user desires to program a value for RXPOLLINT other than the default, then the correct procedure is to first set INIT only in CSR0. Then, when the initialization sequence is complete, the user must set STOP (CSR0, bit 2). Then the user may write to CSR49 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, then it is imperative that the user also writes all zeros to CSR49 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 STOP.

CSR58: Software Style

This register is an alias of the location BCR20. Accesses to and from this register are equivalent to accesses to BCR20.

Bit	Name	Description
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 1, the BPE bits (RMD1 and TMD1, bit 23) start having a meaning. BPE will be set in the descriptor associated with the buffer that was accessed when a data parity error occurred. 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 2 or 3 to program the Am79C971 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 Am79C971 controller is the target of the transfer. Read anytime, 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 STOP.
9	RES	Reserved locations. Written as zeros and read as undefined.
8	SSIZE32	Software Size 32 bits. When set, this bit indicates that the Am79C971 controller utilizes 32-bit software structures for the initialization block and the transmit and receive descriptor entries. When cleared, this bit indicates that the Am79C971 controller utilizes 16-bit software structures for the initialization block and the transmit and receive descriptor entries. In this mode, the Am79C971 controller is backwards compatible with the Am7990 LANCE and Am79C960 PCnet-ISA controllers. The value of SSIZE32 is determined by the Am79C971 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 0) and is not affected by S_RESET or STOP. If SSIZE32 is reset, 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 Am79C971 controller. This action is required, since the 16-bit software structures specified by the SSIZE32 = 0 setting will yield only 24 bits of address for the Am79C971 controller bus master accesses. If SSIZE32 is set, then the software structures that are common to the Am79C971 controller and the host system will supply a full 32 bits for each address pointer that is needed by the Am79C971 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 ad-

dress 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 defined width for I/O resources. I/O resource 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 Am79C971 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 Am79C971 controller CSR bits and BCR bits and all descriptor, buffer, and initialization block entries not cited in Table 26 are unaffected by the Software Style selection and are, therefore, always fully functional as specified in the CSR and BCR sections.

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 STOP.

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. Am79C971 controller has the capability to 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 STOP.

Table 26. Software Styles

SWSTYLE [7:0]	Style Name	SSIZE32	Initialization Block Entries	Descriptor Ring Entries
00h	LANCE/ PCnet-ISA controller	0	16-bit software structures, non-burst or burst access	16-bit software structures, non-burst access only
01h	RES	1	RES	RES
02h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only
03h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only
All Other	Reserved	Undefined	Undefined	Undefined

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 Am79C971 controller has the capability to 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 STOP.

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 STOP.

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 STOP.

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 Am79C971 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 STOP.

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 Am79C971 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 STOP.

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 STOP.

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 STOP.
7-0	RES	Reserved locations. Read and written as zeros. Accessible only when either the STOP or the SPND bit is set.

CSR72: Receive Ring Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCVRC	Receive 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 STOP.

CSR74: Transmit Ring Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	XMTRC	Transmit 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 STOP.

CSR76: Receive Ring Length

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCVRL	Receive Ring Length. Contains the two's complement of the receive descriptor ring length. This register is initialized during the Am79C971 controller initialization routine based on the value in the RLEN field of the initialization block. However, this register can be manually altered. The actual receive ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR78: Transmit Ring Length

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	XMTRL	Transmit Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the Am79C971 controller initialization routine based on the value in the TLEN field of the initialization block. However, this register can be manually altered. The actual transmit ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. Read/Write accessible only when either the STOP or the SPND bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR80: DMA Transfer Counter and FIFO Threshold Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-14	RES	Reserved locations. Written as zeros and read as undefined.
13-12	RCVFW[1:0]	Receive FIFO Watermark. RCVFW controls the point at which receive DMA is requested in relation to the number of received bytes in the Receive FIFO. RCVFW specifies the number of bytes which must be present (once the frame has been verified as a non-runt) before receive

DMA is requested. Note however that, if the network interface is operating in half-duplex mode, in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. 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 RCVFW threshold is reached, or a complete valid receive frame is detected (regardless of length). When the FDRPAD (BCR9, bit 2) is set and the Am79C971 controller is in full-duplex mode, in order to receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively disables the runt packet accept feature in full duplex.

When operating in the NO-SRAM mode (no SRAM present), the Bus Receive FIFO and the MAC Receive operate like a single FIFO and the watermark value selected by RCVFW[1:0] sets the number of bytes that must be present in the FIFO before receive DMA is requested.

NOTE: A "No SRAM configuration" is only valid for 10Mb mode. In 100Mb mode, SRAM is mandatory and must always be used.

When operating with an external SRAM, the Bus Receive FIFO, and the MAC Receive FIFO operate independently on the bus side and MAC side of the external SRAM, respectively. In this case, the watermark value set by RCVFW[1:0] sets the number of bytes that must be present in the Bus Receive FIFO only. See Table 27.

Table 27. Receive Watermark Programming

RCVFW[1:0]	Bytes Received
00	16
01	64
10	112
11	Reserved

Read/Write accessible only when either the STOP or the SPND bit is set. RCVFW[1:0] is set to a value of 01b (64 bytes) after H_RESET or S_RESET and is unaffected by STOP.

11-10 XMTSP[1:0] Transmit Start Point. XMTSP controls the point at which preamble transmission attempts to commence in relation to the number of bytes written to the MAC Transmit FIFO for the current transmit frame. When the entire frame is in the MAC Transmit FIFO, transmission 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 rewritten to 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 Am79C971 controller can overwrite the beginning of the frame as soon as the data is transmitted, because no collision handling is required in these modes.

Note that when an external SRAM is being used, if the NOUFLO bit (BCR18, bit 11) is set to 1, there is the additional restriction that the complete transmit frame must be DMA'd into the Am79C971 controller and reside within a combination of the Bus Transmit FIFO, the external SRAM, and the MAC Transmit FIFO.

When an external SRAM is used, SRAM_SIZE > 0, there is a restriction that the number of bytes written is a combination of bytes written into the Bus Transmit FIFO and the MAC Transmit FIFO. The Am79C971 controller supports a mode that will wait until a full packet is available before commencing with the transmission of preamble. This mode is useful in a system where high latencies cannot be avoided. See Table 28.

Table 28. Transmit Start Point Programming

XMTSP[1:0]	SRAM_SIZE	Bytes Written
00	0	20
01	0	64
10	0	128
11	0	248
00	>0	44
01	>0	64
10	>0	128
11	>0	Full Packet

NOTE: A "No SRAM configuration" is only valid for 10Mb mode. In 100Mb mode, SRAM is mandatory and must always be used.

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 STOP.

9-8 XMTFW[1:0] Transmit FIFO Watermark. XMTFW specifies the point at which transmit DMA is requested, based upon the number of bytes that could be written to the Transmit FIFO without FIFO overflow. Transmit DMA is requested at any time when the number of bytes specified by XMTFW could be written to the FIFO without causing Transmit FIFO overflow, and the internal microcode engine has reached a point where the Transmit FIFO is checked to determine if DMA servicing is required.

When operating in the NO-SRAM mode (no SRAM present), SRAM_SIZE set to 0, the Bus Transmit FIFO and the MAC Transmit FIFO operate like a single FIFO and the watermark value selected by XMTFW[1:0] sets the number of FIFO byte locations that must be available in the FIFO before receive DMA is requested.

NOTE: A "No SRAM configuration" is only valid for 10Mb mode. In 100Mb mode, SRAM is mandatory and must always be used.

When operating with an external SRAM, the Bus Transmit FIFO and the MAC Transmit FIFO operate independently on the bus side and MAC side of the external SRAM, respectively. In this case, the watermark value set by XMTFW[1:0] sets the number of FIFO byte locations that must be available in the Bus Transmit FIFO. See Table 29.

Table 29. Transmit Watermark Programming

XMTFW[1:0]	Bytes Available
00	16
01	64
10	108
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 STOP.

7-0 DMATC[7:0] DMA Transfer Counter. Writing and reading to this field has no effect. Use MAX_LAT and MIN_GNT in the PCI configuration space.

CSR82: Transmit Descriptor Address Pointer Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.

15-0	TXDAPL	Contains the lower 16 bits of the transmit descriptor address corresponding to the last buffer of the previous transmit frame. If the previous transmit frame did not use buffer chaining, then TXDAPL contains the lower 16 bits of the previous frame's transmit descriptor address. When both the STOP or SPND bits are cleared, this register is updated by Am79C971 controller immediately before a transmit descriptor write. Read accessible always. Write accessible through the PXDAL bits (CSR60) when the STOP or SPND bit is set. TXDAPL is set to 0 by H_RESET and are unaffected by S_RESET or STOP.

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 increment commands to increment the memory address for sequential operations. The DMABAL register is undefined until the first Am79C971 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 STOP.

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 increment commands to increment the memory address for sequential operations. The DMABAU register is undefined until the first Am79C971 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 STOP.

CSR86: Buffer Byte Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	RES	Reserved. Read and written with ones.
11-0	DMABC	DMA Byte Count Register. Contains the two's complement of the current size of the remaining 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 STOP.

CSR88: Chip ID Register Lower

Bit	Name	Description
31-28	VER	Version. This 4-bit pattern is silicon-revision dependent. Read accessible only when either the STOP or the SPND bit is set. VER is read only. Write operations are ignored.
27-12	PARTID	Part number. The 16-bit code for the Am79C971 controller is 0010 0110 0010 0011b (2623h). This register is exactly the same as the Device ID register in the JTAG description. It is, however,

		a different ID as that stored in the Device ID register in the PCI configuration space. Read accessible only when either the STOP or the SPND bit is set. VER is read only. 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 only when either the STOP or the SPND bit is set. VER is read only. MANFID is read only. Write operations are ignored.
0	ONE	Always a logic 1. Read accessible only when either the STOP or the SPND bit is set. VER is read only. 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 only when either the STOP or the SPND bit is set. VER is read only. VER is read only. Write operations are ignored.
11-0	PARTIDU	Upper 12 bits of the Am79C971 controller part number, i.e., 0010 0110 0010b (262h). Read accessible only when either the STOP or the SPND bit is set. VER is read only. PARTIDU is read only. Write operations are ignored.

CSR92: Ring Length Conversion

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCON	Ring Length Conversion Register. This register performs a ring length conversion from an encoded value as found in the initialization block to a two's complement value used for internal counting. By writing bits 15-12 with an encoded ring length, a two's complemented value is read. The RCON 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 STOP.

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 $\overline{REQ}$ and assertion of $\overline{GNT}$) that a system may insert into an Am79C971 controller master transfer. If this value of bus latency is exceeded, then a MERR will be indicated in CSR0, bit 11, 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 0 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 STOP.

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 0 from the value 65535. The MFCO bit of CSR4 (bit 8) will be set each time that this occurs. 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 encountered by the receiver since the last reset of the counter. RCC will roll over to a count of 0 from the value 65535. The RCVCCO bit of CSR4 (bit 5) will be set each time that this occurs. 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.
15-1	RES	Reserved locations. Written as zeros and read as undefined.
0	RCVALGN	Receive Packet Align. When set, this bit forces the data field of ISO 8802-3 (IEEE/ANSI 802.3) packets to align to 0. MOD 4 address boundaries (i.e., DWord aligned addresses). 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 Am79C971 controller simply inserts two bytes of random data at the beginning of the receive packet (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

This register is used to place the Am79C971 controller into various test modes. The Runt Packet Accept is the only user accessible test mode. All other test modes are for AMD internal use only.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-4	RES	Reserved locations. Written as zeros and read as undefined.
3	RPA	Runt Packet Accept. This bit forces the Am79C971 controller to accept runt packets (packets shorter than 64 bytes). Read accessible always; write accessible only when STOP is set to 1. RPA is cleared by H_RESET or S_RESET and is not affected by STOP.
2-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR125: MAC Enhanced Configuration Control

Bit	Name	Description

31-16 RES — Reserved locations. Written as zeros and read as undefined.

15-8 IPG — Inter Packet Gap. Changing IPG allows the user to program the Am79C971 controller for aggressiveness on a network. By changing the default value of 96 bit times (6oh) the user can adjust the fairness or aggressiveness of the Am79C971 MAC on the network. By programming a lower number of bit times other then the ISO/IEC 8802-3 standard requires, the Am79C971 MAC will become more aggressive on the network. This aggressive nature will give rise to the Am79C971 controller possibly "capturing the network" at times by forcing other less aggressive nodes to defer. By programming a larger number of bit times, the Am79C971 MAC will become less aggressive on the network and may defer more often than normal. The performance of the Am79C971 controller may decrease as the IPG value is increased from the default value.

***Note**: Programming of the IPG should be done in nibble intervals instead of absolute bit times. The decimal and hex values do not match due to delays in the part used to make up the final IPG. Changes should be added or subtracted from the provided hex value on a one-for-one basis.*

CAUTION: Use this parameter with care. By lowering the IPG below the ISO/IEC 8802-3 standard 96 bit times, the Am79C971 controller can interrupt normal network behavior.

Read accessible always. Write accessible when the STOP bit is set to 1. IPG is set to 60h (96 Bit times) by H_RESET or S_RESET and is not affected by STOP.

7-0 IFS1 — InterFrameSpacingPart1. Changing IFS1 allows the user to program the value of the InterFrameSpacePart1 timing. The Am79C971 controller sets the default value at 60 bit times (3ch). See the subsection on *Medium Allocation* in the section *Media Access Management* for more details. The equation for setting IFS1 when IPG ≥ 96 bit times is:

IFS1 = IPG - 36 bit times

***Note**: Programming of the IPG should be done in nibble intervals instead of absolute bit times due to the MII. The decimal and hex values do not match due to delays in the part used to make up the final IPG.*

Changes should be added or subtracted from the provided hex value on a one-for-one basis. Due to changes in synchronization delays internally through different network ports, the IFS1 can be off by as much as +12 bit times.

Read accessible always. Write accessible only when the SPND bit or the STOP bit is set to 1. IFS1 is set to 3ch (60 bit times) by H_RESET or S_RESET and is not affected by STOP.

Bus Configuration Registers

The Bus Configuration Registers (BCR) are used to program the configuration of the bus interface and other special features of the Am79C971 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. See Table 30.

All BCR registers are 16 bits in width in Word I/O mode (DWIO = 0, BCR18, bit 7) and 32 bits in width in DWord I/O mode (DWIO = 1). The upper 16 bits of all BCR registers is 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. 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. BCR2 and BCR34 are not observable without first being 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 have no effect on the operation of the Am79C971 controller.

Writes to those registers marked as “Reserved” will have no effect. Reads from these locations will produce undefined values.

Table 30. BCR Registers

RAP	Mnemonic	Default	Name	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	LED0	00C0h	LED0 Status	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
23	PCISID	0000h	PCI Subsystem ID	No	Yes
24	PCISVID	0000h	PCI Subsystem Vendor ID	No	Yes
25	SRAMSIZ	0000h	SRAM Size	Yes	Yes
26	SRAMB	0000h	SRAM Boundary	Yes	Yes
27	SRAMIC	0000h	SRAM Interface Control	Yes	Yes
28	EBADDRL	N/A	Expansion Bus Address Lower	Yes	No
29	EBADDRU	N/A	Expansion Bus Address Upper	Yes	No
30	EBD	N/A	Expansion Bus Data Port	Yes	No
31	STVAL	FFFFh	Software Timer Value	Yes	No
32	MIICAS	0000h	MII Control and Status	Yes	Yes
33	MIIADDR	0000h	MII Address	Yes	Yes
34	MIIMDR	N/A	MII Management Data	Yes	No
35	PCIVID	1022h	PCI Vendor ID	No	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 Am79C971 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 Am79C971 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: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	RES	Reserved location. Written and read as zeros.
14	TMAULOOP	When set, this bit allows external loopback packets to pass onto 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 reset to 0 by H_RESET and is unaffected by S_RESET or STOP.
13	RES	Reserved location. Written and read as zero.
12	LEDPE	LED Program Enable. When LEDPE is set to 1, programming of the LED0 (BCR4), LED1 (BCR5), LED2 (BCR6), and LED3 (BCR7) registers is enabled. When LEDPE is cleared to 0, programming of LED0 (BCR4), LED1 (BCR5), LED2 (BCR6), and LED3 (BCR7) registers is disabled. Writes to those registers will be ignored. Read/Write accessible always. LEDPE is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
11-9	RES	Reserved locations. Written and read as zeros.
8	APROMWE	Address PROM Write Enable. The Am79C971 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 1, then write access to the shadow RAM will be enabled. Read/Write accessible always. APROMWE is cleared to 0 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 0, 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 Am79C971 controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is tri-stated by the Am79C971 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 1, 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 Am79C971 controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is driven to a low level by the Am79C971 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 1 when the Am79C971 controller is used in a PCI bus application.

Read/Write accessible always. INTLEVEL is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

6-4 RES — Reserved locations. Written as zeros and read as undefined.

3 EADISEL — EADI Select. When set to 1, 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, $\overline{\text{LED2}}$ becomes SRDCLK, and $\overline{\text{SLEEP}}$ becomes $\overline{\text{EAR}}$. See the section on *External Address Detection* for more details.

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 1 and the $\overline{\text{SLEEP}}$ pin is asserted, the Am79C971 controller goes into snooze mode. If AWAKE is cleared to 0 and the $\overline{\text{SLEEP}}$ pin is asserted, the Am79C971 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 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

1 ASEL — Auto Select. When set, the Am79C971 controller will automatically select the operating media interface port. If ASEL has been set to a 1, then when the MIIPD bit (BCR32, bit 14) is 1, the MII port is selected. If the MIIPD bit is 0, and then, if the 10BASE-T transceiver is in the link pass state (due to receiving valid frame data and/or Link Test pulses or the DLNKTST bit is set), the 10BASE-T port will be used. If the MIIPD bit is 0 and the 10BASE-T port is in the Link Fail state, the AUI port will be used. Switching between the ports will not occur during transmission to avoid any type of fragment generation.

The network port configurations are found in Table 31.

When ASEL is set to 1 and the MIIPD bit is 0, Link Beat Pulses will be transmitted on the 10BASE-T port, regardless of the state of Link Status. When ASEL is reset to 0, 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 set to a 0, then the selected network port will be determined by the settings of the PORTSEL bits of CSR15. When ASEL is set to 1, the selected network port may be determined through software by reading the MIIPD bit and, if MIIPD is 0, reading the link status through BCR4 or another LED Control register if it is programmed for link status. The PORTSEL[1:0] bits do not reflect the selected network port when ASEL is 1. Read/Write accessible always. ASEL is set to 1 by H_RESET and is unaffected by S_RESET or STOP.

0 XMAUSEL — Reserved location. Read/Write accessible always. This reserved location is cleared by H_RESET and is unaffected by S_RESET or STOP. Writing a 1 to this bit has no effect on the operation of the Am79C971 controller.

Table 31. Network Port Configuration

PORTSEL[1:0]	ASEL (BCR2[1])	Link Status (of 10BASE-T)	MII Status (BCR32[14])	Network Port
XX	1	Fail	0	AUI
XX	1	Pass	0	10BASE-T
XX	1	Don't Care	1	MII
00	0	Don't Care	Don't Care	AUI
01	0	Don't Care	Don't Care	10BASE-T
10	0	Don't Care	Don't Care	GPSI
11	0	Don't Care	Don't Care	MII

BCR4: LED0 Status

BCR4 controls the function(s) that the $\overline{\text{LED0}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR4 defaults to Link Status (LNKST) with pulse stretcher enabled (PSE = 1) and is fully programmable.

Note: *When LEDPE (BCR2, bit 12) is set to 1, programming of the LED0 Status register is enabled. When LEDPE is cleared to 0, programming of the LED0 register is disabled. Writes to those registers will be ignored.*

Note: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
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 1 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 the LED 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 STOP.
14	LEDPOL	LED Polarity. When this bit has the value 0, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false (i.e., 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 1, then the LED pin will be driven to a HIGH level 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 (i.e., 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 setting the STOP bit.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then 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 setting the STOP bit.

12 100E

100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C971 controller is operating at 100 Mbps mode. The indication is valid with both the internal and external PHYs.

Read/Write accessible always. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

11 MIISE

Media Independent Interface Selected Enable. Indicates when the MII interface is selected. This will be set when either the Management Port State Machine is selecting the MII or when ASEL (BCR2, bit1) is disabled and PORTSEL (CSR15, bits 8-7) selects the MII. This could control relays to switch in and out appropriate filters or could control an external PHY when sharing an RJ45 connector.

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

10 DXCVRCTL

DXCVR Control. When the AUI interface is the active network port, DXCVRCTL controls the assertion of the $\overline{\text{LED0}}$ output. The polarity of the asserted state is controlled by the LEDPOL bit (BCR4, bit 14). The $\overline{\text{LED0}}$ 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 Am79C971 AUI port. When DXCVRCTL is set to 1, the $\overline{\text{LED0}}$ 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 LEDPOL is cleared to 0). When DXCVRCTL is cleared to 0, the $\overline{\text{LED0}}$ 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.

9 MPSE

Magic Packet Status Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when Magic Packet frame mode is enabled and a Magic Packet frame is detected on the network.

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

8 FDLSE

Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LEDOUT signal when the Am79C971 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C971 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal.

When the 10BASE-T port is active, a value of 1 is passed to the LEDOUT signal whenever the Link Test Function (described in the T-MAU section) detects a Link Pass state and the FDEN (BCR9, bit 0) bit is set. When the AUI port is active, a value of 1 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 1). When the MII port is active, a value of 1 is passed to the LEDOUT signal whenever full-duplex operation on the MII port is enabled (FDEN bit in BCR9 is set to 1).

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

7	PSE	Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher. Read/Write accessible always. PSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.
6	LNKSE	Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register when the T-MAU mode is in Link Pass state. When the T-MAU is in Link Fail state, a value of 0 is passed to the LEDOUT bit. This bit does not reflect the link status of the external PHY. The function of this bit is masked if the 10BASE-T port is operating in full-duplex mode. This allows a Half-Duplex Link Status LED and a Full-Duplex Link Status LED at the same time. Read/Write accessible always. LNKSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.
5	RCVME	Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when 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 setting the STOP bit.
4	XMTE	Transmit Status Enable. When this bit is set, a value of 1 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 setting the STOP bit.
3	RXPOLE	Receive Polarity Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the polarity of the RXD± pair has not been 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 setting the STOP bit.
2	RCVE	Receive Status Enable. When this bit is set, a value of 1 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 setting the STOP bit.
1	JABE	Jabber Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the Am79C971 controller is jabbering on the network. Read/Write accessible always. JABE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
0	COLE	Collision Status Enable. When this bit is set, a value of 1 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 ports 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 setting the STOP bit.

BCR5: LED1 Status

BCR5 controls the function(s) that the $\overline{\text{LED1}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR5 defaults to Receive Status (RCV) with pulse stretcher enabled (PSE = 1) and is fully programmable.

Note: *When LEDPE (BCR2, bit 12) is set to 1, programming of the LED1 Status register is enabled. When LEDPE is cleared to 0, programming of the LED1 register is disabled. Writes to those registers will be ignored.*

Note: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
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 1 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 the LED 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 STOP.
14	LEDPOL	LED Polarity. When this bit has the value 0, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false (i.e., 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 1, then the LED pin will be driven to a HIGH level 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 (i.e., 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 setting the STOP bit.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then 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 setting the STOP bit.
12	100E	100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C971 controller is operating at 100 Mbps mode. The indication is valid with both the internal and external PHYs. Read/Write accessible always. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
11	MIISE	Media Independent Interface Selected Enable. Indicates when the MII interface is selected. This will be set when either the Management Port State Machine is selecting the MII or when ASEL (BCR2, bit1) is disabled and PORTSEL (CSR15, bits 8-7) selects the MII. This could control relays to switch in and out appropriate filters or could control an external PHY when sharing an RJ45 connector. Read/Write accessible always. MIISE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

10 DXCVRCTL DXCVR Control. When the AUI interface is the active network port, DXCVRCTL controls the assertion of the $\overline{\text{LED1}}$ output. The polarity of the asserted state is controlled by the LEDPOL bit (BCR4, bit 14). The $\overline{\text{LED1}}$ 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 Am79C971 AUI port. When DXCVRCTL is set to 1, the $\overline{\text{LED1}}$ 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 LEDPOL is cleared to 0). When DXCVRCTL is cleared to 0, the $\overline{\text{LED1}}$ 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.

9 MPSE Magic Packet Status Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when Magic Packet mode is enabled and a Magic Packet frame is detected on the network.

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

8 FDLSE Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LEDOUT signal when the Am79C971 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C971 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal.

When the 10BASE-T port is active, a value of 1 is passed to the LEDOUT signal whenever the Link Test Function (described in the T-MAU section) detects a Link Pass state and the FDEN (BCR9, bit 0) bit is set. When the AUI port is active, a value of 1 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 1).

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

7 PSE Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher.

Read/Write accessible always. PSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

6 LNKSE Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register when the T-MAU is in Link Pass state. When the T-MAU is in Link Fail state, a value of 0 is passed to the LEDOUT bit. This bit does not reflect the link status of the external PHY.

The function of this bit is masked if the 10BASE-T port is operating in full-duplex mode. This allows a Half-Duplex Link Status LED and a Full-Duplex Link Status LED at the same time.

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

5	RCVME	Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when 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 setting the STOP bit.
4	XMTE	Transmit Status Enable. When this bit is set, a value of 1 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 setting the STOP bit.
3	RXPOLE	Receive Polarity Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when normal polarity of the RXD± pair has not been 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 setting the STOP bit.
2	RCVE	Receive Status Enable. When this bit is set, a value of 1 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 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.
1	JABE	Jabber Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the Am79C971 controller is jabbering on the network. Read/Write accessible always. JABE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
0	COLE	Collision Status Enable. When this bit is set, a value of 1 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 setting the STOP bit.

BCR6: LED2 Status

BCR6 controls the function(s) that the $\overline{\text{LED2}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR6 defaults to Receive Polarity Status (RXPOL) with pulse stretcher enabled (PSE = 1) and is fully programmable.

Note: *When LEDPE (BCR2, bit 12) is set to 1, programming of the LED2 Status register is enabled. When LEDPE is cleared to 0, programming of the LED2 register is disabled. Writes to those registers will be ignored.*

Note: *Bits 15-0 in this register are programmable through the EEPROM PREAD operation.*

Bit	Name	Description
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 1 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 the LED 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 STOP.

14 LEDPOL

LED Polarity. When this bit has the value 0, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false (i.e., 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 1, then the LED pin will be driven to a HIGH level 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 (i.e., 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 setting the STOP bit.

13 LEDDIS

LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then 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 setting the STOP bit.

12 100E

100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C971 controller is operating at 100 Mbps mode. The indication is valid with both the internal and external PHYs.

Read/Write accessible always. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

11 MIISE

Media Independent Interface Selected Enable. Indicates when the MII interface is selected. This will be set when either the Management Port State Machine is selecting the MII or when ASEL (BCR2, bit1) is disabled and PORTSEL (CSR15, bits 8-7) selects the MII. This could control relays to switch in and out appropriate filters or could control an external PHY when sharing an RJ45 connector.

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

10 DXCVRCTL

DXCVR Control. When the AUI interface is the active network port, DXCVRCTL controls the assertion of the $\overline{\text{LED2}}$ output. The polarity of the asserted state is controlled by the LEDPOL bit (BCR4, bit 14). The $\overline{\text{LED2}}$ 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 Am79C971 AUI port. When DXCVRCTL is set to 1, the $\overline{\text{LED2}}$ 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 LEDPOL is cleared to 0). When DXCVRCTL is cleared to 0, the $\overline{\text{LED2}}$ 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.

9 MPSE

Magic Packet Status Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when Magic Packet frame mode is enabled and a Magic Packet frame is detected on the network.

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

8 FDLSE

Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LED-OUT signal when the Am79C971 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C971 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal.

When the 10BASE-T port is active, a value of 1 is passed to the LEDOUT signal whenever the Link Test Function (described in the T-MAU section) detects a Link Pass state and the FDEN (BCR9, bit 0) bit is set. When the AUI port is active, a value of 1 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 1).

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

7 PSE

Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher.

Read/Write accessible always. PSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

6 LNKSE

Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register when the T-MAU is in Link Pass state. When the T-MAU is in Link Fail state, a value of 0 is passed to the LEDOUT bit. This bit does not reflect the link status of the external PHY.

The function of this bit is masked if the 10BASE-T port is operating in full-duplex mode. This allows a Half-Duplex Link Status LED and a Full-Duplex Link Status LED at the same time.

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

5 RCVME

Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when 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 setting the STOP bit.

4 XMTE

Transmit Status Enable. When this bit is set, a value of 1 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 setting the STOP bit.

3 RXPOLE

Receive Polarity Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when normal polarity of the RXD± pair has not been reversed.

Receive polarity indication is valid only if the T-MAU is in link pass state.

		Read/Write accessible always. RXPOLE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.
2	RCVE	Receive Status Enable. When this bit is set, a value of 1 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 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.
1	JABE	Jabber Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the Am79C971 controller is jabbering on the network. Read/Write accessible always. JABE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
0	COLE	Collision Status Enable. When this bit is set, a value of 1 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 setting the STOP bit.

BCR7: LED3 Status

BCR7 controls the function(s) that the $\overline{\text{LED3}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR7 defaults to Transmit Status (XMT) with pulse stretcher enabled (PSE = 1) and is fully programmable.

Note: *When LEDPE (BCR2, bit 12) is set to 1, programming of the LED3 Status register is enabled. When LEDPE is cleared to 0, programming of the LED3 register is disabled. Writes to those registers will be ignored.*

Note: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
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 1 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 the LED 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 STOP.
14	LEDPOL	LED Polarity. When this bit has the value 0, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false (i.e., 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 1, then the LED pin will be driven to a HIGH level 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 (i.e., 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 setting the STOP bit.

13 LEDDIS

LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then 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 setting the STOP bit.

12 100E

100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C971 controller is operating at 100 Mbps mode. The indication is valid with both the internal and external PHYs.

Read/Write accessible always. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

11 MIISE

Media Independent Interface Selected Enable. Indicates when the MII interface is selected. This will be set when either the Management Port State Machine is selecting the MII or when ASEL (BCR2, bit1) is disabled and PORTSEL (CSR15, bits 8-7) selects the MII. This could control relays to switch in and out appropriate filters or could control an external PHY when sharing an RJ45 connector.

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

10 DXCVRCTL

DXCVR Control. When the AUI interface is the active network port, DXCVRCTL controls the assertion of the $\overline{\text{LED3}}$ output. The polarity of the asserted state is controlled by the LEDPOL bit (BCR4, bit 14). The $\overline{\text{LED3}}$ 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 Am79C971 AUI port. When DXCVRCTL is set to 1, the $\overline{\text{LED3}}$ 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 LEDPOL is cleared to 0). When DXCVRCTL is cleared to 0, the $\overline{\text{LED3}}$ 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.

9 MPSE

Magic Packet Status Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when magic frame mode is enabled and a magic frame is detected on the network.

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

8 FDLSE

Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LEDOUT signal when the Am79C971 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C971 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal.

When the 10BASE-T port is active, a value of 1 is passed to the LEDOUT signal whenever the Link Test Function (described in the T-MAU section) detects a Link Pass state and the FDEN (BCR9, bit 0) bit is set. When the AUI port is active, a value of 1 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 1).

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

7 PSE — Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher.

Read/Write accessible always. PSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

6 LNKSE — Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register when the T-MAU is in Link Pass state. When the T-MAU is in Link Fail state, a value of 0 is passed to the LEDOUT bit. This bit does not reflect the link status of the external PHY.

The function of this bit is masked if the 10BASE-T port is operating in full-duplex mode. This allows a Half-Duplex Link Status LED and a Full-Duplex Link Status LED at the same time.

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

5 RCVME — Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when 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 setting the STOP bit.

4 XMTE — Transmit Status Enable. When this bit is set, a value of 1 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 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

3 RXPOLE — Receive Polarity Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when normal polarity of the RXD± pair has not been 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 setting the STOP bit.

2 RCVE — Receive Status Enable. When this bit is set, a value of 1 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 setting the STOP bit.

1 JABE — Jabber Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the Am79C971 controller is jabbering on the network.

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

0 COLE — Collision Status Enable. When this bit is set, a value of 1 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 setting the STOP bit.

BCR9: Full-Duplex Control

Note: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
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 1 and full-duplex mode is enabled, the Am79C971 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 0. The Am79C971 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 controls whether or not full-duplex operation on the AUI port is enabled. AUIFD is only meaningful if FDEN (BCR9, bit 0) is set to 1. If the FDEN bit is 0, the AUI port will always operate in half-duplex mode. In addition, if FDEN is set to 1 but the AUIFD bit is reset to 0, the AUI port will always operate in half-duplex mode. If FDEN is set to 1 and AUIFD is set to 1, full-duplex operation on the AUI port is enabled. Read/Write accessible always. AUIFD is reset to 0 by H_RESET, and is unaffected by S_RESET and the STOP bit.
0	FDEN	Full-Duplex Enable. FDEN controls whether full-duplex operation is enabled. When FDEN is cleared, full-duplex operation is not enabled and the Am79C971 controller will always operate in the half-duplex mode. When FDEN is set, the Am79C971 controller will operate in full-duplex mode when the 10BASE-T or MII port is enabled or when the AUI port is enabled and the AUIFD (BCR9, bit 1) bit is set. When DLNKTST (CSR15, bit 12) is set to 1, full-duplex operation will not be enabled on the 10BASE-T port. FDEN will override the Auto-Negotiation portion of the internal 10BASE-T MAU. The internal TMAU will no longer try to automatically negotiate for the link. It assumes that the software is programming the FDEN bit for a reason and defers control. See Table 32. Do not set this bit when Auto-Negotiation is enabled. Read/Write accessible always. FDEN is reset to 0 by H_RESET, and is unaffected by S_RESET and 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 Am79C971 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 STOP.

4-0	RES	Reserved locations. Written as zeros, read as undefined.

Table 32. Network Port Configuration

AUIFD (bit 1)	FDEN (bit 0)	Effect on the AUI Port	Effect on the 10BASE-T Port	Effect on the GPSI Port	Effect On the MII Port (ASEL = 0, PORTSEL = MII)
X	0	Half-Duplex	Half-Duplex	Half-Duplex	Half-Duplex
0	1	Half-Duplex	Full-Duplex	Full-Duplex	Full-Duplex
1	1	Full-Duplex	Full-Duplex	Full-Duplex	Full-Duplex

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 settings of this register will have no effect on any Am79C971 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 STOP.

BCR18: Burst and Bus Control Register

Note: *Note that bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
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 for all EBDATA (BCR30) accesses to SRAM/ Flash/EPROM as well as all Expansion ROM accesses to Flash/ EPROM. ROMTMG, during read operations, defines the time from when the Am79C971 controller drives the lower 8 or 16 bits of the Expansion Bus Address bus to when the Am79C971 controller latches in the data on the 8 or 16 bits of the Expansion Bus Data inputs. ROMTMG, during write operations, defines the time from when the Am79C971 controller drives the lower 8 or 16 bits of the Expansion Bus Data to when the $\overline{EBWE}$ and $\overline{EROMCS}$ deassert. The differences in the sizes of the Expansion Bus Address and Data busses is due to the difference in the access for SRAM versus Flash/EPROM. The register value specifies the time in number of clock cycles +1 according to Table 33.

Table 33. ROMTNG Programming Values

ROMTMG (bits 15-12)	No. of Expansion Bus Cycles
1h<=n <=Fh	n+1

Note*: Programming ROMTNG with a value of 0 is not permitted.*

The access time for the Expansion ROM or the EDBATA (BCR30) device (t_{ACC}) during read operations can be calculated by subtracting the clock to output delay for the EBUA_EBA[7:0] outputs ($t_{v\_A\_D}$) and by subtracting the input to clock setup time for the EBD[7:0] inputs (t_{s_D}) from the time defined by ROMTMG:

t_{ACC} = ROMTMG * CLK period *CLK_FAC - ($t_{v\_A\_D}$) - (t_{s_D})

The access time for the Expansion ROM or for the EDBATA (BCR30) device (t_{ACC}) during write operations can be calculated by subtracting the clock to output delay for the EBUA EBA[7:0] outputs ($t_{v\_A\_D}$) and by adding the input to clock setup time for SRAM/Flash/EPRO inputs (t_{s_D}) from the time defined by ROMTMG:

t_{ACC} = ROMTMG * CLK period * CLK_FAC - ($t_{v\_A\_D}$) - (t_{s_D})

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 the STOP bit is set. ROMTMG is set to the value of 1001b by H_RESET and is not affected by S_RESET or STOP. 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 NOUFLO

No Underflow on Transmit. When the NOUFLO bit is set to 1, the Am79C971 controller will not start transmitting the preamble for a packet until the Transmit Start Point (CSR80, bits 10-11) requirement (except when XMTSP = 3h, Full Packet has no meaning when NOUFLO is set to 1) has been met *and* the complete packet has been DMA'd into the Am79C971 controller. The complete packet may reside in any combination of the Bus Transmit FIFO, the external SRAM, and the MAC Transmit FIFO, as long as enough of the packet is in the MAC Transmit FIFO to meet the Transmit Start Point requirement. When the NOUFLO bit is cleared to 0, the Transmit Start Point is the only restriction on when preamble transmission begins for transmit packets.

Setting the NOUFLO bit guarantees that the Am79C971 controller will never suffer transmit underflows, because the arbiter that controls transfers to and from the external SRAM guarantees a worst case latency on transfers to and from the MAC and Bus Transmit FIFOs such that it will never underflow if the complete packet has been DMA'd into the Am79C971 controller before packet transmission begins.

The NOUFLO bit should not be set when the Am79C971 controller is operating in the NO-SRAM mode with no external SRAM.

Read/Write accessible only when either the STOP or the SPND bit is set. NOUFLO is cleared to 0 after H_RESET or S_RESET and is unaffected by STOP.

10 RES

Reserved location. Written as zeros and read as undefined.

9 MEMCMD

Memory Command used for burst read accesses to the transmit buffer. When MEMCMD is set to 0, all burst read accesses to the transmit buffer are of the PCI command type Memory Read Line (type 14). When MEMCMD is set to 1, all burst read accesses to the transmit buffer 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 STOP.

8 EXTREQ

Extended Request. This bit controls the deassertion of $\overline{REQ}$ for a burst transaction. If EXTREQ is set to 0, $\overline{REQ}$ is deasserted at the beginning of a burst transaction. (The Am79C971 controller never performs more than one burst transaction within a single bus mastership period.) In this mode, the Am79C971 controller relies on the PCI latency timer to get enough bus bandwidth, in case the system arbiter also removes $\overline{GNT}$ at the beginning of the burst transaction. If EXTREQ is set to 1, $\overline{REQ}$ stays asserted until the last but one data phase of the burst transaction is done. This mode is useful for systems that implement an arbitration scheme without preemption and require that $\overline{REQ}$ stays asserted throughout the transaction.

EXTREQ should not be set to 1 when the Am79C971 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 is not affected by S_RESET or STOP.

7 DWIO — Double Word I/O. When set, this bit indicates that the Am79C971 controller is programmed for DWord I/O (DWIO) mode. When cleared, this bit indicates that the Am79C971 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 Am79C971 controllers I/O resources. See the DWIO and WIO sections for more details.

The initial value of the DWIO bit is determined by the programming of the EEPROM.

The value of DWIO can be altered automatically by the Am79C971 controller. Specifically, the Am79C971 controller will set DWIO if it detects a DWord write access to offset 10h from the Am79C971 controller I/O base address (corresponding to the RDP resource).

Once the DWIO bit has been set to a 1, only a H_RESET or an EEPROM read can reset it to a 0. (Note that the EEPROM read operation will only set DWIO to a 0 if the appropriate bit inside of the EEPROM is set to 0.)

Read accessible always. DWIO is read only, write operations have no effect. DWIO is cleared by H_RESET and is not affected S_RESET or by setting the STOP bit.

6 BREADE — Burst Read Enable. When set, this bit enables burst mode during memory read accesses. When cleared, this bit prevents the device from performing bursting during read accesses. The Am79C971 controller can perform burst transfers when reading the initialization block, the descriptor ring entries (when SWSTYLE = 3) and the buffer memory.

BREADE should be set to 1 when the Am79C971 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. BREADE is cleared by H_RESET and is not affected by S_RESET or STOP.

5 BWRITE — Burst Write Enable. When set, this bit enables burst mode during memory write accesses. When cleared, this bit prevents the device from performing bursting during write accesses. The Am79C971 controller can perform burst transfers when writing the descriptor ring entries (when SWSTYLE = 3) and the buffer memory.

BWRITE should be set to 1 when the Am79C971 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 STOP.

4-3 TSTSHDW — Reserved locations. Written an read as zeros.

2-0 LINBC — 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 have no effect on any Am79C971 controller function. LINBC is not affected by S_RESET or STOP.

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. Read accessible only. PVALID is read only; write operations have no effect. A value of 1 in this bit indicates that a PREAD operation has occurred, and that (1) there is an EEPROM connected to the Am79C971 controller interface pins and (2) the contents read from the EEPROM have passed the checksum verification operation. A value of 0 in this bit indicates a failure in reading the EEPROM. The checksum for the entire 64 bytes of EEPROM is incorrect or no EEPROM is connected to the interface pins. PVALID is set to 0 during H_RESET and is unaffected by S_RESET or the STOP bit. However, following the H_RESET operation, an automatic read of the EEPROM will be performed. Just as is true for the normal PREAD command, at the end of this automatic read operation, the PVALID bit may be set to 1. Therefore, H_RESET will set the PVALID bit to 0 at first, but the automatic EEPROM read operation may later set PVALID to a 1. If PVALID becomes 0 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 content of the Address PROM locations, however, will not be cleared. If no EEPROM is present at the EESK, EEDI, and EEDO pins, 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.
14	PREAD	EEPROM Read command bit. When this bit is set to a 1 by the host, the PVALID bit (BCR19, bit 15) will immediately be reset to a 0, and then the Am79C971 controller will perform a read operation of 64 bytes from the EEPROM through the interface. The EEPROM data that is fetched during the read will be stored in the appropriate internal registers on board the Am79C971 controller. Upon completion of the EEPROM read operation, the Am79C971 controller will assert the PVALID bit. EEPROM contents will be indirectly accessible to the host through read accesses to the Address PROM (offsets 0h through Oh) and through read accesses to other EEPROM programmable registers. Note that read accesses from these locations will not actually access the EEPROM itself, but instead will access the Am79C971 controllers internal copy of the EEPROM contents. Write accesses to these locations may change the Am79C971 controller register contents, but the EEPROM locations will not be affected. EEPROM locations may be accessed directly through BCR19. At the end of the read operation, the PREAD bit will automatically be reset to a 0 by the Am79C971 controller and PVALID will bet set, provided that an EEPROM existed on the interface pins and that the checksum for the entire 64 bytes of EEPROM was correct. Note that when PREAD is set to a 1, then the Am79C971 controller will no longer respond to any accesses directed toward it, until the PREAD operation has completed successfully. The Am79C971 controller will terminate these accesses with the as-

sertion of DEVSEL and STOP while 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 Am79C971 controller but no EEPROM is attached to the interface pins, the PREAD bit will be cleared to a 0, and the PVALID bit will remain reset with a value of 0. This applies to the automatic read of the EEPROM after H_RESET as well as to host initiated PREAD commands. EEPROM programmable locations on board the Am79C971 controller will be set to their default values by such an aborted PREAD operation. For example, if the aborted PREAD operation immediately followed the H_RESET operation, then the final state of the EEPROM programmable locations will be equal to the H_RESET programming for those locations.

If a PREAD command is given to the Am79C971 controller and the auto-detection pin (EESK/LED1/SFBD) indicates that no EEPROM is present, then the EEPROM read operation will still be attempted.

Note that at the end of the H_RESET operation, a read of the EEPROM will be performed automatically. This H_RESET-generated EEPROM read function will not proceed if the auto-detection pin (EESK/LED1/SFBD) indicates that no EEPROM is present.

Read accessible always; write accessible only when either the STOP or the SPND bit is set. PREAD is set to 0 during H_RESET and is unaffected by S_RESET or the STOP bit.

13 EEDET

EEPROM Detect. This bit indicates the sampled value of the EESK/LED1/SFBD pin at the end of H_RESET. This value indicates whether or not an EEPROM is present at the EEPROM interface. If this bit is a 1, it indicates that an EEPROM is present. If this bit is a 0, it indicates that an EEPROM is not present.

Read accessible only. EEDET is read only; write operations have no effect. The value of this bit is determined at the end of the H_RESET operation. It is unaffected by S_RESET or the STOP bit.

Table 34 indicates the possible combinations of EEDET and the existence of an EEPROM and the resulting operations that are possible on the EEPROM interface.

12-5 RES

Reserved locations. Written as zeros; read as undefined.

4 EEN

EEPROM Port Enable. When this bit is set to a 1, it causes the values of ECS, ESK, and EDI to be driven onto the EECS, EESK, and EEDI pins, respectively. If EEN = 0 and no EEPROM read function is currently active, then EECS will be driven LOW. When EEN = 0 and no EEPROM read function is currently active, EESK and EEDI pins will be driven by the LED registers BCR5 and BCR4, respectively. See Table 35.

Read accessible always, write accessible only when either the STOP or the SPND bit is set. EEN is set to 0 by H_RESET and is unaffected by the S_RESET or 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 interface when the EEN bit is set to 1 and the PREAD bit is set to 0. If EEN = 1

and PREAD = 0 and ECS is set to a 1, then the EECS pin will be forced to a HIGH level at the rising edge of the next clock following bit programming.

If EEN = 1 and PREAD = 0 and ECS is set to a 0, 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 set to 0 and the EEN bit is set to 1.

Read accessible always, write accessible only when either the STOP or the SPND bit is set. ECS is set to 0 by H_RESET and is not affected by S_RESET or STOP.

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 onto the EESK pin at the rising edge of the next clock following bit programming, except when the PREAD bit is set to 1 or the EEN bit is set to 0. If both the ESK bit and the EDI/EDO bit values are changed during one BCR19 write operation, while EEN = 1, 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 set to 0 and the EEN bit is set to 1.

Read accessible always, write accessible only when either the STOP or the SPND bit is set. ESK is reset to 1 by H_RESET and is not affected by S_RESET or STOP.

0 EDI/EDO — EEPROM Data In/ EEPROM Data Out. Data that is written to this bit will appear on the EEDI output of the interface, except when the PREAD bit is set to 1 or the EEN bit is set to 0. Data that is read from this bit reflects the value of the EEDO input of the interface.

EDI/EDO has no effect on the EEDI pin unless the PREAD bit is set to 0 and the EEN bit is set to 1.

Read accessible always; write accessible only when either the STOP or the SPND bit is set. EDI/EDO is reset to 0 by H_RESET and is not affected by S_RESET or STOP.

Table 34. EEDET Setting

EEDET Value (BCR19[13])	EEPROM Connected?	Result if PREAD is Set to 1	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 reset to 0.	First two EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to 0.
0	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to 1.	First two EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to 0.
1	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to 0.	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to 0.
1	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to 1.	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to 1.

Table 35. Interface Pin Assignment

$\overline{\text{RST}}$ Pin	PREAD or Auto Read in Progress	EEN	EECS	EESK	EEDI
Low	X	X	0	Tri-State	Tri-State
High	1	X	Active	Active	Active
High	0	1	From ECS Bit of BCR19	From ESK Bit of BCR19	From EEDI Bit of BCR19
High	0	0	0	LED1	LED0

BCR20: Software Style

This register is an alias of the location CSR58. Accesses to and from this register are equivalent to accesses to CSR58.

Bit	Name	Description
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 1, the BPE bits (RMD1 and TMD1, bit 23) start having a meaning. BPE will be set in the descriptor associated with the buffer that was accessed when a data parity error occurred. 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 2 or 3 to program the Am79C971 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 Am79C971 controller is the target of the transfer. Read anytime; 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 STOP.
9	RES	Reserved locations. Written as zeros; read as undefined.
8	SSIZE32	Software Size 32 bits. When set, this bit indicates that the Am79C971 controller utilizes 32-bit software structures for the initialization block and the transmit and receive descriptor entries. When cleared, this bit indicates that the Am79C971 controller utilizes 16-bit software structures for the initialization block and the transmit and receive descriptor entries. In this mode, the Am79C971 controller is backwards compatible with the Am7990 LANCE and Am79C960 PCnet-ISA controllers. The value of SSIZE32 is determined by the Am79C971 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 0) and is not affected by S_RESET or STOP. If SSIZE32 is reset, 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 Am79C971 controller. This action is required, since the 16-bit software structures specified by the SSIZE32 = 0 setting will yield only 24 bits of address for Am79C971 controller bus master accesses. If SSIZE32 is set, then the software structures that are common to the Am79C971 controller and the host system will supply a full 32 bits for each address pointer that is needed by the Am79C971 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 defined width for I/O resources. I/O resource 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 Am79C971 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 Am79C971 controller CSR bits and all descriptor, buffer, and initialization block entries not cited in the Table 36 are unaffected by the Software Style selection and are, therefore, always fully functional as specified in the CSR and BCR sections.

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 STOP.

BCR21: Interrupt Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	INTCON	Reserved locations. Writes to this register will have no effect on the operation of the Am79C971 controller.

Table 36. Software Styles

SWSTYLE [7:0]	Style Name	SSIZE32	Initialization Block Entries	Descriptor Ring Entries
00h	LANCE/ PCnet-ISA controller	0	16-bit software structures, non-burst or burst access	16-bit software structures, non-burst access only
01h	RES	1	RES	RES
02h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only
03h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only
All Other	Reserved	Undefined	Undefined	Undefined

BCR22: PCI Latency Register

Note: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	MAX_LAT	Maximum Latency. Specifies the maximum arbitration latency the Am79C971 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 will use the value in the register to determine the setting of the Am79C971 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 results in a default maximum latency of 63.75 microseconds. It is recom-

mended to program the value of 18H via EEPROM. MAX_LAT is not affected by S_RESET or STOP.

7-0 MIN_GNT Minimum Grant. Specifies the minimum length of a burst period the Am79C971 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 µs. MIN_GNT is aliased to the PCI configuration space register MIN_GNT (offset 3Eh). The host will use the value in the register to determine the setting of the Am79C971 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 results in a default minimum grant of 1.5 µs, which is the time it takes the Am79C971 controller to read/write half of the FIFO. (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. It is recommended to program the value of 18H via EEPROM. MIN_GNT is not affected by S_RESET or STOP.

BCR23: PCI Subsystem Vendor ID Register

Note: Bits 15-0 in this register are programmable through the EEPROM.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SVID	Subsystem Vendor ID. SVID is used together with SID (BCR24, bits 15-0) to uniquely identify the add-in board or subsystem the Am79C971 controller is used in. Subsystem Vendor IDs can be obtained from the PCI SIG. A value of 0 (the default) indicates that the Am79C971 controller does not support subsystem identification. SVID is aliased to the PCI configuration space register Subsystem Vendor ID (offset 2Ch). Read accessible always. SVID is read only. Write operations are ignored. SVID is cleared to 0 by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR24: PCI Subsystem ID Register

Note: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SID	Subsystem ID. SID is used together with SVID (BCR23, bits 15-0) to uniquely identify the add-in board or subsystem the Am79C971 controller is used in. The value of SID is up to the system vendor. A value of 0 (the default) indicates that the Am79C971 controller does not support subsystem identification. SID is aliased to the PCI configuration space register Subsystem ID (offset 2Eh). Read accessible always. SID is read only. Write operations are ignored. SID is cleared to 0 by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR25: SRAM Size Register

Bit	Name	Description

Note: Bits 7-0 in this register are programmable through the EEPROM.

31-8	RES	Reserved locations. Written as zeros and read as undefined.

7-0	SRAM_SIZE	SRAM Size. Specifies the upper 8 bits of the 16-bit total size of the SRAM buffer. Each bit in SRAM_SIZE accounts for a 512-byte page. The starting address for the lower 8 bits is assumed to be 00h and the ending address for the lower is assumed to be FFh. Therefore, the maximum address range is the starting address of 0000h to ending address of ((SRAM_SIZE +1) * 256 words) or FFFFh. An SRAM_SIZE value of all zeros specifies that no SRAM is present and the internal FIFOs will be joined into a contiguous FIFO similar to the PCnet-PCI II controller. ***Note**: The minimum allowed number of pages is eight for normal network operation. The Am79C971 controller will not operate correctly with less than the eight pages of memory. When the minimum number of pages is used, these pages must be allocated four each for transmit and receive. Also note that a "No SRAM configuration" is only valid for 10Mb mode. In 100Mb mode, SRAM is mandatory and must always be used.* ***CAUTION: Programming SRAM_BND and SRAM_SIZE to the same value will cause data corruption except in the case where SRAM_SIZE is 0.*** Read accessible always; write accessible only when the STOP bit is set. SRAM_SIZE is set to 000000b during H_RESET and is unaffected by S_RESET or STOP.

BCR26: SRAM Boundary Register

Bit	Name	Description
Note: Bits 7-0 in this register are programmable through the EEPROM.		
31-8	RES	Reserved locations. Written as zeros and read as undefined.
7-0	SRAM_BND	SRAM Boundary. Specifies the upper 8 bits of the 16-bit address boundary where the receive buffer begins in the SRAM. The transmit buffer in the SRAM begins at address 0 and ends at the address located just before the address specified by SRAM_BND. Therefore, the receive buffer always begins on a 512 byte boundary. The lower bits are assumed to be zeros. SRAM_BND has no effect in the Low Latency Receive mode. ***Note**: The minimum allowed number of pages is four. The Am79C971 controller will not operate correctly with less than four pages of memory per queue. See Table 37 for SRAM_BND programming details.*

Table 37. SRAM_BND Programming

SRAM Addresses	SRAM_BND [7:0]	Lower Address [7:0]
Minimum SRAM_BND address	04h	00-FFh
Maximum SRAM_BND address	FCh	00-FFh

CAUTION: Programming SRAM_BND and SRAM_SIZE to the same value will cause data corruption except in the case where SRAM SIZE is 0.

Read accessible always; write accessible only when the STOP bit is set. SRAM_BND is set to 00000000b during H_RESET and is unaffected by S_RESET or STOP.

BCR27: SRAM Interface Control Register

***Note:** Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	PTR TST	Reserved. Reserved for manufacturing tests. Written as zero and read as undefined.

***Note**: Use of this bit will cause data corruption and erroneous operation.*

Read/Write accessible always. PTR_TST is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit.

14 LOLATRX Low Latency Receive. When the LOLATRX bit is set to 1, the Am79C971 controller will switch to an architecture applicable to cut-through switches. The Am79C971 controller will assert a receive frame DMA after only 16 bytes of the current receive frame has been received regardless of where the RCVFW (CSR80, bits 13-12) are set. The watermark is a fixed value and cannot be changed. The receive FIFOs will be in NO_SRAM mode while all transmit traffic is buffered through the external SRAM. This bit is only valid and the low latency receive only enabled when the SRAM_SIZE (BCR25, bits 7-0) bits are non-zero. SRAM_BND (BCR26, bits 7-0) has no meaning when the Am79C971 controller is in the Low Latency mode. See the section on *SRAM Configuration* for more details.

When the LOLATRX bit is set to 0, the Am79C971 controller will return to a normal receive configuration. The runt packet accept bit (RPA, CSR124, bit 3) must be set when LOLATRX is set.

CAUTION: To provide data integrity when switching into and out of the low latency mode, DO NOT SET the FASTSPNDE (CSR7, bit 15) bit when setting the SPND bit. Receive frames WILL be overwritten and the Am79C971 controller may give erratic behavior when it is enable again. The minimum allowed number of pages is four. The Am79C971 controller will not operate correctly in the LOLATRX mode with less than four pages of memory.

Read/Write accessible only when the STOP bit is set. LOLATRX is cleared to 0 after H_RESET or S_RESET and is unaffected by STOP.

13-6 RES Reserved locations. Written as zeros and read as undefined.

5-3 EBCS Expansion Bus Clock Source. These bits are used to select the source of the fundamental clock to drive the SRAM and Expansion ROM access cycles. Table 38 shows the selected clock source for the various values of EBCS. Note that the actual frequency that the Expansion Bus access cycles run at is a function of both the EBCS and CLK_FAC (BCR27, bits 2-0) bit field settings. When EBCS is set to either the PCI clock or the XTAL clock, no external clock source is required as the clocks are routed internally and the EBCLK pin should be pulled to VDD through a resistor.

Table 38. EBCS Values

EBCS	Expansion Bus Clock Source
000	CLK pin (PCI Clock)
001	XTAL1 and XTAL2 pins (20-MHz clock)
010	EBCLK pin
011	Reserved
1XX	Reserved

Read accessible always; write accessible only when the STOP bit is set. EBCS is set to 000b (PCI clock selected) during H_RESET and is unaffected by S_RESET or the STOP bit.

***Note**: The clock frequency driving the Expansion Bus access cycles that results from the settings of the EBCS and CLK FAC bits must not exceed 33 MHz at any time. When EBCS is set to either the PCI clock or the XTAL clock, no external clock source is required because the clocks are routed internally and the EBCLK pin should be pulled to VDD through a resistor.*

CAUTION: Care should be exercised when choosing the PCI clock pin because of the nature of the PCI clock signal. The PCI specification states that the PCI clock can be stopped. If that can occur while it is being used for the Expansion Bus clock data, corruption will result.

CAUtiON: The external clock source used to drive the EBCLK pin must be a continuous clock source at all times.

2-0 CLK_FAC Clock Factor. These bits are used to select whether the clock selected by EBCS is used directly or if it is divided down to give a slower clock for running the Expansion Bus access cycles. The possible factors are given in Table 39.

Table 39. CLK_FAC Values

CLK_FAC	Clock Factor
000	1
001	1/2 (divide by 2)
010	Reserved
011	1/4 (divide by 4)
1XX	Reserved

Read accessible always; write accessible only when the STOP bit is set. CLK_FAC is set to 000b during H_RESET and is unaffected by S_RESET or STOP.

BCR28: Expansion Bus Port Address Lower (Used for Flash/EPROM and SRAM Accesses)

Bit	Name	Description
31-8	RES	Reserved locations. Written as zeros and read as undefined.
15-0	EPADDRL	Expansion Port Address Lower. This address is used to provide addresses for the Flash and SRAM port accesses. SRAM accesses are started when a read or write is performed on BCR30 and the FLASH (BCR 29, bit 15) is set to 0. During SRAM accesses only bits in the EPADDRL are valid. Since all SRAM accesses are word oriented only, EPADDRL[0] is the least significant word address bit. On any byte write accesses to the SRAM, the user will have to follow the read-modify-write scheme. On any byte read accesses to the SRAM, the user will have to chose which byte is needed from the complete word returned in BCR30. Flash accesses are started when a read or write is performed on BCR30 and the FLASH (BCR 29, bit 15) is set to 1. During Flash accesses all bits in EPADDR are valid. Read accessible always; write accessible only when the STOP is set or when SRAM SIZE (BCR25, bits 7-0) is 0. EPADDRL is undefined after H_RESET and is unaffected by S_RESET or STOP.

BCR29: Expansion Port Address Upper (Used for Flash/EPROM Accesses)

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	FLASH	Flash Access. When the FLASH bit is set to 1 the Expansion Bus access will be a Flash cycle. When FLASH is set to 0 the Expansion Bus access will be a SRAM cycle. For a complete description see the section on *Expansion Bus Interface*. This bit is only applicable to reads or writes to EBDATA (BCR30). It does not affect Expansion ROM accesses from the PCI system bus. Read accessible always; write accessible only when the STOP bit is set. FLASH is 0 after H_RESET and is unaffected by S_RESET or the STOP bit.
14	LAAINC	Lower Address Auto Increment. When the LAAINC bit is set to 1, the Expansion Port Lower Address will automatically increment by one after a read or write ac-

		cess to EBDATA (BCR30). When EBADDRL reaches FFFFh and LAAINC is set to 1, the Expansion Port Lower Address (EPADDRL) will roll over to 0000h. When the LAAINC bit is set to 0, the Expansion Port Lower Address will not be affected in any way after an access to EBDATA (BCR30) and must be programmed. Read accessible always; write accessible only when the STOP bit is set. LAINC is 0 after H_RESET and is unaffected by S_RESET or the STOP bit.
13-4	RES	Reserved locations. Written as zeros and read as undefined.
3-0	EPADDRU	Expansion Port Address Upper. This upper portion of the Expansion Bus address is used to provide addresses for Flash/EPROM port accesses. Read accessible always; write accessible only when the STOP bit is set or when SRAM SIZE (BCR25, bits 7-0) is 0. EPADDRU is undefined after H_RESET and is unaffected by S_RESET or the STOP bit.

BCR30: Expansion Bus Data Port Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	EBDATA	Expansion Bus Data Port. EBDATA is the data port for operations on the Expansion Port accesses involving SRAM and Flash accesses. The type of access is set by the FLASH bit (BCR 29, bit 15). When the FLASH bit is set to 1, the Expansion Bus access will follow the Flash access timing. When the FLASH bit is set to 0, the Expansion Bus access will follow the SRAM access timing. ***Note:*** *It is important to set the FLASH bit and load Expansion Port Address EPADDR (BCR28, BCR29) with the required address before attempting read or write to the Expansion Bus data port. The Flash and SRAM accesses use different address phases. Incorrect configuration will result in a possible corruption of data.* Flash read cycles are performed when BCR30 is read and the FLASH bit (BCR29, bit 15) is set to 1. Upon completion of the read cycle, the 8-bit result for Flash access is stored in EBDATA[7:0], EBDATA[15:8] is undefined. Flash write cycles are performed when BCR30 is written and the FLASH bit (BCR29, bit 15) is set to 1. EBDATA[7:0] only is valid for write cycles. SRAM read cycles are performed when BCR30 is read and the FLASH bit (BCR29, bit 15) is set to 0. Upon completion of the read cycle, the 16-bit result for SRAM access is stored in EBDATA. Write cycles to the SRAM are invoked when BCR30 is written and the FLASH bit (BCR29, bit 15) is set to 0. Byte writes to the SRAM must use a read-modify-write scheme since the word is always valid for SRAM write or read accesses. Read and write accessible only when the STOP is set or when SRAM SIZE (BCR25, bits 7-0) is 0. EBDATA is undefined after H_RESET, and is unaffected by S_RESET and the STOP bit.

BCR31: Software Timer Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	STVAL	Software Timer Value. STVAL controls the maximum time for the Software Timer to count before generating the STINT (CSR7, bit 11) interrupt. The Software Timer is a free-running timer that is started upon the first write to STVAL. After the first write, the Software Timer will continually count and set the STINT interrupt at the STVAL period.

The STVAL value is interpreted as an unsigned number with a resolution of 12.8 µs. For instance, a value of 122 ms would be programmed with a value of 9531 (253Bh). A value of 0 is undefined and will result in erratic behavior.

Read and write accessible always. STVAL is set to FFFFh after H_RESET and is unaffected by S_RESET and the STOP bit.

BCR32: MII Control and Status Register

Note: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	ANTST	Reserved. Reserved for manufacturing tests. Written as 0 and read as undefined. ***Note****: Use of this bit will cause data corruption and erroneous operation.* Read/Write accessible always. ANTST is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit.
14	MIIPD	MII PHY Detect. MIIPD reflects the quiescent state of the MDIO pin. MIIPD is continuously updated whenever there is no management operation in progress on the MII interface. When a management operation begins on the interface, the state of MIIPD is preserved until the operation ends, when the quiescent state is again monitored and continuously updates the MIIPD bit. When the MDIO pin is at a quiescent LOW state, MIIPD is cleared to 0. When the MDIO pin is at a quiescent HIGH state, MIIPD is set to 1. MIIPD is used by the automatic port selection logic to select the MII port. When the Auto Select bit (ASEL, BCR2, bit 1) is a 1 and the MIIPD bit is a 1, the MII port is selected. Any transition on the MIIPD bit will set the MIIPDINT bit in CSR7, bit 1. Read accessible always. MIIPD is read only. Write operations are ignored.
13-12	FMDC	Fast Management Data Clock. When FMDC is set to 2h the MII Management Data Clock will run at 10 MHz. The Management Data Clock will no longer be IEEE 802.3u-compliant and setting this bit should be used with care. The accompanying external PHY must also be able to accept management frames at the new clock rate. When FMDC is set to 1h, the MII Management Data Clock will run at 5 MHz. The Management Data Clock will no longer be IEEE 802.3u-compliant and setting this bit should be used with care. The accompanying external PHY must also be able to accept management frames at the new clock rate. When FMDC is set to 0h, the MII Management Data Clock will run at 2.5 MHz and will be fully compliant to IEEE 802.3u standards.

Table 40. FMDC Values

FMDC	Fast Management Data Clock
00	2.5 MHz
01	5 MHz
10	10 MHz
11	Reserved

Read/Write accessible always. FMDC is set to 0 during H_RESET, and is unaffected by S_RESET and the STOP bit

11	APEP	MII Auto-Poll External PHY. APEP when set to 1 the Am79C971 controller will poll the MII status register in the external PHY. This feature allows the software driver or upper layers to see any changes in the status of the external PHY. An interrupt when enabled is generated when the contents of the new status is different from the previous status. Auto-Poll will not function when the internal PHY is selected.

Read/Write accessible always. APEP is set to 0 during H_RESET and is unaffected by S_RESET and the STOP bit.

10-8 APDW — MII Auto-Poll Dwell Time. APDW determines the dwell time between MII Management Frames accesses when Auto-Poll is turned on. See Table 41.

Table 41. APDW Values

APDW	Auto-Poll™ Dwell Time
000	Continuous (26µs @ 2.5 MHz)
001	Every 128 MDC cycles (103µs @ 2.5 MHz)
010	Every 256 MDC cycles (206µs @ 2.5 MHz)
011	Every 512 MDC cycles (410 µs @ 2.5 MHz)
100	Every 1024 MDC cycles (819 µs @ 2.5 MHz)
101	Every 2048 MDC cycles (1640 µs @ 2.5 MHz)
110-111	Reserved

Read/Write accessible always. APDW is set to 100h after H_RESET and is unaffected by S_RESET and the STOP bit.

7 DANAS — Disable Auto-Negotiation Auto Setup. When DANAS is set, the Am79C971 controller after a H_RESET or S_RESET will remain dormant and not automatically startup the Auto-Negotiation section or the enhanced automatic port selection section. Instead, the Am79C971 controller will wait for the software driver to setup the Auto-Negotiation portions of the device. The automatic port selection for Am79C971 controller will resemble the Pcnet-PCI II controller. The MII programming in BCR33 and BCR34 is still valid. The Am79C971 controller will not generate any management frames unless Auto-Poll is enabled.

Read/write accessible always. DANAS is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit.

6 XPHYRST — External PHY Reset. When XPHYRST is set, the Am79C971 controller after an H_RESET or S_RESET will issue an MII management frames that will reset the external PHY. This bit is needed when there is no way to guarantee the state of the external PHY. This bit must be reprogrammed after every H_RESET.

Read/Write accessible always. XPHYRST is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit. XPHYRST is only valid when the internal Network Port Manager is scanning for a network port.

5 XPHYANE — External PHY Auto-Negotiation Enable. This bit will force the external PHY into enabling Auto-Negotiation. When set to 0 the Am79C971 controller will send a MII management frame disabling Auto-Negotiation.

Read/Write accessible always. XPHYANE is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit. XPHYANE is only valid when the internal Network Port Manager is scanning for a network port.

4 XPHYFD — External PHY Full Duplex. When set, this bit will force the external PHY into full duplex when Auto-Negotiation is not enabled.

Read/Write accessible always. XPHYFD is set to 0 by H_RESET, and is unaffected by S_RESET and the STOP bit. XPHYFD is only valid when the internal Network Port Manager is scanning for a network port.

3 XPHYSP — External PHY Speed. When set, this bit will force the external PHY into 100 Mbps mode when Auto-Negotiation is not enabled.

Read/Write accessible always. XPHYSP is set to 0 by H_RESET, and is unaffected by S_RESET and the STOP bit. XPHYSP is only valid when the internal Network Port Manager is scanning for a network port.

2 MIIµL — Media Independent Interface for Micro Linear 6692. When set, this

bit will allow the Am79C971 controller to work seamlessly with the Micro Linear 6692 PHY. See the section on *Working with Micro Linear 6692* for details.

Read/Write accessible always. MIIµL is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit. MIIµL is only valid when the internal Network Port Manager is scanning for a network port.

1 MIIILP Media Independent Interface Internal Loopback. When set, this bit will cause the internal portion of the MII data port to loopback on itself. The interface is mapped in the following way. The TXD[3:0] nibble data path is looped back onto the RXD[3:0] nibble data path. TX_CLK is looped back as RX_CLK. TX_EN is looped back as RX_DV. CRS is correctly OR'd with TX_EN and RX_DV and always encompasses the transmit frame. TX_ER is looped back as RX_ER. However, TX_ER will not get asserted by the Am79C971 controller to signal an error. The TX_ER function is reserved for future use.

Read/Write accessible always. MIIILP is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit.

0 FCON Fast Configuration Mode. When set this bit will force the internal Management Port State Machine into a Fast Configuration Mode. During this mode, the Management Port State Machine will not attempt to start Auto-Negotiation on the internal as well as the external PHY. Instead, it will rely on link beats for link pass state. This will accelerate the automatic port selection on the Am79C971 controller.

Read/Write accessible always. FCON is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit. FCON is only valid when the internal Network Port Manager is scanning for a network port. See Table 40.

BCR33: MII Address Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-5	PHYAD	MII Management Frame PHY Address. PHYAD contains the 5-bit PHY Address field that is used in the management frame that gets clocked out via the MII management port pins (MDC and MDIO) whenever a read or write transaction occurs to BCR34. The internal PHY device is always addressed as 11111b. The MII management frame will not appear on the MII when reading or writing to the internal PHY. This is done for MII compatibility sake. The Network Port Manager copies the PHYAD after the Am79C971 controller reads the EEPROM and uses it to communicate with the external PHY. The PHY address must be programmed into the EEPROM prior to starting the Am79C971 controller. Read/Write accessible always. PHYAD is undefined after H_RESET and is unaffected by S_RESET and the STOP bit.
4-0	REGAD	MII Management Frame Register Address. REGAD contains the 5-bit Register Address field that is used in the management frame that gets clocked out via the MII management port pins (MDC and MDIO) whenever a read or write transaction occurs to BCR34. Read/Write accessible always. REGAD is undefined after H_RESET and is unaffected by S_RESET and the STOP bit.

BCR34: MII Management Data Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MIIMD	MII Management Data. MIIMD is the data port for operations on the MII management interface (MDIO and MDC). The Am79C971 device builds management frames using the PHYAD and REGAD values from BCR33. The operation code used in each frame is based upon whether a read or write operation has been performed to BCR34. Read cycles on the MII management interface are invoked when BCR34 is read. Upon completion of the read cycle, the 16-bit result of the read operation is stored in MIIMD. Write cycles on the MII management interface are invoked when BCR34 is written. The value written to MIIMD is the value used in the data field of the management write frame. When the PHYAD (BCR33, bits 9-5) is 11111b the data written and read from the MIIMD will be from the internal PHY only. No MII management frame will be sent across the MII when the PHYAD is 11111b. Read/Write accessible always. MIIMD is undefined after H_RESET and is unaffected by S_RESET and the STOP bit.

BCR35: PCI Vendor ID Register

Note: *Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	VID	Vendor ID. The PCI Vendor ID register is a 16-bit register that identifies the manufacturer of the Am79C971 controller. AMD's Vendor ID is 1022h. Note that this Vendor ID is not the same as the Manufacturer ID in CSR88 and CSR89. The Vendor ID is assigned by the PCI Special Interest Group. The Vendor ID is not normally programmable, but the Am79C971 controller allows this due to legacy operating systems that do not look at the PCI Subsystem Vendor ID and the Vendor ID to uniquely identify the add-in board or subsystem that the Am79C971 controller is used in. ***Note: If the operating system or the network operating system supports PCI Subsystem Vendor ID and Subsystem ID, use those to identify the add-in board or subsystem and program the VID with the default value of 1022h***. VID is aliased to the PCI configuration space register Vendor ID (offset 00h). Read accessible always. VID is read only. Write operations are ignored. VID is set to 1022h by H_RESET and is not affected by S_RESET or by setting the STOP bit.

Initialization Block

When SSIZE32 (BCR20, bit 8) is set to 0, 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 0. When SSIZE32 is set to 0, the initialization block looks like Table 42.

Note: *The Am79C971 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 1, 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 0. When SSIZE32 is set to 1, the initialization block looks like Table 43

.

Table 42. 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 43. 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	LADRF 31-00							
IADR+10h	LADRF 63-32							
IADR+14h	RDRA 31-00							
IADR+18h	TDRA 31-00							

RLEN and TLEN

When SSIZE32 (BCR20, bit 8) is set to 0, the software structures are defined to be 16 bits wide, and the RLEN and TLEN fields in the initialization block are each three 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 shown in Table 44. If a value other than those listed in Table 44 is desired, CSR76 and CSR78 can be written after initialization is complete.

When SSIZE32 (BCR20, bit 8) is set to 1, 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 shown in Table 45.

If a value other than those listed in Table 45 is desired, CSR76 and CSR78 can be written after initialization is complete.

RDRA and TDRA

RDRA and TDRA 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 1 (BCR20, bit 8). Each DRE must be located at an 8-byte address boundary when SSIZE32 is set to 0 (BCR20, bit 8).

Table 44. R/TLEN Decoding (SSIZE32 = 0)

R/TLEN	Number of DREs
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Table 45. R/TLEN Decoding (SSIZE32 = 1)

R/TLEN	Number 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

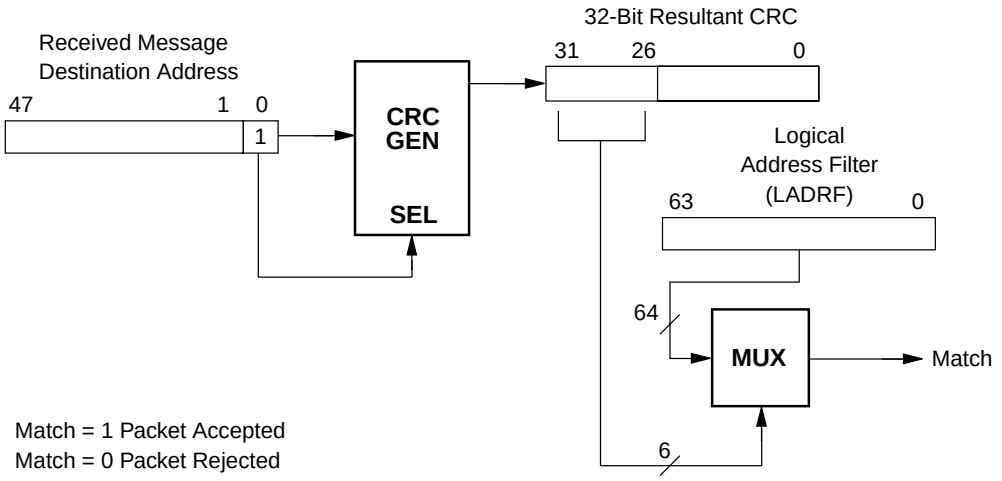
LADRF

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 1, it indicates a logical address. If the first bit is a 0, it is a physical address and is compared against the physical address that was loaded through the initialization block.

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. See Figure 57.



20550D-60

Figure 57. Address Match Logic

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 0 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 Am79C971 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 0, then the software structures are defined to be 16 bits wide, and receive descriptors look like Table 46 (CRDA = Current Receive Descriptor Address).

When SWSTYLE (BCR 20, bits 7-0) is set to 2, then the software structures are defined to be 32 bits wide, and receive descriptors look like Table 47 (CRDA = Current Receive Descriptor Address).

When SWSTYLE (BCR 20, bits 7-0) is set to 3, then the software structures are defined to be 32 bits wide, and receive descriptors look like Table 48 (CRDA = Current Receive Descriptor Address).

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 Am79C971 controller (OWN = 1). The Am79C971 controller clears the OWN bit after filling the buffer that the descriptor points to. The host sets the OWN bit after emptying the buffer

.

Table 46. 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 47. Receive Descriptor (SWSTYLE = 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	RES	RFRTAG[14:0]												0000	MCNT
CRDA+0Ch	USER SPACE														

Table 48. 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	RES	RFRTAG[14:0]												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	USER SPACE														

Once the Am79C971 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 Am79C971 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 Am79C971 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 set by the Am79C971 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 Am79C971 controller and cleared by the host. CRC will also be set when Am79C971 receives an RX_ER indication from the external PHY through the MII.

26 BUFF — Buffer error is set any time the Am79C971 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 0.
2. FIFO overflow occurred before the Am79C971 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 Am79C971 controller and cleared by the host.

25 STP — Start of Packet indicates that this is the first buffer used by the Am79C971 controller for this frame. If STP and ENP are both set to 1, 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 0, STP is set by the Am79C971 controller and cleared by the host. When LAPPEN is set to 1, STP must be set by the host.

24 ENP — End of Packet indicates that this is the last buffer used by the Am79C971 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 Am79C971 controller and cleared by the host.

23 BPE — Bus Parity Error is set by the Am79C971 controller when a parity error occurred on the bus interface during data transfers to a receive buffer. BPE is valid only when ENP, OFLO, or BUFF are set. The Am79C971 controller will only set BPE when the advanced parity error handling is enabled by setting APERREN (BCR20, bit 10) to 1. BPE is set by the Am79C971 controller and cleared by the host.

This bit does not exist when the Am79C971 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).

22 PAM — Physical Address Match is set by the Am79C971 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 Am79C971 controller and cleared by the host.

This bit does not exist when the Am79C971 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).

21 LAFM — Logical Address Filter Match is set by the Am79C971 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 Am79C971 controller and cleared by the host.

Note that if DRCVBC (CSR15, bit 14) is cleared to 0, 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 1 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 Am79C971 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).

20 BAM — Broadcast Address Match is set by the Am79C971 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 Am79C971 controller and cleared by the host.

This bit does not exist when the Am79C971 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).

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 Am79C971 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 Am79C971 controller.

RMD2

Bit	Name	Description
	ZERO	This field is reserved. The Am79C971 controller will write a zero to this location.
30-16	RFRTAG	Receive Frame Tag. Indicates the Receive Frame Tag applied from the EADI interface. This field is user defined and has a default value of all zeros. When RXFRTG (CSR7, bit 14) is set to 0, RFRTAG will be read as all zeros. See the section on *Receive Frame Tagging* for details.
15-12	ZEROS	This field is reserved. Am79C971 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 Am79C971 controller and cleared by the host. ***Note***: This is a 13-bit internal counter.

RMD3

Bit	Name	Description
31-0	US	User Space. Reserved for user defined space.

Transmit Descriptors

When SWSTYLE (BCR20, bits 7-0) is set to 0, the software structures are defined to be 16 bits wide, and transmit descriptors look like Table 49 (CXDA = Current Transmit Descriptor Address).

When SWSTYLE (BCR 20, bits 7-0) is set to 2, the software structures are defined to be 32 bits wide, and transmit descriptors look like Table 50 (CXDA = Current Transmit Descriptor Address).

When SWSTYLE (BCR 20, bits 7-0) is set to 3, then the software structures are defined to be 32 bits wide, and transmit descriptors look like Table 51 (CXDA = Current Transmit Descriptor Address).

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.

Table 49. 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_ FCS	MORE/ LTINT	ONE	DEF	STP	ENP	TBADR[23:16]
CXDA+04h	1	1	1	1	BCNT				
CXDA+06h	BUFF	UFLO	EX DEF	LCOL	LCAR	RTRY	TDR		

Table 50. Transmit Descriptor (SWSTYLE = 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_ FCS	MORE/ LTINT	ONE	DEF	STP	ENP	BPE	RES	1111	BCNT	
CXDA+08h	BUFF	UFLO	EX DEF	LCOL	LCAR	RTRY	RES	RES	RES	RES	RES	RES	TRC
CXDA+0Ch	USER SPACE												

Table 51. 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	RES					RES	TRC
CXDA+04h	OWN	ERR	ADD_ FCS	MORE/ LTINT	ONE	DEF	STP	ENP	BPE	RES	1111	BCNT	
CXDA+08h	TBADR[31:0]												
CXDA+0Ch	USER SPACE												

TMD1

Bit	Name	Description
31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C971 controller (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The Am79C971 controller clears the OWN bit after transmitting the contents of the buffer. Both the Am79C971 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 Am79C971 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	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 0, FCS generation is controlled by DXMTFCS. When APAD_XMT (CSR4, bit 11) is set to 1, the setting of ADD_FCS has no effect on frames shorter than 64 bytes. ADD_FCS is set by the host, and is not changed by the Am79C971 controller. This is a reserved bit in the C-LANCE (Am79C90) controller.
28	MORE/LTINT	Bit 28 always functions as MORE. The value of MORE is written by the Am79C971 controller and is read by the host. When LTINTEN is cleared to 0 (CSR5, bit 14), the Am79C971 controller will never look at the contents of bit 28, write operations by the host have no effect. When LTINTEN is set to 1 bit 28 changes its function to LTINT on host write operations and on Am79C971 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 Am79C971 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 0 and ENP is set to 1, the Am79C971 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 1. When LTINT is cleared to 0, 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 Am79C971 controller. This bit has meaning only if the ENP bit is set.
26	DEF	Deferred indicates that the Am79C971 controller had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the Am79C971 controller is ready to transmit. DEF is set by the Am79C971 controller and cleared by the host.
25	STP	Start of Packet indicates that this is the first buffer to be used by the Am79C971 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 Am79C971 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 Am79C971 controller.

24	ENP	End of Packet indicates that this is the last buffer to be used by the Am79C971 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 Am79C971 controller.
23	BPE	Bus Parity Error is set by the Am79C971 controller when a parity error occurred on the bus interface during a data transfers from the transmit buffer associated with this descriptor. The Am79C971 controller will only set BPE when the advanced parity error handling is enabled by setting APERREN (BCR20, bit 10) to 1. BPE is set by the Am79C971 controller and cleared by the host. This bit does not exist, when the Am79C971 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
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 Am79C971 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 Am79C971 controller. This field is written by the host and is not changed by the Am79C971 controller. There are no minimum buffer size restrictions.

TMD2

Bit	Name	Description
31	BUFF	Buffer error is set by the Am79C971 controller during transmission when the Am79C971 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 buffer is 0. 2. FIFO underflow occurred before the Am79C971 controller obtained the STATUS byte (TMD1[31:24]) of the next descriptor. BUFF is set by the Am79C971 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 Am79C971 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 0, the transmitter is turned off when an UFLO error occurs (CSR0, TXON = 0). When DXSUFLO is set to 1, the Am79C971 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 Am79C971 controller and cleared by the host.
29	EXDEF	Excessive Deferral. Indicates that the transmitter has experienced Excessive Deferral on this trans-

		mit frame, where Excessive Deferral is defined in the ISO 8802-3 (IEEE/ANSI 802.3) standard. 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 channel slot time has elapsed. The Am79C971 controller does not retry on late collisions. LCOL is set by the Am79C971 controller and cleared by the host.
27	LCAR	Loss of Carrier is set when the carrier is lost during an Am79C971 controller initiated transmission when in AUI mode and the device is operating in half-duplex mode. The Am79C971 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 Am79C971 controller and cleared by the host. LCAR does not reflect a link status problem on the external PHY. 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 1 in the MODE register, RTRY will set after one failed transmission attempt. RTRY is set by the Am79C971 controller and cleared by the host.
25-16	RES	Reserved locations.
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 0. In this case only, the Transmit Retry Count value of 0 should be interpreted as meaning 16. TRC is written by the Am79C971 controller into the last transmit descriptor of a frame, or when an error terminates a frame. Valid only when OWN is cleared to 0.

TMD3

Bit	Name	Description
31-0	US	User Space. Reserved for user defined space.

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	10h
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	RW	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 - 2Bh	Reserved	8	RO	00h
2Ch	PCI Subsystem Vendor ID	16	RO	00h
2Eh	PCI Subsystem ID	16	RO	00h
30h	PCI Expansion ROM Base Address	32	RW	0000 0000h
34h - 3Bh	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

Control and Status Registers

RAP Addr	Symbol	Default Value	Comments	Use
00	CSR0	uuuu 0004	Am79C971 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 1	R
06	CSR6	uuuu uuuu	RXTX: RX/TX Encoded Ring Lengths	S
07	CSR7	0uuu 0000	Extended Control and Interrupt 1	R
08	CSR8	uuuu uuuu	LADRF0: Logical Address Filter — LADRF[15:0]	S
09	CSR9	uuuu uuuu	LADRF1: Logical Address Filter — LADRF[31:16]	S
10	CSR10	uuuu uuuu	LADRF2: Logical Address Filter — LADRF[47:32]	S
11	CSR11	uuuu uuuu	LADRF3: 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 register description	MODE: Mode Register	S
16	CSR16	uuuu uuuu	IADRL: Base Address of INIT Block Lower (Copy)	T
17	CSR17	uuuu uuuu	IADRH: Base Address of INIT Block Upper (Copy)	T
18	CSR18	uuuu uuuu	CRBAL: Current RCV Buffer Address Lower	T
19	CSR22	uuuu uuuu	CRBAU: Current RCV Buffer Address Upper	T
20	CSR20	uuuu uuuu	CXBAL: Current XMT Buffer Address Lower	T
21	CSR21	uuuu uuuu	CXBAU: Current XMT Buffer Address Upper	T
22	CSR22	uuuu uuuu	NRBAL: Next RCV Buffer Address Lower	T
23	CSR23	uuuu uuuu	NRBAU: Next RCV Buffer Address Upper	T
24	CSR24	uuuu uuuu	BADRL: Base Address of RCV Ring Lower	S
25	CSR25	uuuu uuuu	BADRU: Base Address of RCV Ring Upper	S
26	CSR26	uuuu uuuu	NRDAL: Next RCV Descriptor Address Lower	T
27	CSR27	uuuu uuuu	NRDAU: Next RCV Descriptor Address Upper	T
28	CSR28	uuuu uuuu	CRDAL: Current RCV Descriptor Address Lower	T
29	CSR29	uuuu uuuu	CRDAU: Current RCV Descriptor Address Upper	T
30	CSR30	uuuu uuuu	BADXL: Base Address of XMT Ring Lower	S
31	CSR31	uuuu uuuu	BADXU: Base Address of XMT 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

Note:

u = undefined value, R = Running register, S = Setup register, T = Test register; all default values are in hexadecimal format.

CONTROL AND STATUS REGISTERS (CONTINUED)

RAP Addr	Symbol	Default Value After H_RESET	Comments	Use
34	CSR34	uuuu uuuu	CXDAL: Current XMT Descriptor Address Lower	T
35	CSR35	uuuu uuuu	CXDAU: Current XMT 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
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	T
43	CSR43	uuuu uuuu	CXST: Current Transmit Status	T
44	CSR44	uuuu uuuu	NRBC: Next RCV Byte Count	T
45	CSR45	uuuu uuuu	NRST: Next RCV Status	T
46	CSR46	uuuu uuuu	TXDPOLL: Transmit Poll Time Counter	T
47	CSR47	uuuu uuuu	TXPI: Transmit Polling Interval	S
48	CSR48	uuuu uuuu	RXPOLL: Receive Poll Time Counter	
49	CSR49	uuuu uuuu	RXPI: Receive Polling Interval	
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 register description	SWS: Software Style	S
59	CSR59	uuuu uuuu	Reserved	T
60	CSR60	uuuu uuuu	PXDAL: Previous XMT Descriptor Address Lower	T
61	CSR61	uuuu uuuu	PXDAU: Previous XMT 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	NXBAL: Next XMT Buffer Address Lower	T
65	CSR65	uuuu uuuu	NXBAU: Next XMT 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	

Control and Status Registers (Continued)

RAP Addr	Symbol	Default Value After H_RESET	Comments	Use
71	CSR71	uuuu uuuu	Reserved	
72	CSR72	uuuu uuuu	RCVRC: RCV Ring Counter	T
73	CSR73	uuuu uuuu	Reserved	
74	CSR74	uuuu uuuu	XMTRC: XMT Ring Counter	T
75	CSR75	uuuu uuuu	Reserved	
76	CSR76	uuuu uuuu	RCVRL: RCV Ring Length	S
77	CSR77	uuuu uuuu	Reserved	
78	CSR78	uuuu uuuu	XMTRL: XMT Ring Length	S
79	CSR79	uuuu uuuu	Reserved	
80	CSR80	uuuu 1410	DMATCFW: DMA Transfer Counter and FIFO Threshold	S
81	CSR81	uuuu uuuu	Reserved	
82	CSR82	uuuu uuuu	Transmit Descriptor Pointer Address Lower	S
83	CSR83	uuuu uuuu	Reserved	
84	CSR84	uuuu uuuu	DMABA: Address Register Lower	T
85	CSR85	uuuu uuuu	DMABA: Address Register Upper	T
86	CSR86	uuuu uuuu	DMABC: Buffer Byte Counter	T
87	CSR87	uuuu uuuu	Reserved	
88	CSR88	x262 3003	Chip ID Register Lower	T
89	CSR89	uuuu x262	Chip ID Register Upper	T
90	CSR90	uuuu uuuu	Reserved	
91	CSR91	uuuu uuuu	Reserved	T
92	CSR92	uuuu uuuu	RCON: Ring Length Conversion	T
93	CSR93	uuuu uuuu	Reserved	
94	CSR94	uuuu uuuu	Reserved	
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 Timeout	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	

CONTROL AND STATUS REGISTERS (CONCLUDED)

RAP Addr	Symbol	Default Value After H_RESET	Comments	Use
108	CSR108	uuuu uuuu	Reserved	
109	CSR109	uuuu uuuu	Reserved	
110	CSR110	uuuu uuuu	Reserved	
111	CSR111	uuuu uuuu	Reserved	
112	CSR112	uuuu uuuu	Missed Frame Count	R
113	CSR113	uuuu uuuu	Reserved	
114	CSR114	uuuu uuuu	Received Collision Count	R
115	CSR115	uuuu uuuu	Reserved	
116	CSR116	uuuu 0200	Reserved	
117	CSR117	uuuu uuuu	Reserved	
118	CSR118	uuuu uuuu	Reserved	
119	CSR119	uuuu 0105	Reserved	
120	CSR120	uuuu uuuu	Reserved	
121	CSR121	uuuu uuuu	Reserved	
122	CSR226	uuuu 0000	Receive Frame Alignment Control	S
123	CSR237	uuuu uuuu	Reserved	
124	CSR248	uuuu 0000	Test Register 1	T
125	CSR125	003c 0060	MAC Enhanced Configuration Control	T
126	CSR126	uuuu uuuu	Reserved	
127	CSR127	uuuu uuuu	Reserved	

Bus Configuration Registers

Writes to those registers marked as “Reserved” will have no effect. Reads from these locations will produce undefined values.

RAP	Mnemonic	Default	Name	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	LED0	00C0h	LED0 Status	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	0200h	Software Style	Yes	No
21	INTCON	N/A	Reserved	No	No
22	PCILAT	FF06h	PCI Latency	Yes	Yes
23	PCISID	0000h	PCI Subsystem ID	No	Yes
24	PCISVID	0000h	PCI Subsystem Vendor ID	No	Yes
25	SRAMSIZ	0000h	SRAM Size	Yes	Yes
26	SRAMB	0000h	SRAM Boundary	Yes	Yes
27	SRAMIC	0000h	SRAM Interface Control	Yes	Yes
28	EBADDRL	N/A	Expansion Bus Address Lower	Yes	No
29	EBADDRU	N/A	Expansion Bus Address Upper	Yes	No
30	EBDR	N/A	Expansion Bus Data Port	Yes	No
31	STVAL	FFFFh	Software Timer Value	Yes	No
32	MIICAS	0000h	MII Control and Status	Yes	Yes
33	MIIADDR	N/A	MII Address	Yes	Yes
34	MIIMDR	N/A	MII Management Data	Yes	No
35	PCIVID	1022h	PCI Vendor ID	No	Yes

REGISTER PROGRAMMING SUMMARY

Am79C971 Programmable Registers

Am79C971 Control and Status Registers

Register	Contents			
CSR0	Status and control bits: (DEFAULT = 0004)			
	8000 ERR 4000 BABL 2000 CERR 1000 MISS	0800 MERR 0400 RINT 0200 TINT 0100I IDON	0080 INTR 0040 IENA 0020 RXON 0010 TXON	0008 TDMD 0004 STOP 0002 STRT 0001 INIT
CSR1	Lower IADR (Maps to CSR 16)			
CSR2	Upper IADR (Maps to CSR 17)			
CSR3	Interrupt masks and Deferral Control: (DEFAULT = 0)			
	8000-- 4000 BABLM 2000 - 1000 MISSM	0800 MERRM 0400 RINTM 0200 TINTM 0100I DONM	0080- - 0040 DXSUFLO 0020 LAPPEN 0010 DXMT2PD	0008 EMBA 0004 BSWP 0002 -- 0001 --
CSR4	Interrupt masks, configuration and status bits: (DEFAULT = 0115)			
	8000 EN124 4000 DMAPLUS 2000 TIMER 1000 TXDPOLL	0800 APAD_XMT 0400 ASTRP_RCV 0200 MFCO 0100 MFCOM	0080 UNITCMD 0040 UNIT 0020 RCVCCO 0010 RCVCCOM	0008 TXSTRT 0004 TXSTRTM 0002 JAB 0001 JABM
CSR5	Extended Interrupt masks, configuration and status bits: (DEFAULT = 0XXX)			
	8000TOKINTD 4000LTINTEN 2000-- 1000--	0800 SINT 0400 SINTE 0200 SLPINT 0100 SLPINTE	0080 EXDINT 0040 EXDINTE 0020 MPPLBA 0010 MPINT	0008 MPINTE 0004 MPEN 0002 MPMODE 0001 SPND
CSR7	Extended Interrupt masks, configuration and status bits: (DEFAULT = 0000)			
	8000 FASTSPND 4000 RXFRMTG 2000 RDMD 1000 RXDPOLL	0800 STINT 0400 STINTE 0200 MREINT 0100 MREINTE	0080 MAPINT 0040 MAPINTE 0020 MCCINT 0010 MCCINTE	0008 MCCIINT 0004 MCCIINTE 0002 MIIPDTINT 0001 MIIPDTNTE
CSR8 - CSR11	Logical Address Filter			
CSR12 - CSR14	Physical Address Register			
CSR15	MODE: (DEFAULT = 0) bits [8:7] = PORTSEL, Port Selection 0000AUI port 008010BASE-T 0101Media Independent Interface			
	8000PROM 4000DRCVBC 2000DRCVPA 1000DLNKTST	0800 DAPC 0400 MENDECL 0200 LRT/TSEL 0100 PORTSEL1	0080 POTSEL0 0040 INTL 0020 DRTY 0010 FCOLL	0008 DXMTFCS 0004 LOOP 0002 DTX 0001 DRX
CSR47	TXPOLLINT: Transmit Polling Interval			
CSR49	RXPOLLINT: Receive Polling Interval			

AM79C971 CONTROL AND STATUS REGISTERS (CONTINUED)

Register	Contents
CSR58	Software Style (mapped to BCR20) bits [7:0] = SWSTYLE, Software Style Register. 0000 LANCE/PCnet-ISA 0002 PCnet-32
	8000 -- 0800 -- 0080 SWSTYLE7 0008 SWSTYLE3 4000 -- 0400 APERREN 0040 SWSTYLE6 0004 SWSTYLE2 2000 -- 0200 CSRPCNET 0020 SWSTYLE5 0002 SWSTYLE1 1000 -- 0100 SSIZE32 0010 SWSTYLE4 0001 SWSTYLE0
CSR76	RCVRL: RCV Descriptor Ring length
CSR78	XMTRL: XMT Descriptor Ring length
CSR80	FIFO threshold and DMA burst control. (DEFAULT = 2810) 8000 Reserved 4000 Reserved bits [13:12] = RCVFW, Receive FIFO Watermark 0000 Request DMA when 16 bytes are present 1000 Request DMA when 64 bytes are present 2000 Request DMA when 112 bytes are present 3000 Reserved bits [11:10] = XMTSP, Transmit Start Point 0000 Start transmission after 20/44 (No SRAM/SRM) bytes have been written 0400 Start transmission after 64 bytes have been written 0800 Start transmission after 128 bytes have been written 0C00 Start transmission after 248 bytes (full packet) have been written bits [9:8] = XMTFW, Transmit FIFO Watermark 0000 Start DMA when 16 write cycles can be made 0100 Start DMA when 64 write cycles can be made 0200 Start DMA when 108 write cycles can be made 0300 Reserved bits [7:0] = DMA Transfer Counter
CSR88~89	Chip ID (Contents = v2623003; v = Version Number)
CSR112	Missed Frame Count
CSR114	Receive Collision Count
CSR122	Receive Frame Alignment Control
	8000 -- 0800 -- 0080 -- 0008 -- 4000 -- 0400 -- 0040 -- 0004 -- 2000 -- 0200 -- 0020 -- 0002 -- 1000 -- 0100 -- 0010 -- 0001 RCVALGN
CSR124	BMU Test Register (DEFAULT = 0000)
	8000 -- 0800 -- 0080 -- 0008 GPSIEN 4000 -- 0400 -- 0040 -- 0004 RPA 2000 -- 0200 -- 0020 -- 0002 -- 1000 -- 0100 -- 0010 -- 0001 --
CSR125	MAC Enhanced Configuration Control (DEFAULT = 603c bits [15:8] = IPG, InterPacket Gap (Default=60xx, 96 bit times) bits [8:0] = IFS1, InterFrame Space Part 1 (Default=xx3c, 60 bit times)

Am79C971 Bus Configuration Registers

RAP Addr	Register	Contents							
0	MSRDA	Programs width of DMA read signal (DEFAULT = 5)							
1	MSWRA	Programs width of DMA write signal (DEFAULT = 5)							
2	MC	Miscellaneous Configuration bits: (DEFAULT = 2)							
		8000	--	0800	--	0080	INITLEVEL	0008	EADISEL
		4000	TMAULOOP	0400	--	0040	--	0004	AWAKE
		2000	--	0200	--	0020	--	0002	ASEL
		1000	--	0100	APROMWE	0010	DRACC	0001	XMAUSEL
4	LED0	Programs the function and width of the LED0 signal. (DEFAULT = 00C0)							
		8000	LEDOUT	0800	MIISE	0080	PSE	0008	RXPOLE
		4000	LEDPOL	0400	DXCVRCTL	0040	LINKSE	0004	RCVE
		2000	LEDDIS	0200	MPSE	0020	RCVME	0002	JABE
		1000	100E	0100	FDLSE	0010	XMTE	0001	COLE
5	LED1	Programs the function and width of the LED1 signal. (DEFAULT = 0084)							
		8000	LEDOUT	0800	MIISE	0080	PSE	0008	RXPOLE
		4000	LEDPOL	0400	DXCVRCTL	0040	LINKSE	0004	RCVE
		2000	LEDDIS	0200	MPSE	0020	RCVME	0002	JABE
		1000	100E	0100	FDLSE	0010	XMTE	0001	COLE
6	LED2	Programs the function and width of the LED2 signal. (DEFAULT = 0088)							
		8000	LEDOUT	0800	MIISE	0080	PSE	0008	RXPOLE
		4000	LEDPOL	0400	DXCVRCTL	0040	LINKSE	0004	RCVE
		2000	LEDDIS	0200	MPSE	0020	RCVME	0002	JABE
		1000	100E	0100	FDLSE	0010	XMTE	0001	COLE
7	LED3	Programs the function and width of the LED3 signal. (DEFAULT = 0090)							
		8000	LEDOUT	0800	MIISE	0080	PSE	0008	RXPOLE
		4000	LEDPOL	0400	DXCVRCTL	0040	LINKSE	0004	RCVE
		2000	LEDDIS	0200	MPSE	0020	RCVME	0002	JABE
		1000	100E	0100	FDLSE	0010	XMTE	0001	COLE
9	FDC	Full-Duplex Control. (DEFAULT= 0000)							
		8000	--	0800	--	0080	--	0008	--
		4000	--	0400	--	0040	--	0004	FDRPAD
		2000	--	0200	--	0020	--	0002	AUIFD
		1000	--	0100	--	0010	--	0001	FDEN
16	IOBASEL	I/O Base Address Lower							
17	IOBASEU	I/O Base Address Upper							
18	BSBC	Burst Size and Bus Control (DEFAULT = 2101)							
		8000	ROMTMG3	0800	NOUFLO	0080	DWIO	0008	TSTSHDW0
		4000	ROMTMG2	0400	--	0040	BREADE	0004	LINBC2
		2000	ROMTMG1	0200	MEMCMD	0020	BWRITE	0002	LINBC1
		1000	ROMTMG0	0100	EXTREQ	0010	TSTSHDW1	0001	LINBC0
19	EECAS	EEPROM Control and Status (DEFAULT = 0002)							
		8000	PVALID	0800	--	0080	--	0008	--
		4000	PREAD	0400	--	0040	--	0004	ECS
		2000	EEDET	0200	--	0020	--	0002	ESK
		1000	--	0100	--	0010	EEN	0001	EDI/ EDO

AM79C971 BUS CONFIGURATION REGISTERS (CONTINUED)

20	SWSTYLE	Software Style (DEFAULT = 0000, maps to CSR 58)					
21	INTCON	Interrupt Control					
		8000 --		0800--		0080--	0008--
		4000 --		0400--		0040--	0004--
		2000 --		0200--		0020--	0002--
		1000 --		0100--		0010--	0001--
22	PCILAT	PCI Latency (DEFAULT = FF06)					
		bits [15:8] = MAX_LAT					
		bits [7:0] = MIN_GNT					
25	SRAMSIZE	SRAM Size (DEFAULT = 000)					
		bits [7:0] = SRAM_SIZE					
26	SRAMBND	SRAM Boundary (DEFAULT = 0000)					
		bits [7:0] = SRAM_BND					
27	SRAMIC	SRAM Interface Control					
		8000 PTR TST 4000 LOLATRX bits [5:3] = EBCS, Expansion Bus Clock Source 0000 CLK pin, PCI clock (external clock not required) 0008 XTAL1 and XTAL2 pins, 20 MHz clock (external clock not required) 0010 EBCLK pin, Expansion Bus Clock (external clock not required) bits [2:0] = CLK_FAC, Expansion Bus Clock Factor 0000 1/1 clock factor 0001 1/2 clock factor 0002 -- 0003 --					
	EPADDRL	Expansion Port Address Lower					
29	EPADDRU	Expansion Port Address Upper					
		8000 FLASH	0800 --	0080 --	0008 EPADDRU3		
		4000 AINC	0400 --	0040 --	0004 EPADDRU2		
		2000 --	0200 --	0020 --	0002 EPADDRU1		
		1000 --	0100 --	0010 --	0001 EPADDRU0		
30	EBDATA	Expansion Bus Data Port					
31	STVAL	Software Timer Interrupt Value (DEFAULT = FFFF)					
32	MIICAS	MII Status and Control (DEFAULT = 0400)					
		8000 ANTST	0800 APEP	0080 DANAS	0008 XPHYSP		
		4000 MIIPD	0400 APDW2	0040 XPHYRST	0004 MIIµL		
		2000 FMDC1	0200 APDW1	0020 XPHYANE	0002 MIILP		
		1000 FMD0	0100 APDW0	0010 XPHYFD	0001 FCON		
33	MIIADDR	MII Address					
		bits [9:5] = PHYAD, Physical Layer Device Address bits [4:0] = REGAD, MII/Auto-Negotiation Register Address					
34	MIIMDR	MII Data Port					

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65°C to +150°C

Ambient Temperature ..... -65°C to +70°C

Supply voltage with respect to AV_{SS}, V_{SSB}, V_{SSM}, V_{SS_PLL}, V_{SS} (AV_{DD}, V_{DDB}, V_{DDM}, V_{DD_PLL}, V_{DD} ..... –0.3 V to +6.0 V

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

Temperature (TA) ..... 0°C to +70°C

Supply Voltages (AV_{DD}, V_{DD_PLL}, V_{DD}) .....+5 V ±5%

(V_{DDB} for 5-V Signaling) .....+5 V ±5%

(V_{DD_PCI} for 3.3-V Signaling) .....+ 3.3 V ±10%

All inputs within the range:

AV_{SS} - 0.5 V _ V_{IN} _ AV_{DD} + 0.5 V,

or V_{SSB} - 0.5 V _ V_{IN} _ V_{DD_PCI} + 0.5 V,

or V_{SS_PLL} - 0.5 V _ V_{IN} _ V_{DD_PLL} + 0.5V,

or V_{SS} - 0.5 V _ V_{IN} _ V_{DD} + 0.5 V,

or V_{SSB} - 0.5 < V_{IN} < V_{DDB} + 0.5V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS OVER COMMERCIAL 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)	$I_{OH\ 1}$= -2 mA $I_{OH\ 2}$= -4 mA (Note 3)	2.4		V
Digital Input Leakage Current for 5-V Signaling					
I_{IX}	Input Low Leakage Current (Note 4)	$V\overline{IN}$ = 0 V, V_{DD} = V_{DDB} = V_{DD_PCI} = 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
Digital Input Voltage for 3.3-V Signaling					
V_{IL}	Input LOW Voltage			-0.5	0.325 V_{DDB}
V_{IH}	Input HIGH Voltage			0.475 V_{DDB}	V_{DDB} + 0.5
Digital Output Voltage for 3.3-V Signaling					
V_{OL}	Output LOW Voltage	IOL = 1.5 mA			0.1 V_{DDB}
V_{OH}	Output HIGH Voltage (Note 2)	IOH = -0.5 mA		0.9 V_{DDB}	

DC CHARACTERISTICS OVER COMMERCIAL OPERATING RANGES UNLESS OTHERWISE SPECIFIED (CONTINUED)

Parameter Symbol	Parameter Description	Test Conditions		Min	Max	Units
Digital Input Leakage Current for 3.3-V Signaling						
I_{IX}	Input Low Leakage Current (Note 4)	V_{IN} = 0 V, V_{DD} = V_{DD_PCI} = 3.3 V		-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_{DD_PCI}			10	µA
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
IILX	XTAL1 Input LOW Current	V_{IN} = External Clock	Active	-120	0	µA
		V_{IN} = VSS	Sleep	-10	+10	µA
IIHX	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, MDC = 2.5 MHz, TX_CLK=RX_CLK=25 MHz			190	mA
I_{DDCOMA}	Sleep Mode Power Supply Current	SLEEP active			700	µA
$I_{DDSNOOZE}$	Auto Wake Mode Power Supply Current	Awake bit set active			20	mA
$I_{DDmagic0}$	Magic Packet Mode Power Supply Current	CLK = 0 MHz (Note 11)			55	mA
$I_{DDmagic33}$	Magic Packet Mode Power Supply Current	CLK = 33 MHz (Note 11)			125	mA
Pin Capacitance						
C_{IN}	Input Pin Capacitance	F_C = 1 MHz (Note 6, 10)			10	pF
C_O	I/O or Output Pin Capacitance	F_C = 1 MHz (Note 6)			10	pF
C_{CLK}	CLK Pin Capacitance	F_C = 1 MHz (Note 6)		5	12	pF
Twisted Pair Interface (10BASE-T)						
I_{IRXD}	Input Current at RXD±	AV_{DD}< V_{IN} < AV_{DD}	-500	500	µA	
R_{RXD}	RXD± differential input resistance		10		kΩ	
V_{TIVB}	RXD±, RXD- open circuit input voltage (Bias)	I_{IN} = 0 mA	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	

DC CHARACTERISTICS OVER COMMERCIAL OPERATING RANGES UNLESS OTHERWISE SPECIFIED (CONTINUED)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
Twisted Pair Interface (10BASE-T) (Cont'd)					
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
V_{RXDTH}	RXD switching threshold	(Note 4)	-35	35	mV
V_{TXH}	TXD± and TXP± output HIGH voltage	V_{SS}= 0 V	V_{DD} - 0.6	V_{DD}	V
V_{TXL}	TXD± and TXP± output LOW voltage	V_{DD}= 5 V	V_{SS}	V_{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 4)		80	_
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 < VIN < AVDD + 0.5 V	-500	+500	µA
V_{AOD}	Differential Output Voltage |(DO+)-(DO-)|	RL = 78 Ω	630	1200	mV
V_{AODOFF}	Transmit Differential Output Idle Voltage	RL = 78 Ω (Note 9)	-40	40	mV
I_{AODOFF}	Transmit Differential Output Idle Current	RL = 78 Ω (Note 8)	-1	1	mA
V_{CMT}	Transmit Output Common Mode Voltage	RL = 78 Ω	2.5	AVDD	V
V_{ODI}	DO± Transmit Differential Output Voltage Imbalance	RL = 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

Notes:

2. OL1 applies to AD[31:00], C/$\overline{BE}$[3:0], PAR and $\overline{REQ}$
OL2 applies to $\overline{DEVSEL}$, $\overline{FRAME}$, $\overline{INTA}$, $\overline{IRDY}$, $\overline{PERR}$, $\overline{SERR}$, $\overline{STOP}$, $\overline{TRDY}$, EECS, EEDI, EBUA_EBA[7:0], EBDA[15:8], EBD[7:0], $\overline{EROMCS}$, $\overline{ERAMCS}$, AS_$\overline{EBOE}$, $\overline{EBWE}$, TXD[3:0], TX_EN, TX_ER, MDC, MDIO and TDO.

IOL3 applies to EESK/$\overline{LED1}$/SFBD, $\overline{LED2}$/SRDCLK, EEDO/$\overline{LED3}$/SRD, and EEDI/$\overline{LED0}$.

3. *VOH does not apply to open-drain output pins.*
4. *OH1 applies to all other outputs.*
   IOH2 applies to TXD[3:0], TX_EN,TX_ER,MDC, and MDIO
   Outputs are CMOS and will be driven to rail if the load is not resistive.
5. *IL and IIH apply to all input pins except XTAL1.*
6. *OZL and IOZH apply to all three-state output pins and bidirectional pins.*
7. *Parameter not tested. Value determined by characterization.*
8. *Tested, but to values in excess of limits. Test accuracy not sufficient to allow screening guard bands.*
9. *Correlated to other tested parameters - not tested directly.*
10. *Test not implemented to data sheet specification.*
11. *CIN = 8 pF for the IDSEL input and all input pins on the MII interface (TX_CLK, COL, CRS, RX_CLK, RXD[3:0], RX_DV, RX_ER).*
12. *The power supply current in Magic Packet mode is linear with respect to the PCI Clock frequency operation, assuming the network port remains constant. For example, at CLK = 20 MHz, the maximum Magic Packet power supply current would be 87 mA.*

SWITCHING CHARACTERISTICS: BUS INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
Clock Timing - PCI Bus Interface					
F_{CLK}	CLK Frequency		0	33	MHz
t_{CYC}	CLK Period	@ 1.5 V for V_{DDB} = 5 V @ 0.4 V_{DD_PCI} V for V_{DD_PCI} = 3.3 V	30	–	ns
t_{HIGH}	CLK High Time	@ 2.0 V for V_{DDB} = 5 V @ 0.475 V_{DD_PCI} V for V_{DD_PCI} = 3.3 V	12		ns
t_{LOW}	CLK Low Time	@ 0.8 V for V_{DDB} = 5 V @ 0.325 V_{DD_PCI} V for V_{DD_PCI} = 3.3 V	12		ns
t_{FALL}	CLK Fall Time	over 2 V p-p for V_{DDB} = 5 V over 0.4 V_{DD_PCI} p-p for V_{DD_PCI} = 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_{DD_PCI} p-p for V_{DD_PCI} = 3.3 V (Note 1)	1	4	V/ns
Output and Float Delay Timing - PCI Bus Interface					
t_{VAL}	AD[31:00], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, DEVSEL, PERR, SERR Valid Delay		2	11	ns
t_{VAL} (REQ)	REQ Valid Delay		2	12	ns
t_{ON}	AD[31:00], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, DEVSEL Active Delay		2		ns
t_{OFF}	AD[31:00], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, DEVSEL Float Delay			28	ns

SWITCHING CHARACTERISTICS: BUS INTERFACE (CONTINUED)

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
Setup and Hold Timing					
t_{SU}	AD[31:00], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, DEVSEL, IDSEL Setup Time		7		ns
t_H	AD[31:00], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, DEVSEL, IDSEL Hold Time		0		ns
t_{SU} (GNT)	GNT Setup Time		10		ns
t_H (GNT)	GNT Hold Time		0		ns
EEPROM Timing					
f_{EESK}	EESK Frequency	(Note 2)		650	kHz
t_{HIGH} (EESK)	EESK High Time		780		ns
t_{LOW} (EESK)	EESK Low Time		780		ns
t_{VAL} (EEDI)	EEDI Valid Output Delay from EESK	(Note 2)	-15	15	ns
t_{VAL} (EECS)	EECS Valid Output Delay from EESK	(Note 2)	-15	15	ns
t_{LOW} (EECS)	EECS Low Time		1550		ns
t_{SU} (EEDO)	EEDO Setup Time to EESK	(Note 2)	50		ns
t_H (EEDO)	EEDO Hold Time from EESK	(Note 2)	0		ns
JTAG (IEEE 1149.1) Test Signal Timing					
t_{J1}	TCK Frequency			10	MHz
t_{J2}	TCK Period		100		ns
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_{J9}	TDO Valid Delay		3	30	ns
t_{J10}	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 Inputs (Non-Test)) Setup Time		8		ns
t_{J14}	All Inputs (Non-Test) Hold Time		7		ns

Notes:

1. *Not tested; parameter guaranteed by design 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: BUS INTERFACE (CONCLUDED)

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
Expansion Bus Interface Timing					
F_{CLK}	EBCLK Frequency		2.5	33	MHz
t_{CYC}	EBCLK Period	@ 1.5 V	30	400	ns
t_{HIGH}	EBCLK High Time	@ 2.0 V	12		ns
t_{LOW}	EBCLK Low Time	@ 0.8 V	12		ns
t_{FALL}	EBCLK Fall Time	over 2 V p-p	1	4	V/ns
t_{RISE}	EBCLK Rise Time	over 2 V p-p	1	4	V/ns
$t_{v\_A\_D}$	Address and Data valid time from the rising edge of EBCLK			13	ns
t_{CS_L}	ERAMCS assert time from the rising edge of EBCLK			13	ns
t_{CS_H}	ERAMCS deassert time from rising edge of EBCLK		7	10	ns
t_{s_D}	Data setup time to the rising edge of EBCLK		0		ns
t_{h_D}	Data hold time from the rising edge of EBCLK		6		ns
$t_{\_AS\_H}$	AS_EBOE (Address Strobe) rising edge from the falling edge of EBCLK		10	13	ns
$t_{\_AS\_L}$	AS_EBOE falling edge from the rising edge of EBCLK		10	13	ns
$t_{\_OE\_H}$	AS_EBOE (Output enable deassert) rising edge from the rising edge of EBCLK		7		ns
$t_{\_LZ}$	Data Bus driving from the rising edge of EBCLK		5		ns
$t_{\_HZ}$	Data Bus tristated from the rising edge of EBCLK			13	ns
$t_{\_WE\_L}$	EBWE assert time from the falling edge of EBCLK			6	ns
$t_{\_WE\_H}$	EBWE deassert time from the rising edge of EBCLK			10	ns
$t_{\_WE\_CSAD}$	EBWE setup time to Address, ERAMCS, and Data		1		ns

Note:
Address is the EBUA_EBA[7:0] bus. Data is the EBDA[15:8] and the EBD[7:0] buses.

SWITCHING CHARACTERISTICS: 10BASE-T INTERFACE

Parameter Symbol	Parameter Name	Test Condition	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	$t_{TM} = \| t_{TR} - t_{TF} \|$		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	(See note below)	75	120	ns
t_{PWPLP}	Predistortion Idle Link Pulse Width	(See note below)	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_{RCVOFF}	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:
Not tested, parameter guaranteed by characterization.

SWITCHING CHARACTERISTICS: ATTACHMENT UNIT INTERFACE

Parameter Symbol	Parameter Name	Test Condition	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	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 1)	15	45	ns
t_{PWKDI}	DI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 2)	136	200	ns
t_{PWOCI}	CI Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 3)	10	26	ns
t_{PWKCI}	CI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (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

Notes:

1. *DI pulses narrower than tP_{WODI} (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: MEDIA INDEPENDENT INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
Transmit Timing					
t_{TVAL}	TX_EN, TX_ER, TXD valid from ↑ TX_CLK	measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V (Note 1)	0	25	ns
Receive Timing					
t_{RSU}	RX_DV, RX_ER, RXD setup to ↑ RX_CLK	measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V (Note 1)	10		ns
t_{RH}	RX_DV, RX_ER, RXD hold to ↑ RX_CLK	measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V (Note 1)	10		ns
Management Cycle Timing					
t_{MHIGH}	MDC Pulse Width HIGH Time	C_{LOAD} = 390 pf	160		ns
t_{MLOW}	MDC Pulse Width LOW Time	C_{LOAD} = 390 pf	160		ns
t_{MCYC}	MDC Cycle Period	C_{LOAD} = 390 pf	400		ns
t_{MSU}	MDIO setup to ↑ MDC	C_{LOAD} = 470 pf, measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V (Note 1)	10		ns
t_{MH}	MDIO hold to ↑ MDC	C_{LOAD} = 470 pf, measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V (Note 1)	10		ns
t_{MVAL}	MDIO valid from ↑ MDC	C_{LOAD} = 470 pf, measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V, (Note 1)		t_{MCYC} - t_{MSU}	ns

Notes:

1. *MDIO valid measured at the exposed mechanical Media Independent Interface.*
2. *TXCLK and RXCLK frequency and timing parameters are defined for the external physical layer transceiver as defined in the IEEE 802.3u standard. They are not replicated here.*

SWITCHING CHARACTERISTICS: GENERAL-PURPOSE SERIAL INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
Transmit Timing					
t_{GPT1}	TXCLK Period (802.3 compliant)	@ 1.5 V	99.99	100.01	ns
t_{GPT2}	TXCLK HIGH Time	@ 2.0 V	40	60	ns
t_{GPT3}	TXDAT and TXEN Delay from ↑ TXCLK	@ 1.5 V	0	70	ns
t_{GPT4}	RXEN Setup before ↑ TXCLK (Last Bit)	@ 1.5 V	210		ns
t_{GPT5}	RXEN Hold after ↓ TXEN	@ 1.5 V	0		ns
t_{GPT6}	CLSN Active Time to Trigger Collision	@ 1.5 V (Note 1)	410		ns
t_{GPT7}	CLSN Active to ↓ RXEN to Prevent LCAR Assertion	@ 1.5 V	0		ns
t_{GPT8}	CLSN Active to ↓ RXEN for SQE Heartbeat window	@ 1.5 V	0	4.0	µs
t_{GPT9}	CLSN Active to ↑ RXEN for Normal Collision	@ 1.5 V	0	51.2	µs
Receive Timing					
t_{GPR1}	RXCLK Period	@ 1.5 V (Note 2)	80	120	ns
t_{GPR2}	RXCLK HIGH Time	@ 2.0 V (Note 2)	30	80	ns
t_{GPR3}	RXCLK LOW Time	@ 0.8 V (Note 2)	30	80	ns
t_{GPR4}	RXDAT and RXEN Setup to ↑ RXCLK	@ 1.5 V	15		ns
t_{GPR5}	RXDAT Hold after ↑ RXCLK	@ 1.5 V	15		ns
t_{GPR6}	RXEN Hold after ↓ RXCLK	@ 1.5 V	0		ns
t_{GPR7}	CLSN Active to First ↑ RXCLK (Collision Recognition)	@ 1.5 V	0		ns
t_{GPR8}	CLSN Active to ↑ RXCLK for Address Type Designation Bit	@ 1.5 V (Note 3)	51.2		µs
t_{GPR9}	CLSN Setup to Last ↑ RXCLK for Collision Recognition	@ 1.5 V	210		ns
t_{GPR10}	CLSN Active	@ 1.5 V	410		ns
t_{GPR11}	CLSN Inactive Setup to First ↑ RXCLK	@ 1.5 V	300		ns
t_{GPR12}	CLSN Inactive Hold to Last ↑ RXCLK	@ 1.5 V	300		ns

Notes:

1. *CLSN must be asserted for a continuous period of 110 ns or more. Assertion for less than 110 ns period may or may not result in CLSN recognition.*
2. *RXCLK should meet jitter requirements of IEEE 802.3 specification.*
3. *CLSN assertion before 51.2 µs will be indicated as a normal collision. CLSN assertion after 51.2 µs will be considered as a Late Receive Collision.*

SWITCHING CHARACTERISTICS: EXTERNAL ADDRESS DETECTION INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
External Address Detection Interface: Internal PHY					
t_{EAD1}	SRD setup to ↑ SRDCLK		40		ns
t_{EAD2}	SRD hold to ↑ SRDCLK		40		ns
t_{EAD3}	SFBD# change to ↓ SRDCLK		-15	+15	ns
t_{EAD4}	EAR deassertion to ↑ SRDCLK (first rising edge)		50		ns
t_{EAD5}	$\overline{EAR}$ assertion after SFD event (frame rejection)		0	51,090	ns
t_{EAD6}	$\overline{EAR}$ assertion width		110		ns
External Address Detection Interface: External PHY - MII @ 25 MHz					
t_{EAD7}	SFBD change from ↓ RX_CLK		0	20 (Note 1)	ns
t_{EAD8}	EAR deassertion to ↑ RX_CLK (first rising edge)		40		ns
t_{EAD9}	$\overline{EAR}$ assertion after SFD event (frame rejection)		0	5,080	ns
t_{EAD10}	EAR assertion width		50		ns
External Address Detection Interface: External PHY - MII @ 2.5 MHz					
t_{EAD11}	EAR deassertion to ↑ RX_CLK (first rising edge)		400		ns
t_{EAD12}	$\overline{EAR}$ assertion after SFD event (frame rejection)		0	50,800	ns
t_{EAD13}	EAR assertion width		500		ns
Receive Frame Tag Timing with Media Independent Interface					
t_{EAD14}	RXFRTGE assertion from ↑SFBD (first rising edge)		0		ns
t_{EAD15}	RXFRTGE, RXFRTGD setup to ↑ RX_CLK		10		ns
t_{EAD16}	RXFRTGE, RXFRTGD hold to ↑ RX_CLK		10		ns
t_{EAD17}	RXFRTGE deassertion to ↓ RX_DV	RX_CLK @25 MHz RX_CLK @2.5 MHz	40 400		ns ns

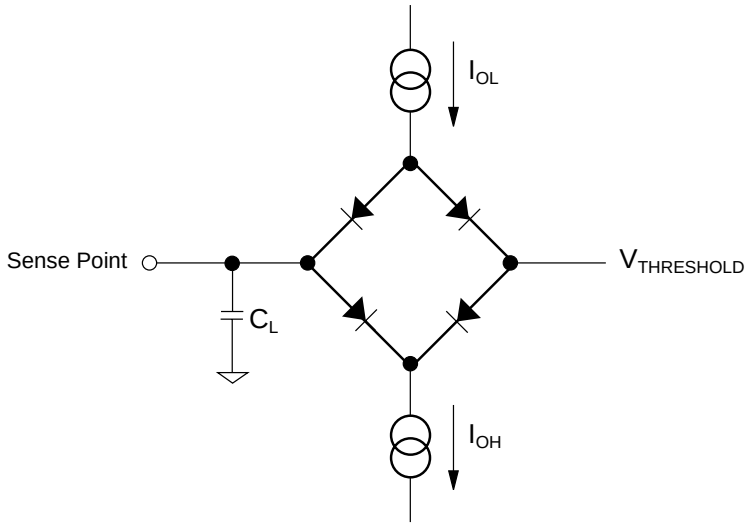
Note:

1. May need to delay RX_CLK to capture Start Frame Byte Delimiter (SFBD) at 100 Mbps operation.

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
	DonÕt Care; Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance ÒOffÓ State

SWITCHING TEST CIRCUITS



20550D-61

Figure 58. Normal and Tri-State Outputs

SWITCHING TEST CIRCUITS (CONTINUED)

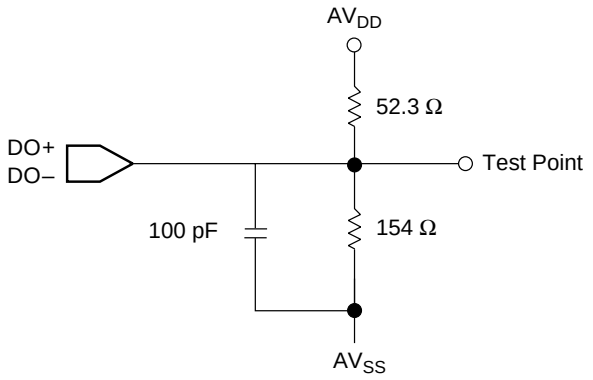


Figure 59. AUI DO Switching Test Circuit

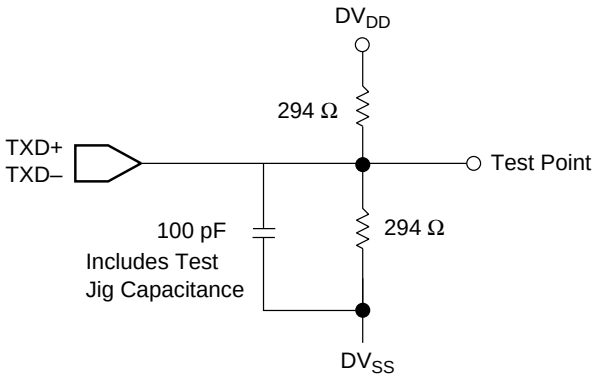


Figure 60. TXD Switching Test Circuit

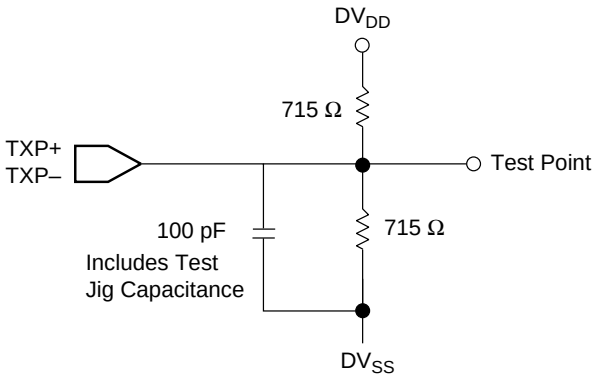


Figure 61. TXP Outputs Test Circuit

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE

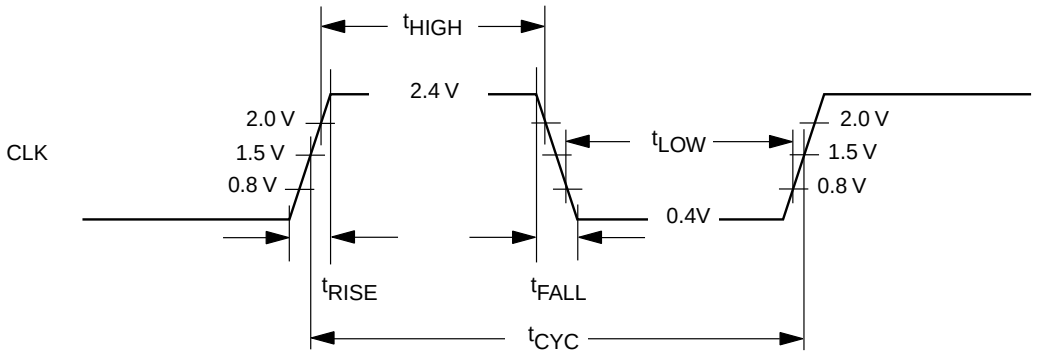


Figure 62. CLK Waveform for 5 V Signaling

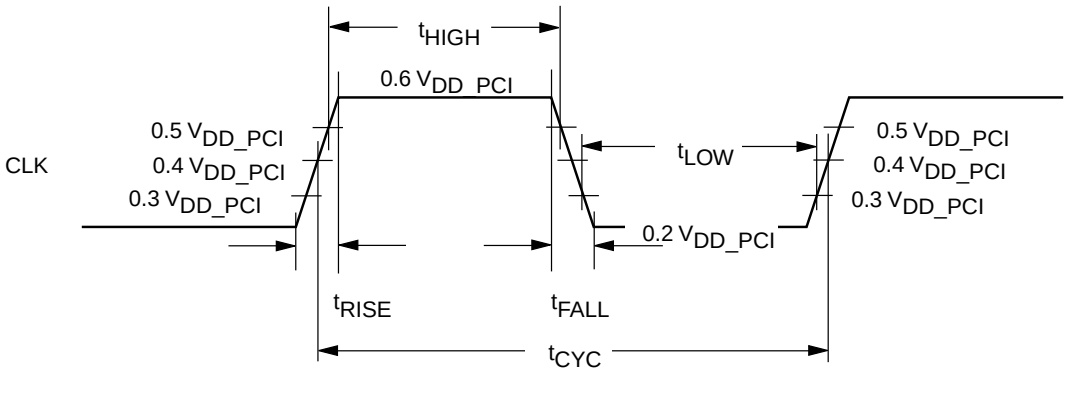


Figure 63. CLK Waveform for 3.3 V Signaling

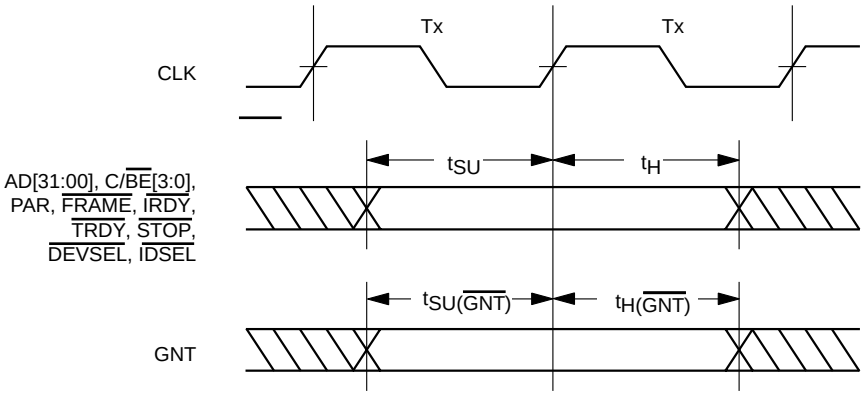


Figure 64. Input Setup and Hold Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE (CONTINUED)

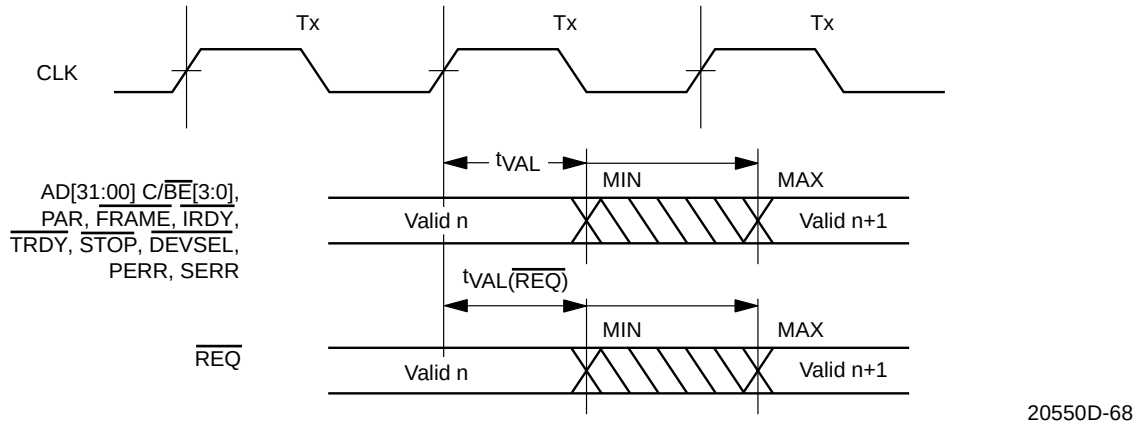


Figure 65. Output Valid Delay Timing

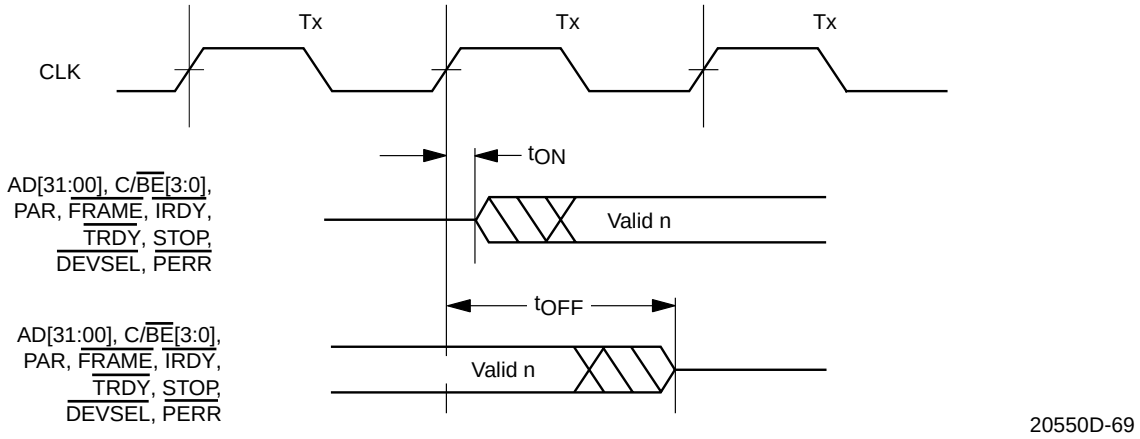


Figure 66. Output Tri-state Delay Timing

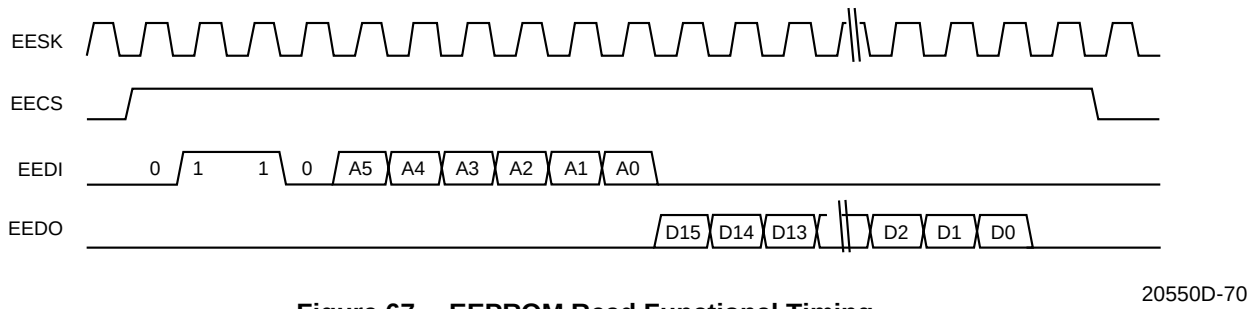


Figure 67. EEPROM Read Functional Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE (CONTINUED)

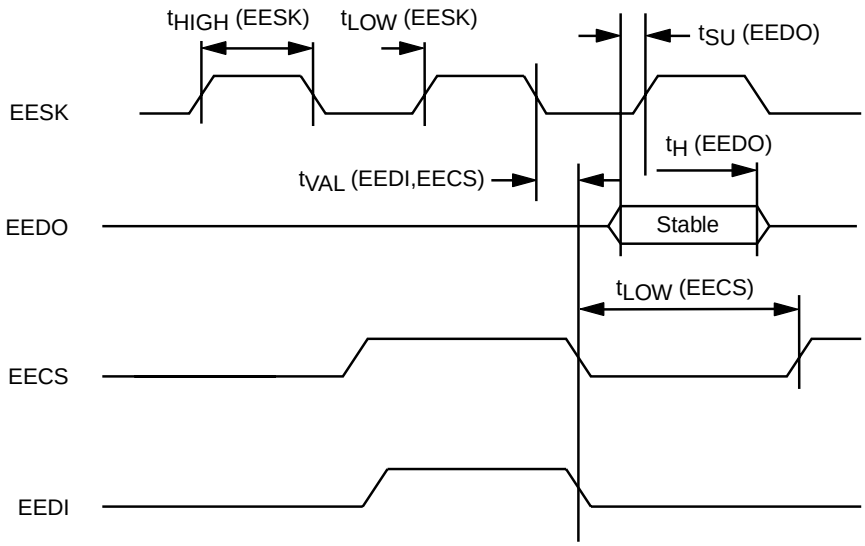


Figure 68. Automatic PREAD EEPROM Timing

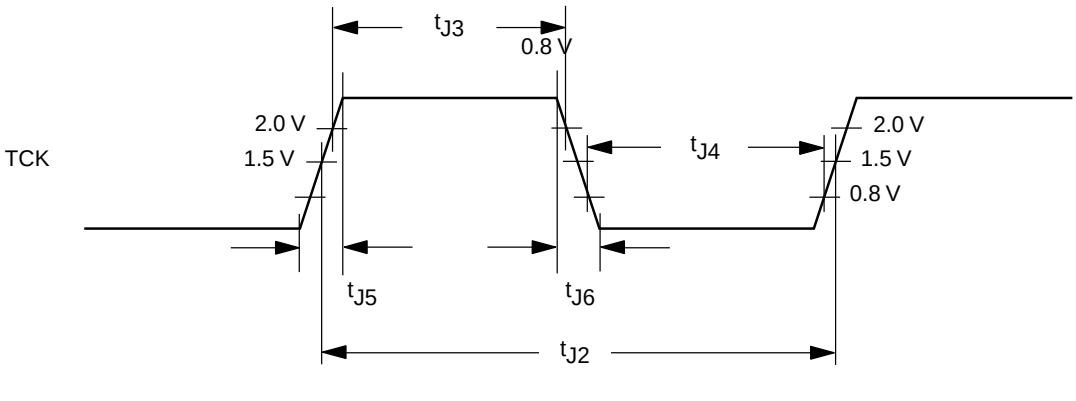
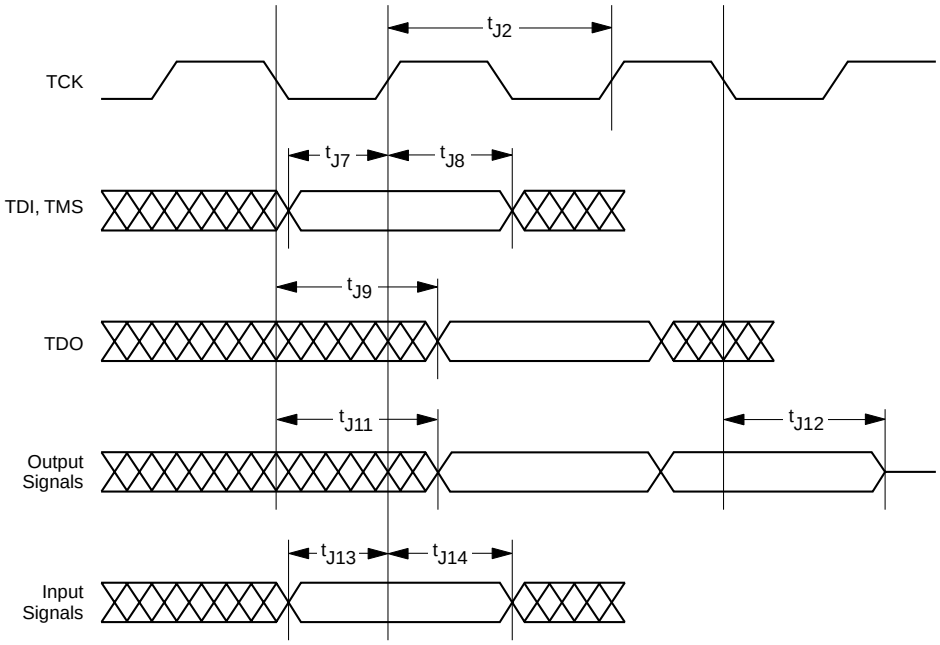


Figure 69. JTAG (IEEE 1149.1) TCK Waveform for 5 V Signaling

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE (CONTINUED)



20550D-73

Figure 70. JTAG (IEEE 1149.1) Test Signal Timing

SWITCHING WAVEFORMS: EXPANSION BUS INTERFACE

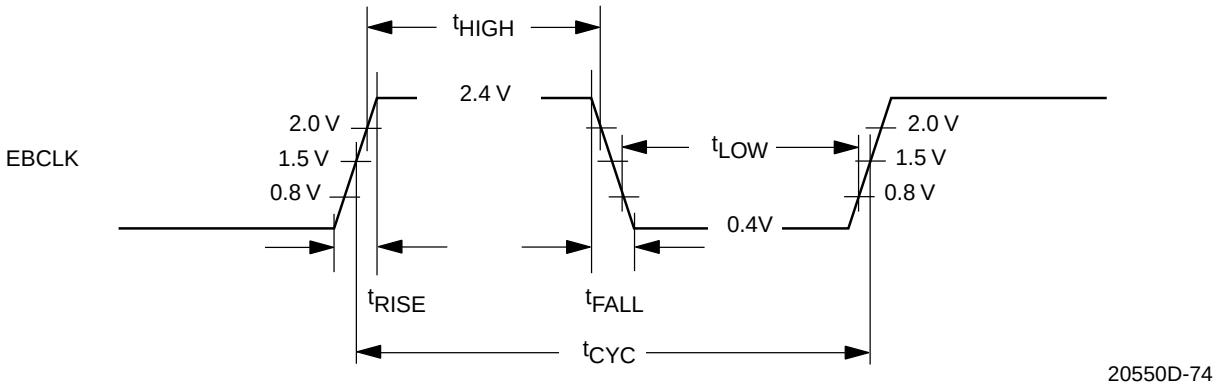


Figure 71. EBCLK Waveform

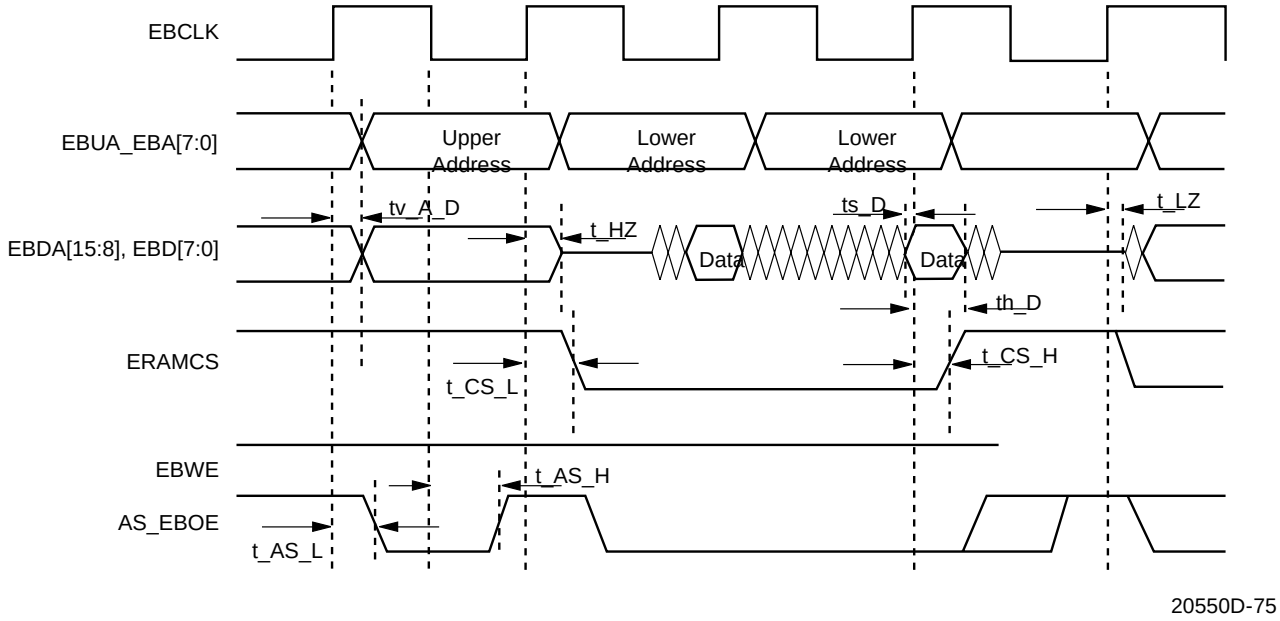


Figure 72. Expansion Bus SRAM Read Timing--Normal SRAM Operation

SWITICHING WAVEFORMS: EXPANSION BUS INTERFACE (CONTINUED)

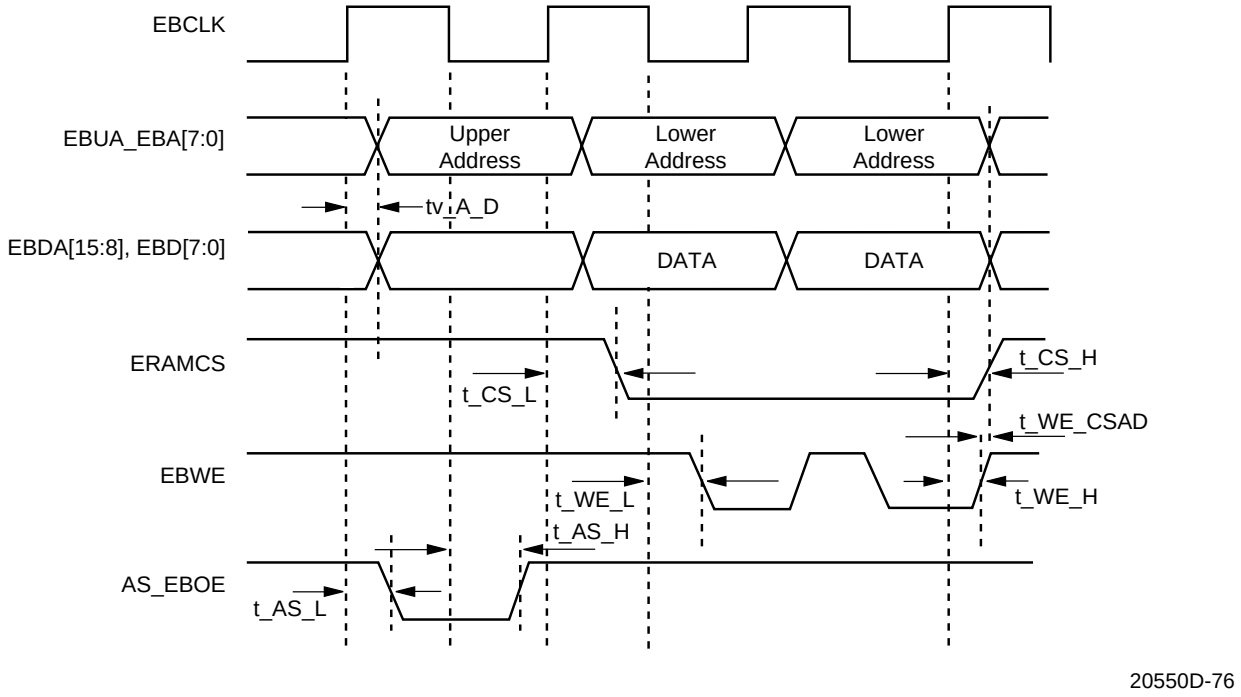
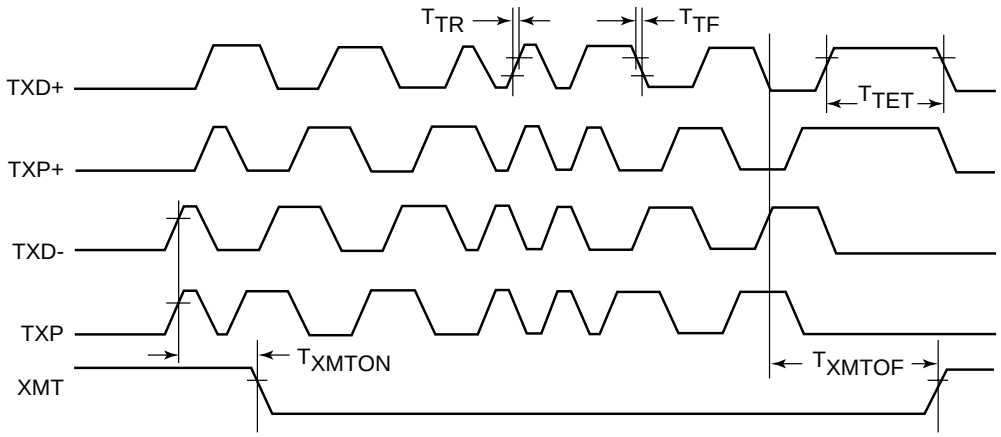


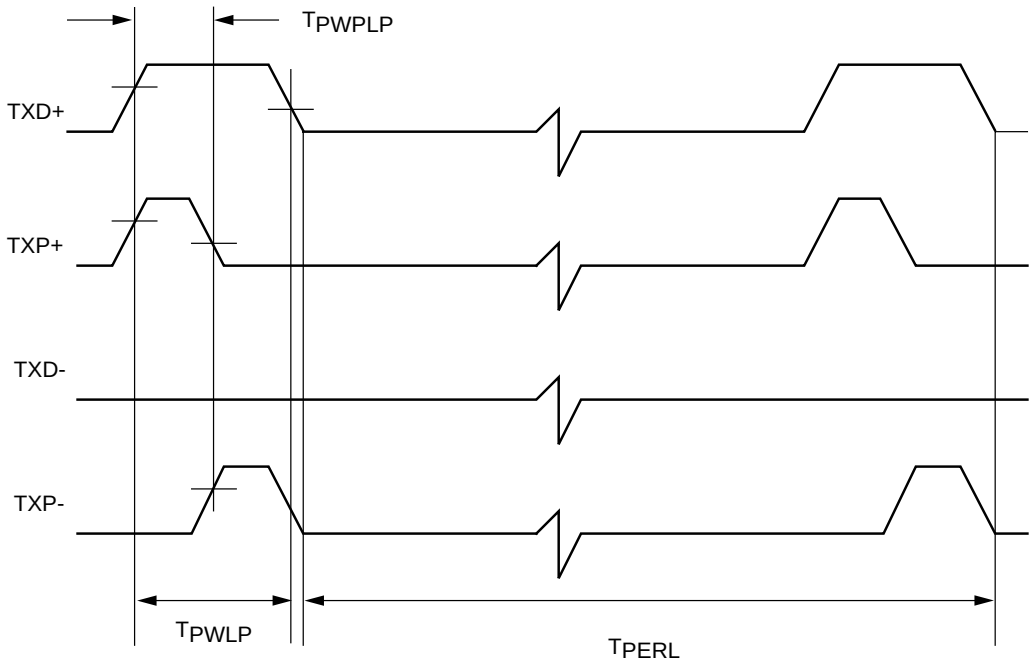
Figure 73. Expansion Bus SRAM Write Timing--Normal SRAM Operation

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



20550D-77

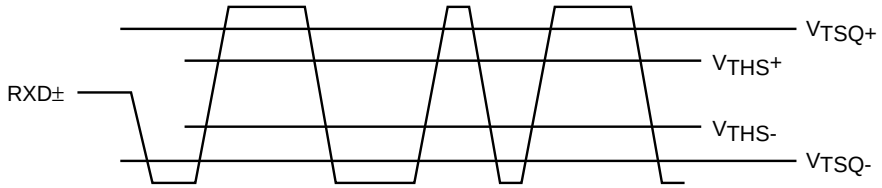
Figure 74. Transmit Timing



20550D-78

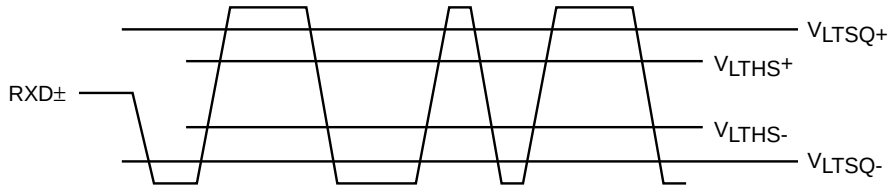
Figure 75. Idle Link Test Pulse

SWITCHING WAVEFORMS: 10BASE-T INTERFACE (CONTINUED)



20550D-79

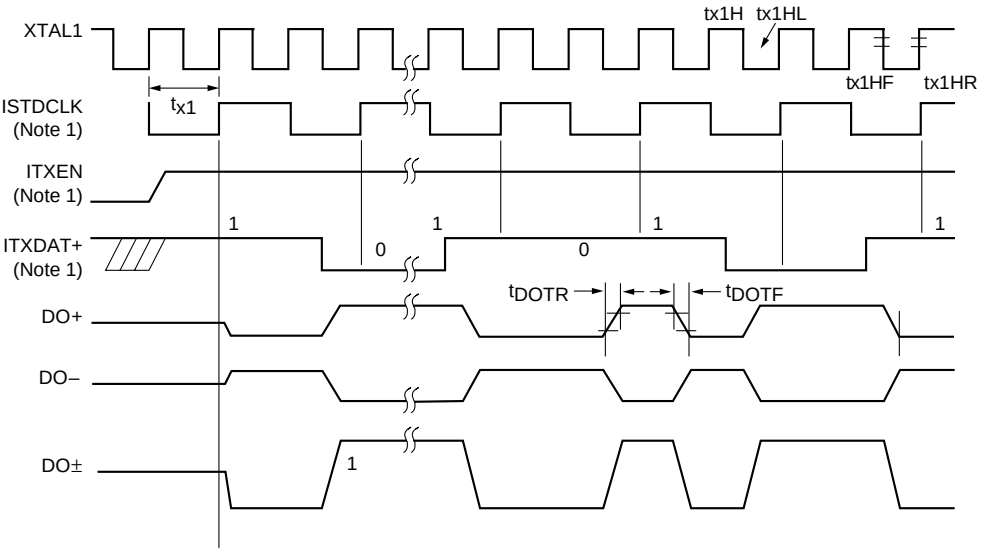
Figure 76. Receive Thresholds (LRT = 0)



20550D-80

Figure 77. Receive Thresholds (LRT = 1)

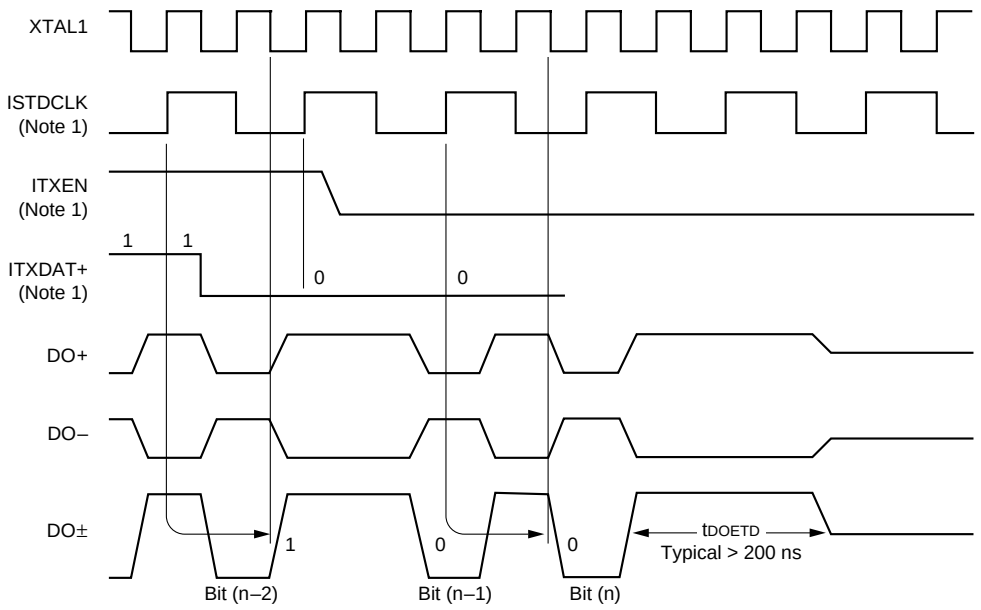
SWITCHING WAVEFORMS: ATTACHMENT UNIT INTERFACE



Note 1:
Internal signal and is shown for clarification only.

20550D-81

Figure 78. Transmit Timing - Start of Frame

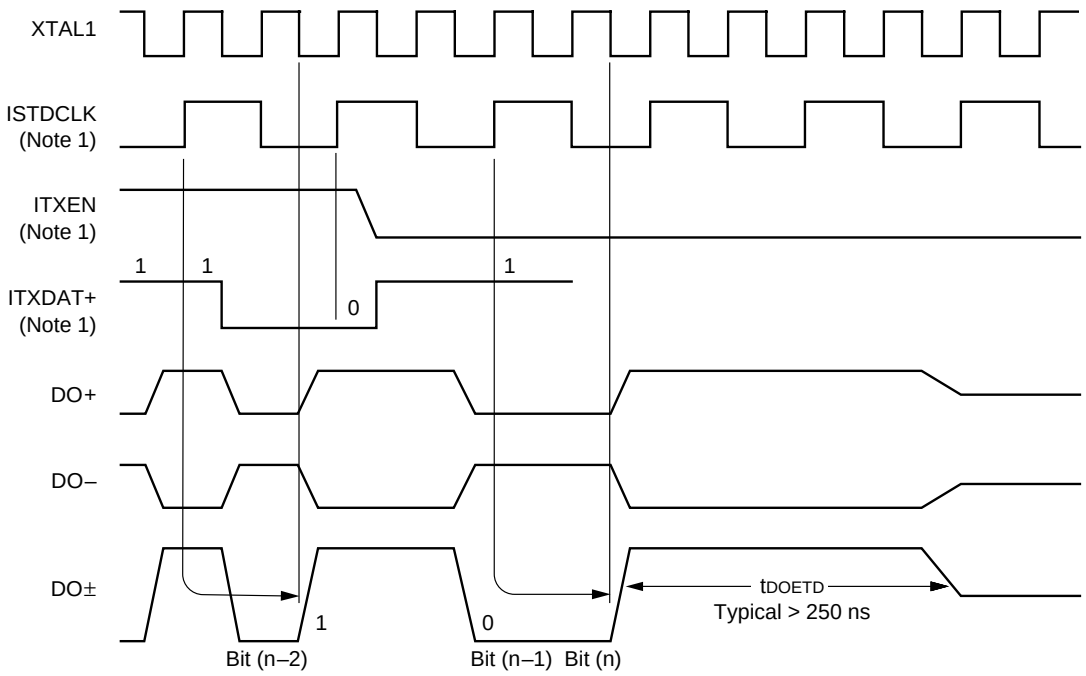


Note 1:
Internal signal and is shown for clarification only.

20550D-82

Figure 79. Transmit Timing - End of Frame (Last Bit = 0)

SWITCHING WAVEFORMS: ATTACHMENT UNIT INTERFACE (CONTINUED)



Note 1:
Internal signal and is shown for clarification only.

20550D-83

Figure 80. Transmit Timing - End of Frame (Last Bit = 1)

SWITCHING WAVEFORMS: ATTACHMENT UNIT INTERFACE (CONTINUED)

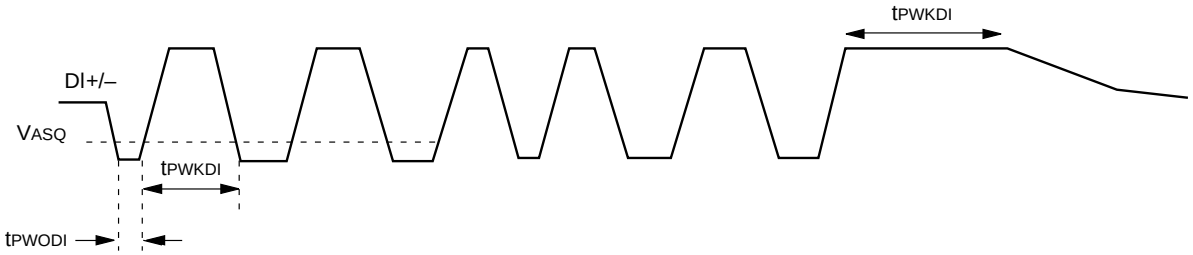


Figure 81. Receive Timing

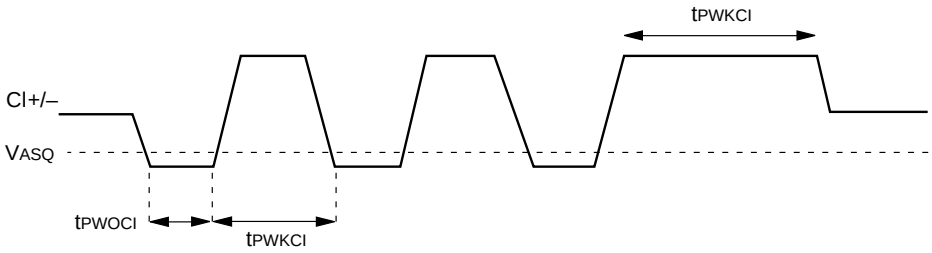


Figure 82. Collision Timing

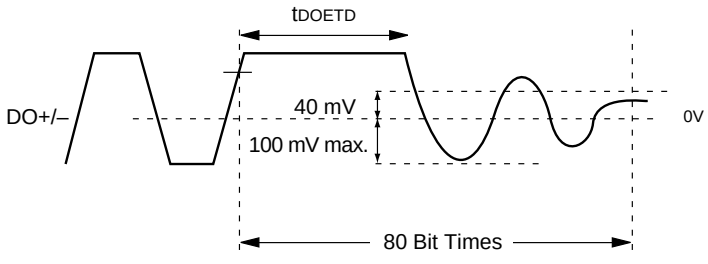
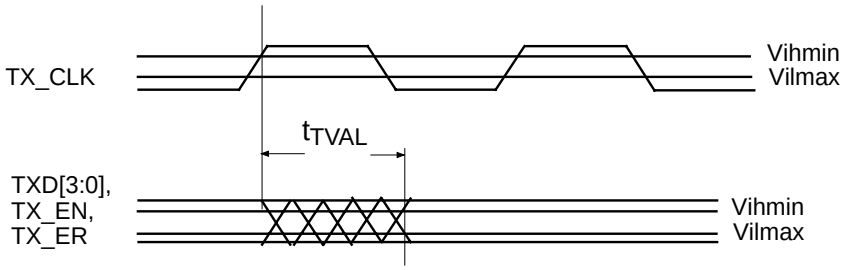


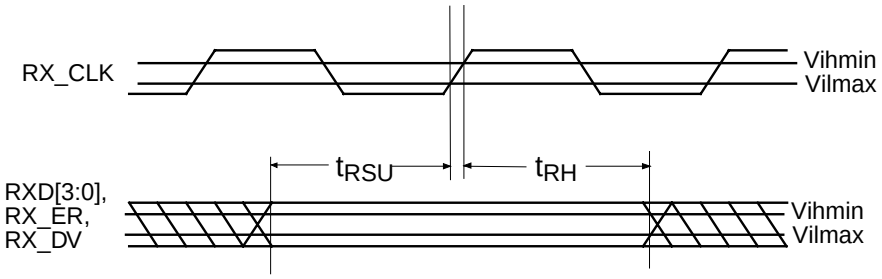
Figure 83. Port DO ETD Waveform

SWITCHING WAVEFORMS: MEDIA INDEPENDENT INTERFACE



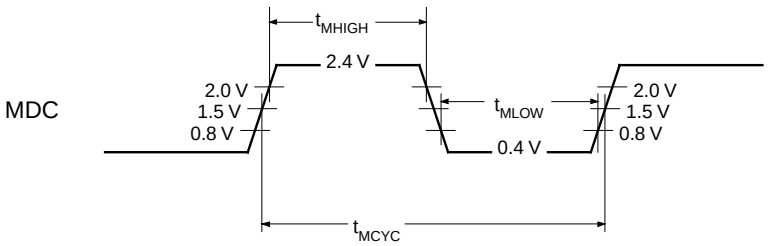
20550D-87

Figure 84. Transmit Timing



20550D-88

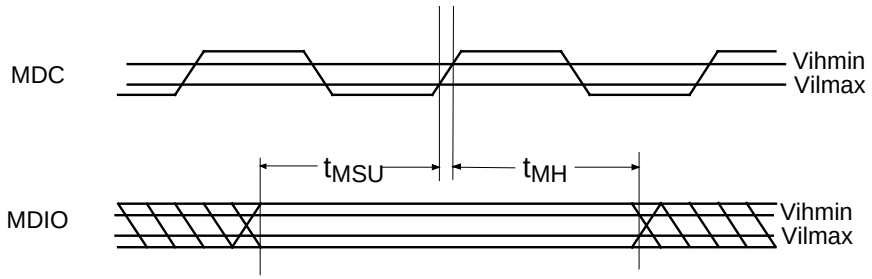
Figure 85. Receive Timing



20550D-89

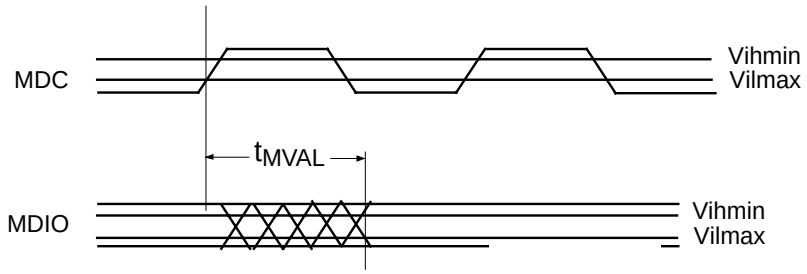
Figure 86. MDC Waveform

SWITCHING WAVEFORMS: MEDIA INDEPENDENT INTERFACE (CONTINUED)



20550D-90

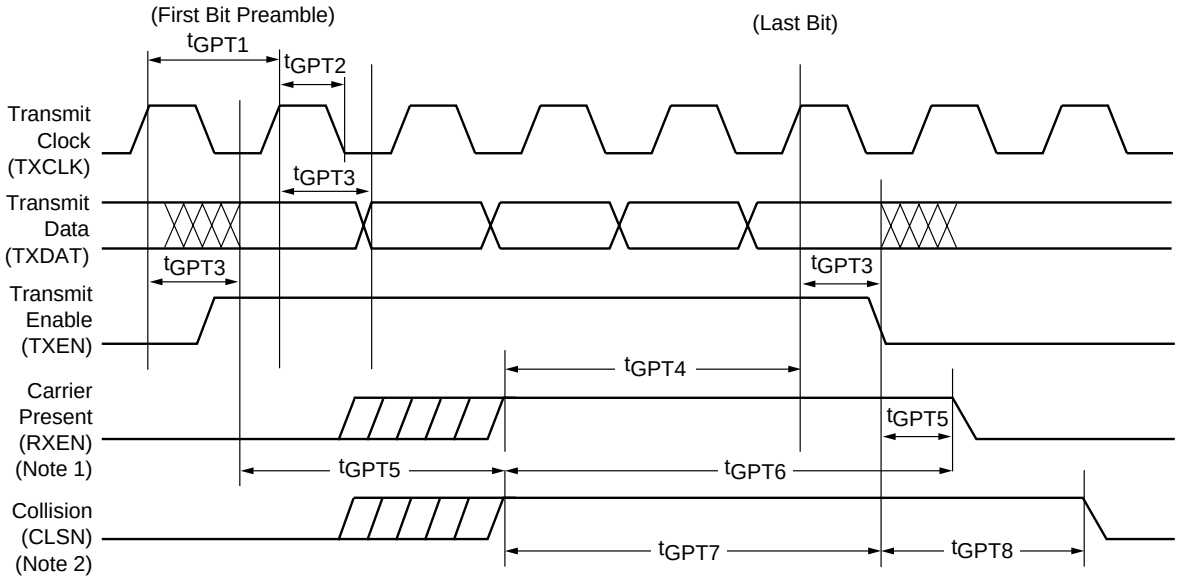
Figure 87. Management Data Setup and Hold Timing



20550D-91

Figure 88. Management Data Output Valid Delay Timing

SWITCHING WAVEFORMS: GENERAL-PURPOSE SERIAL INTERFACE



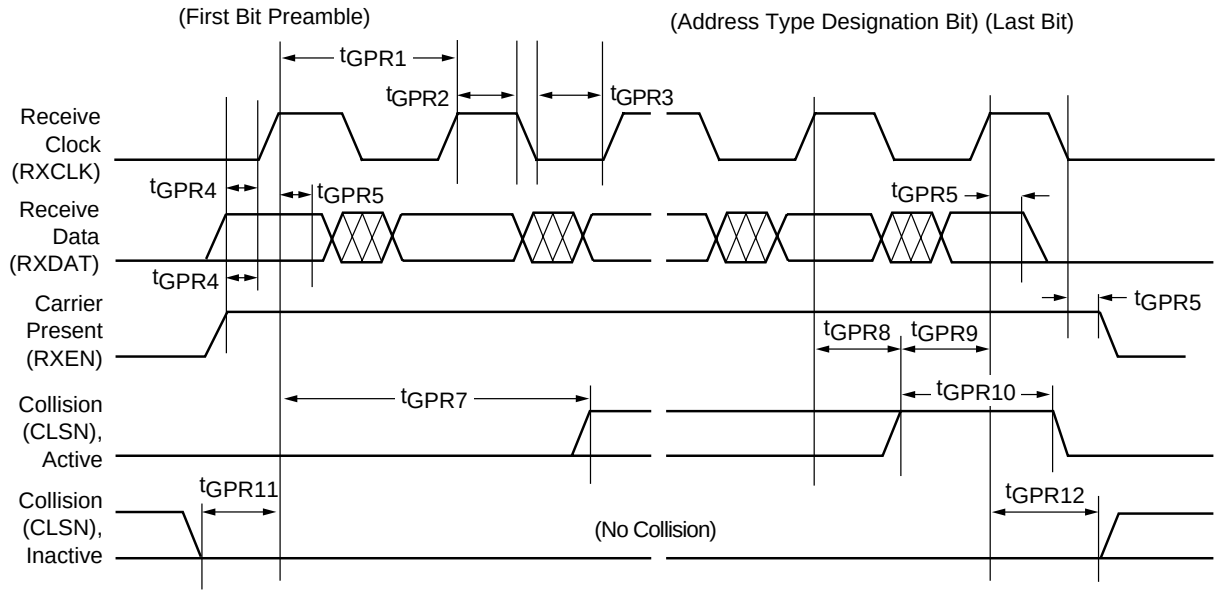
Notes:

1. If RXCRS is not present during transmission, LCAR bit in TMD2 will be set.

2. If CLSN is not present during or shortly after transmission, CERR in CSR0 will be set.

20550D-92

Figure 89. Transmit Timing



20550D-93

Figure 90. Receive Timing

SWITCHING WAVEFORMS: EXTERNAL ADDRESS DETECTION INTERFACE

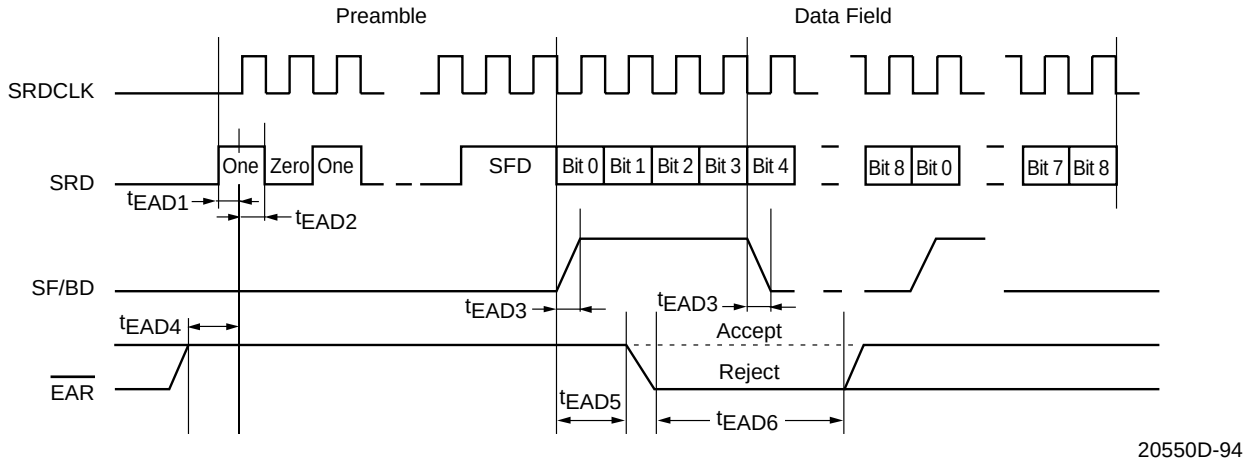


Figure 91. Reject Timing

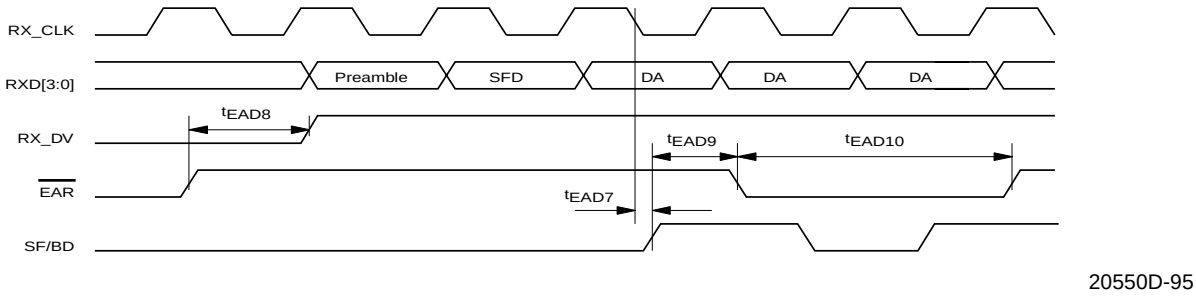


Figure 92. Reject Timing - External PHY MII @ 25 MHz

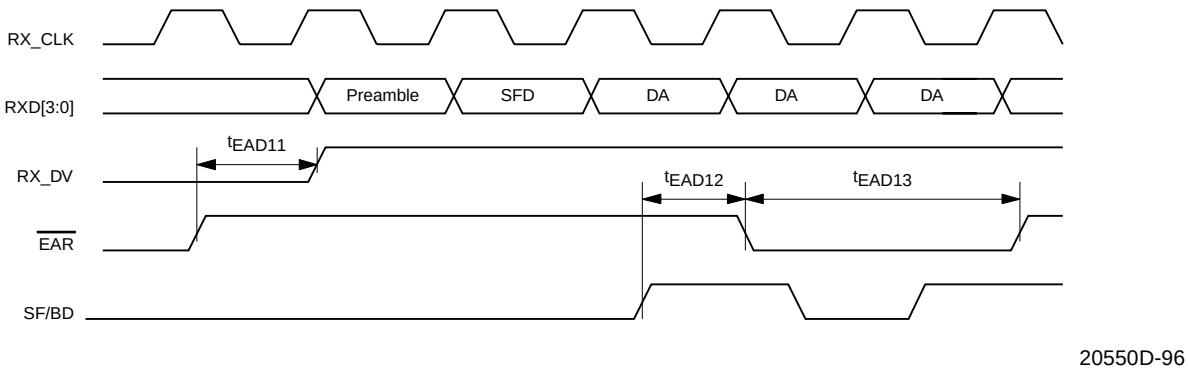


Figure 93. Reject Timing - External PHY MII @ 2.5 MHz

SWITCHING WAVEFORMS: RECEIVE FRAME TAG

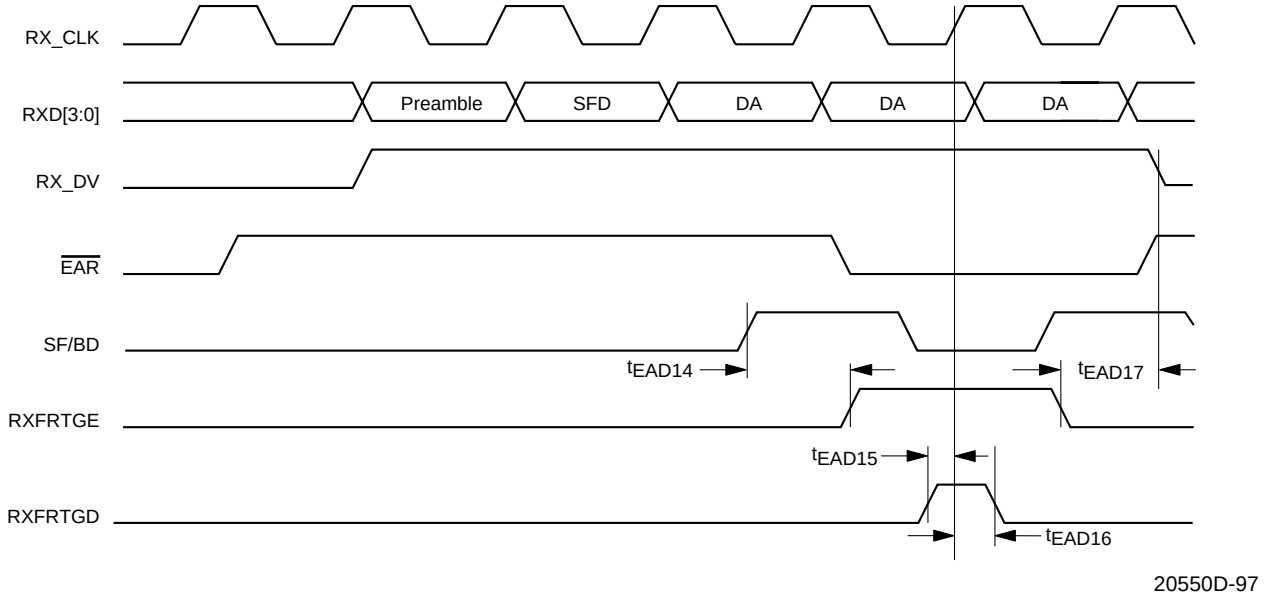
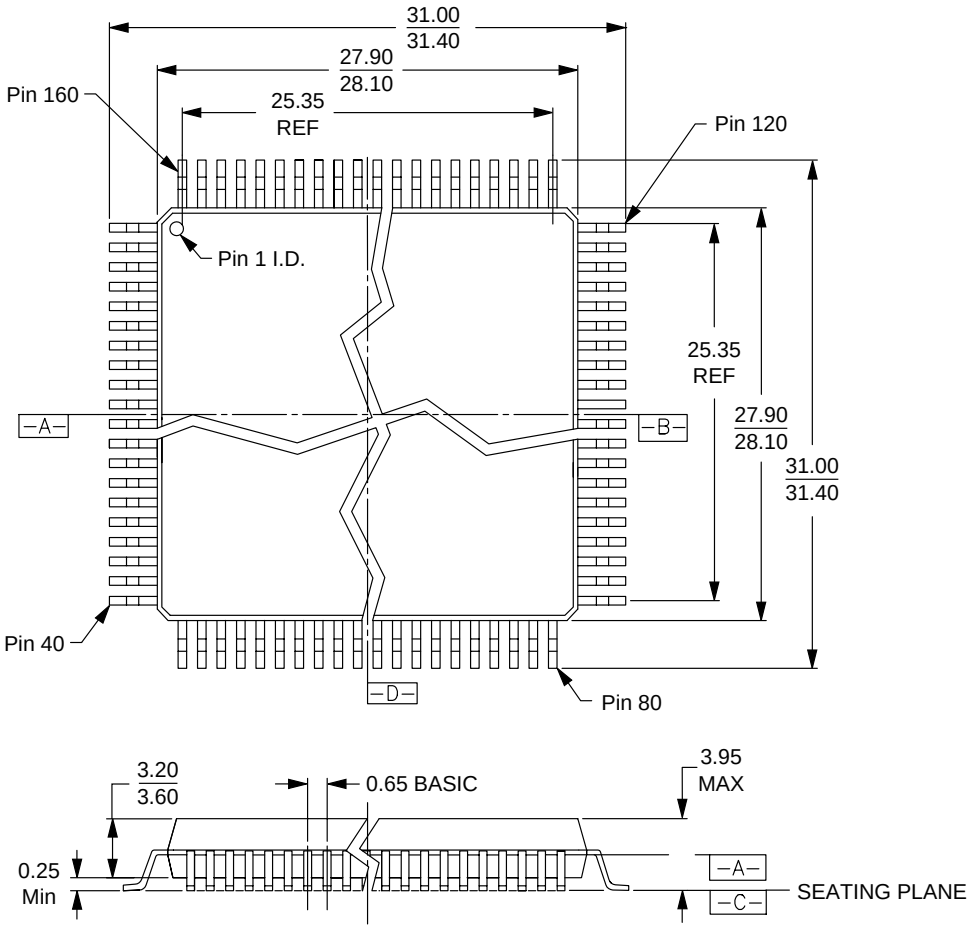


Figure 94. Receive Frame Tag Timing with Media Independent Interface

PHYSICAL DIMENSIONS*
PQR160

Plastic Quad Flat Pack (measured in millimeters)

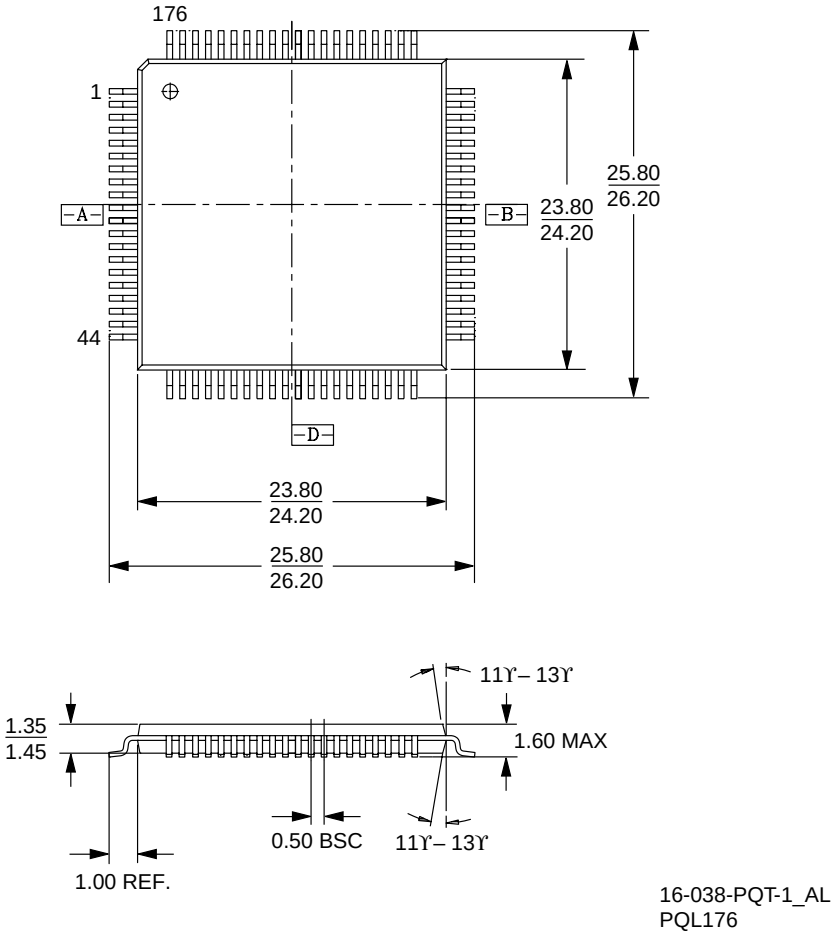


16-038-PQR-1
PQR160
12-22-95 lv

**For reference only. BSC is an ANSI standard for Basic Space Centering.*

PQL176

Thin Quad Flat Pack (measured in millimeters)



16-038-PQT-1_AL
PQL176
5.12.97. lv

APPENDIX A

Am79C971 Compatible Media Interface Modules

AM79C971 COMPATIBLE 10BASE-T FILTERS AND TRANSFORMERS

The table below provides a sample list of Am79C971 compatible 10BASE-T filter and transformer modules available form various vendors. Contact the respective manufacturer for a complete updated component listing.

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	X			
Bel Fuse	0556-2006-00	14-pin SIP	X			
Bel Fuse	0556-2006-01	14-pin SIP			X	
Bel Fuse	0556-6392-00	16-pin 0.5 DIL			X	
Halo Electronics	FD02-101G	16-pin 0.3 DIL	X			
Halo Electronics	FD12-101G	16-pin 0.3 DIL		X		
Halo Electronics	FD22-101G	16-pin 0.3 DIL			X	
PCA Electronics	EPA 1990A	16-pin 0.3 DIL	X			
PCA Electronics	EPA 2013D	16-pin 0.3 DIL		X		
PCA Electronics	EPA 2162	16-pin 0.3 SIP			X	
Pulse Engineering	PE-65421	16-pin 0.3 DIL	X			
Pulse Engineering	PE-65434	16-pin 0.3 SIL			X	
Pulse Engineering	PE-65445	16-pin 0.3 DIL			X	
Pulse Engineering	PE-65467	12-pin 0.5 SMT				X
Valor Electronics	PT3877	16-pin 0.3 DIL	X			
Valor Electronics	PT3877	16-pin 0.3 DIL			X	

***Note**: 100BASE-TX and 100BASE-T4 interfaces are unique to the external PHY. Contact the PHY vendor for the appropriate information.*

Am79C971 Compatible AUI Isolation Transformers

The table below provides a sample list of Am79C971 compatible AUI isolation transformers available from various vendors. Contact the respective manufacturer for a complete updated component listing.

Manufacturer	Part No.	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3 DIL	50 mH
Bel Fuse	S553-0756-AE	16-pin 0.3 SMD	75 µH
Halo Electronics	TD01-0765K	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	PE-64106	16-pin 0.3 DIL	50 µH
Pulse Engineering	PE-65723	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

Am79C971 Compatible DC/DC Converters

The table below provides a sample list of Am79C971 compatible DC/DC converters available from various vendors. Contact the respective manufacturer for a complete updated component listing.

Manufacturer	Part No.	Package	Voltage	Remote On/Off
Halo Electronics	DCUO-0509D	24-pin DIP	5/-9	No
Halo Electronics	DCUO=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.

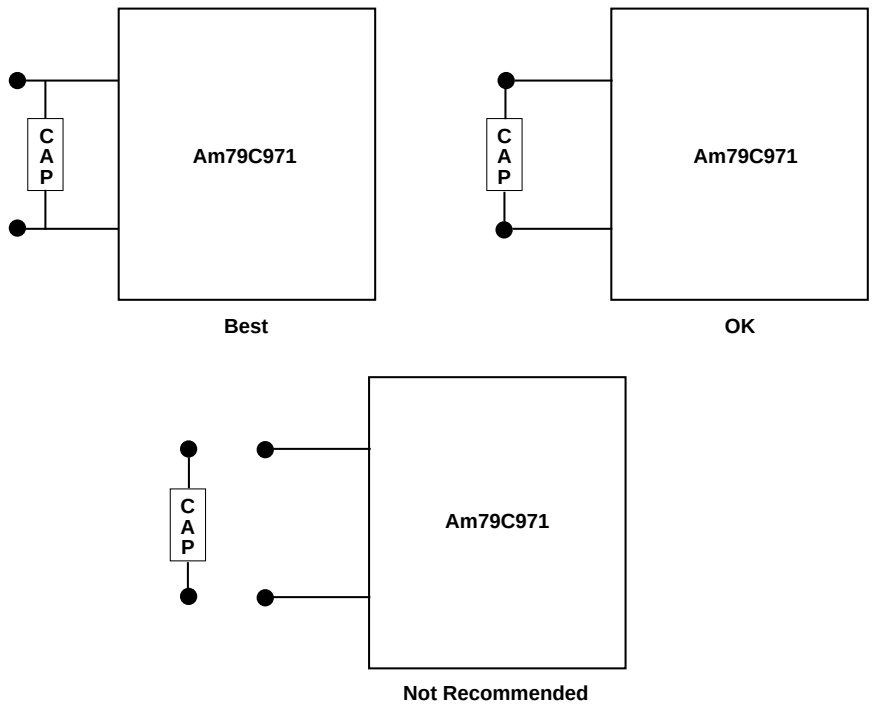
Company	U. S. and Domestic	Asia	Europe
Bel Fuse	Phone: (201) 432-0463 Fax: (201) 432-9542	852-328-5515 852-352-3706	33-1-69410402 33-1-69413320
Halo Electronics	Phone: (415)969-7313 Fax: (415) 367-7158	65-285-1566 65-284-9466	
PCA Electronics (HPC in Hong Kong)	Phone: (818) 892-0761 Fax: (818) 894-5791	852-553-0165 852-352-3706	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	49-89-69233122 49-89-6926542

Recommendation for Power and Ground Decoupling

The mixed analog/digital circuitry in the Am79C971 device makes 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 Am79C971 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 Am79C971 device. The bulk capacitor(s) should be connected directly to the power and ground planes. In addition, **at least eight** high frequency decoupling capacitors (e.g., 0.1 µF multilayer ceramic capacitors of dielectric type XDR) 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 capacitor, 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).



19364D-1

Figure 1. Connection Scheme for Capacitors

The most critical pins in the layout of a Am79C971 design are the four analog power and two analog ground pins, VDD_PLL, AVDD, VSS_PLL and AVSS 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.

AVSS and AVDD

These pins provide the power and ground for the Twisted Pair and AUI drivers. In addition AVSS 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.

VSS_PLL and VDD_PLL

These pins are the **most critical** pins on the Am79C971 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 when internal ports are used.This filter is not needed when MII is used solely.

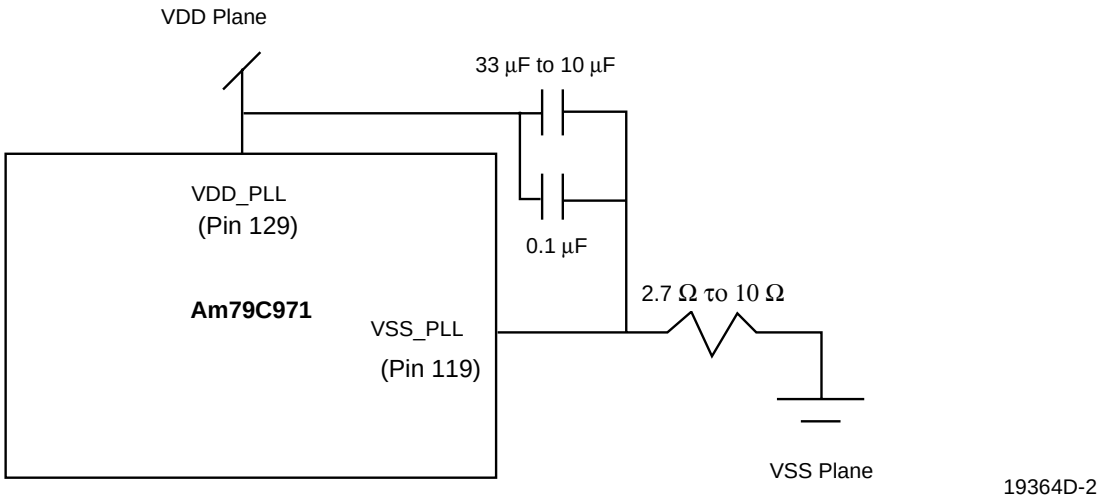


Figure 2. Power and Ground Pin Connections

To determine the value for the resistor and capacitor, the formula is:

$R * C \geq 88$

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 Ω.

Note: The capacitor used should be tantalum not aluminum electrolytic.5.0

R	C
2.7 Ω	33 µF
4.3 Ω	22 µF
6.8 Ω	15 µF
10 Ω	10 µF

VSS_PLL and VDD_PLL/AVDD

These pins provide power and ground for the AUI and twisted pair receive circuitry. In addition, as mentioned earlier, VSS_PLL and VDD_PLL 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.

AVDD

AVDD 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 power and 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 Am79C971 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	TXPOLLINT
CSR49	RXPOLLINT
CSR76	RCVRL
CSR78	XMTRL

Notes:

**The INIT bit must not be set or the initialization block will be accessed instead.*

***Needed only if SSIZE32 =0.*

APPENDIX D

Look-Ahead Packet Processing (LAPP) Concept

INTRODUCTION

A driver for the Am79C971 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 controller. 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 between last byte of a receive frame arriving at the client's Ethernet controller and the client's transmission of the first byte of the next outgoing frame will be separated by:

1. The time that it takes the client's CPU's 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 controller's buffer space into the application's 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, and
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 percent.

An important thing to note is that the Am79C971 controller's data transfers to its buffer space are such that the system bus is needed by the Am79C971 controller for approximately 4 percent of the time. This leaves 96 percent of the system bus bandwidth for the CPU to perform some of the interframe 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 three steps and part of the fourth 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 three steps are performed before the end of the network receive operation. A much more significant performance increase could be realized if the Am79C971 controller could place the frame data directly into the application's 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 Am79C971 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 is 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 Am79C971 controller's 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 LAPP Flow

This section gives a suggested outline for a driver that utilizes the LAPP feature of the Am79C971 controller.

Note: *The labels in the following text are used as references in the timeline diagram that follows (Figure D1).*

Setup

The driver should set up descriptors in groups of three, 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 Am79C971 controller to generate an INTERRUPT when STP has been written to a receive descriptor by the Am79C971 controller.

Flow

The Am79C971 controller polls the current receive descriptor at some point in time before a message arrives. The Am79C971 controller determines that this receive buffer is OWNed by the Am79C971 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 Am79C971 controller to begin frame data DMA operations to the first buffer.

C0 When the 64th byte of the message arrives, the Am79C971 controller performs a lookahead operation to the next receive descriptor. This descriptor should be owned by the Am79C971 controller.

C1 The Am79C971 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 Am79C971 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 Am79C971 controller to the CPU, the Am79C971 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 Am79C971 controller's driver to run.

C4 During the CPU interrupt-generated task switching, the Am79C971 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 Am79C971 controller, existing AMD Ethernet controllers 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 this point that it discovers that it does not yet own descriptor number 3.*

S2 The first task of the drivers interrupt service routing is to collect the header information from the Am79C971 controller's 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 Am79C971 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 Am79C971 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 Am79C971 controller.

C5 Interleaved with S2, S3, and S4 driver activity, the Am79C971 controller will write frame data to buffer number 2.

S4 The driver will next proceed to copy the contents of the Am79C971 controller's 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 Am79C971 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 Am79C971 controller will make a last ditch lookahead to the final (third) descriptor. This time the ownership will be TRUE (i.e., the descriptor belongs tot he controller), because the driver wrote the application pointer into this descriptor and then changed the ownership to give the descriptor to the Am79C971 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 Am79C971 controller will write the sta-

tus and change the ownership bit of descriptor number 2.

S6 After the ownership of descriptor number 2 has been changed by the Am79C971 controller, the next driver poll of the second 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 Am79C971 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.

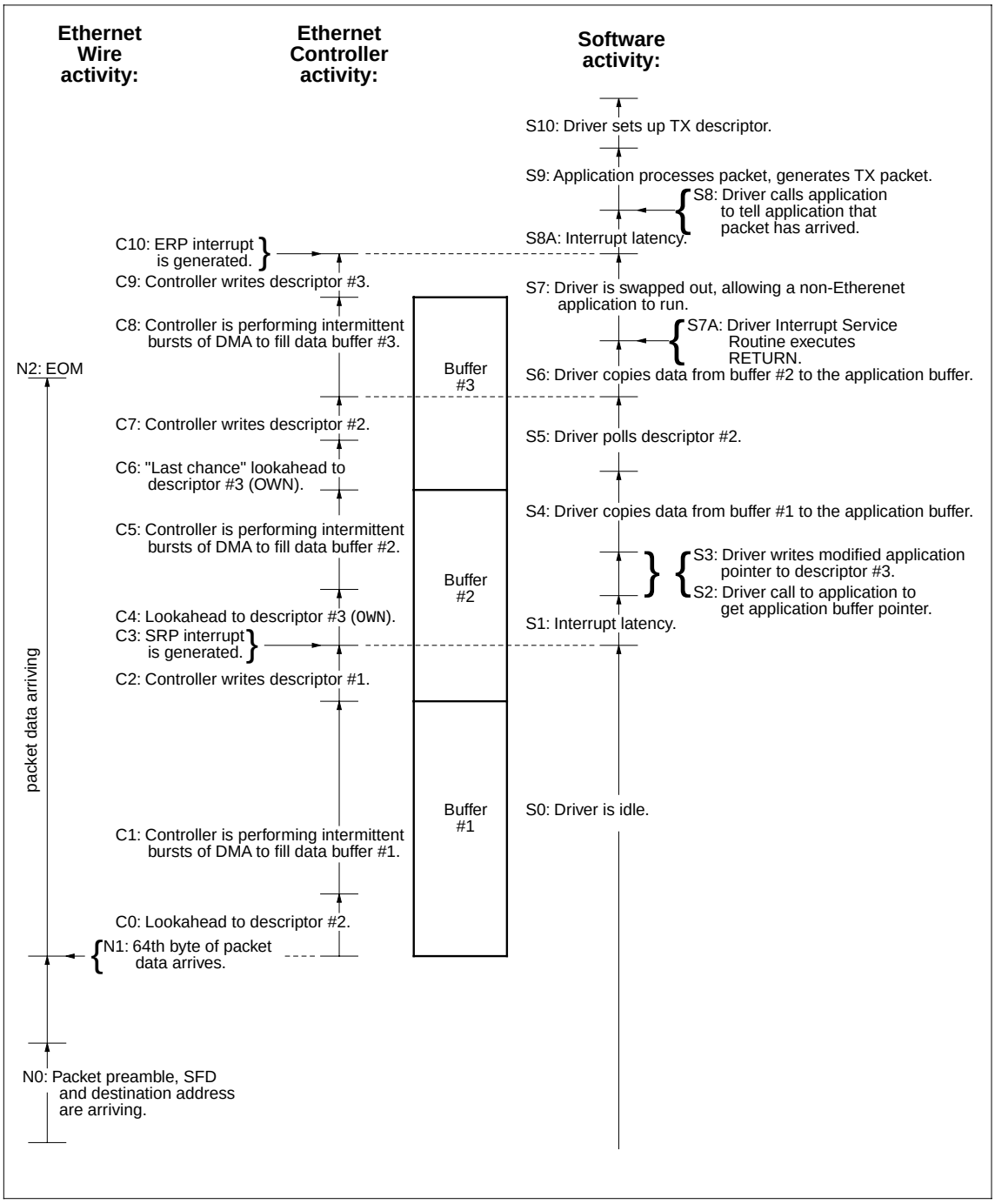
S7 When the driver completes the copy of buffer number 2 data to the application buffer space, it begins polling descriptor number 3.

C9 When the Am79C971 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 Am79C971 controller.



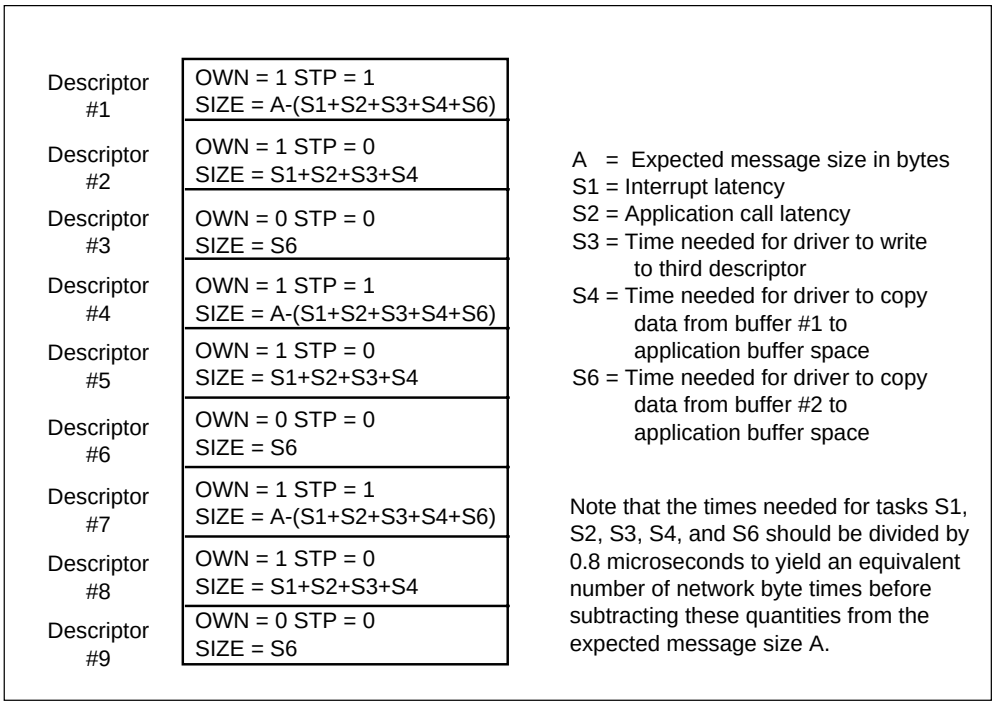
19364D-D1

Figure D1. LAPP Timeline

LAPP Software Requirements

Software needs to set up a receive ring with descriptors formed into groups of three. 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 drive to copy data from buffer number 2 to the application buffer space. Note that the time needed for the copies performed by the driver depends upon the sizes of the second and third 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.

Figure D-2 illustrates this setup for a receive ring size of 9.



19364D-D2

Figure D2. LAPP 3 Buffer Grouping

LAPP Rules for Parsing 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 an 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 an 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.

- **Example 1**: 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 Arrives			Comments (After Frame Arrival)
	OWN	STP	ENP[a]	OWN	STP	ENP[b]	
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	Net yet used

a. & b. ENP or ERR.

- **Example 2**: 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 Arrives			Comments (After Frame Arrival)
	OWN	STP	ENP[a]	OWN	STP	ENP[b]	
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	?*	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	Net yet used

a. & b. ENP or ERR.

Note that the Am79C971 controller might write a ZERO to ENP location in the third descriptor. Here are the two possibilities:

1. If the controller finishes the data transfers into buffer number 2 after the driver writes the application 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 point into the third descriptor, then the controller will complete the frame in buffer number 2 and then skip the then unowned third buffer. In this case, the Am79C971 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.

■ **Example 3**: 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 example 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 Am79C971 controller has completed its poll of the next descriptors. This means that for almost all occurrences of this case, the Am79C971 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 Am79C971 controller will not have had an opportunity to modify it.*

****Note that even though the Am79C971 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 Arrives			Comments (After Frame Arrival)
	OWN	STP	ENP[a]	OWN	STP	ENP[b]	
1	1	1	x	0	1	0	Bytes 1-800
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	Net yet used

a. & b.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 the frame sent to application from driver (frame latency). These objectives are aimed at increasing throughput on the network while decreasing CPU utilization.

Note: *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 Am79C971 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: 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 perfectly 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 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 number 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: Two-Interrupt Method

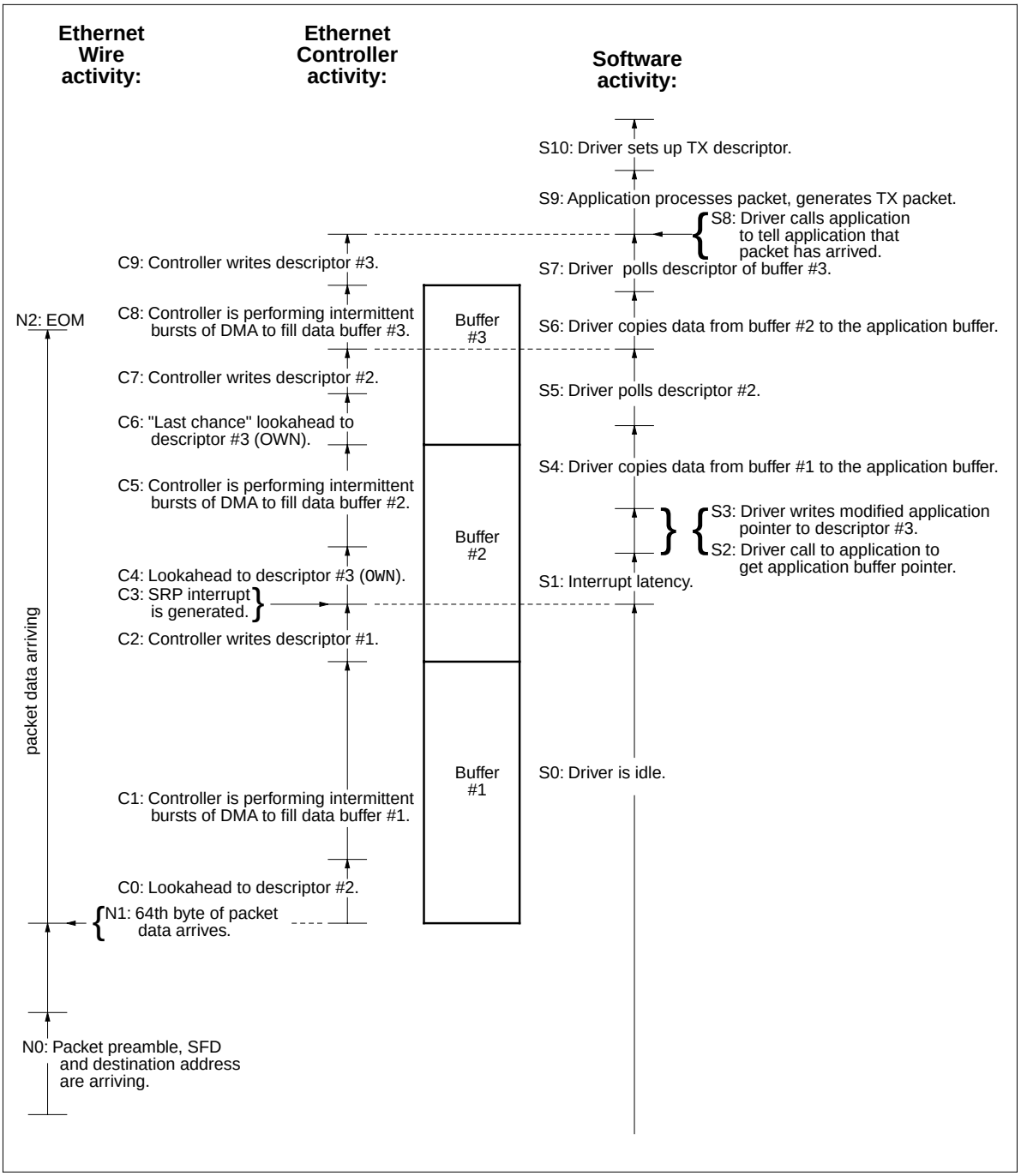
An alternative to the above suggested flow is to use two interrupts, one at the start of the receive frame and the other at the end of the receive frame, instead of just looking for the SRP interrupt as described above. This alternative attempts to reduce the amount of time that the software wastes while 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. See Figure D3.

Note: *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.*

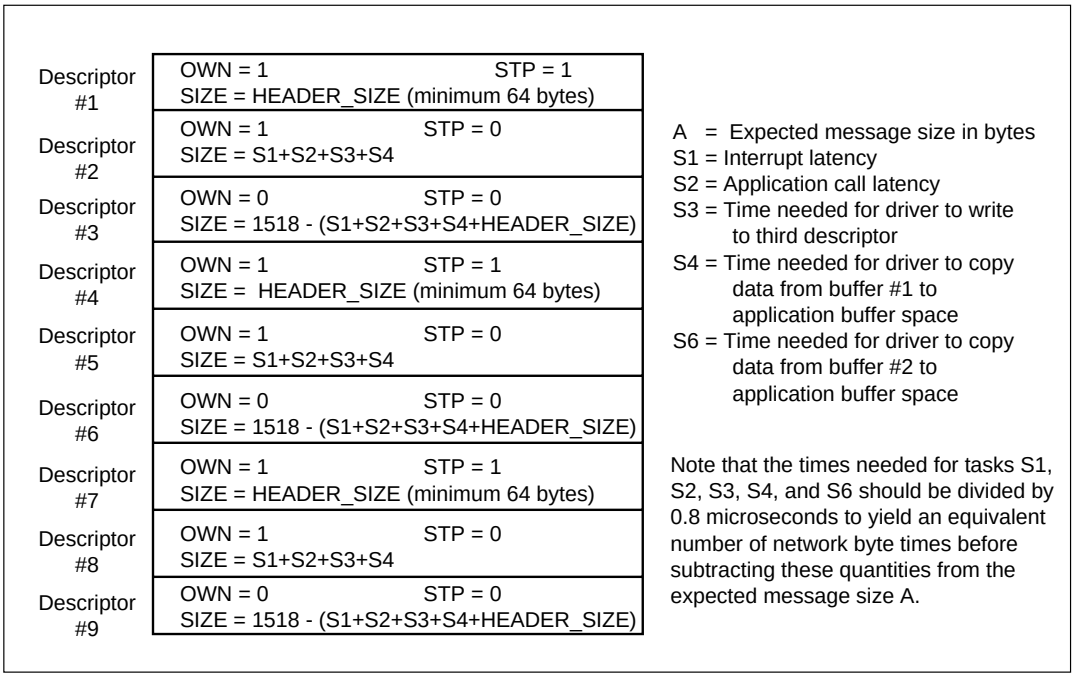
Figure D4 shows the buffer sizing for the two-interrupt method. Note that the second buffer size will be about the same for each method.

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.



19364D-D3

Figure D3. LAPP Timeline for Two-Interrupt Method



19364D-D4

Figure D4. LAPP 3 Buffer Grouping for Two-interrupt Method

Note:

This document demonstrates a 3-buffer per packet implementation of LAPP. For PCnet-FAST application with external SRAMs enabled, a 2-buffer per packet implementation can be used.

Auto-Negotiation Registers

The following registers exist in the Auto-Negotiation block and are written or read by using registers MII Address (BCR33) and MII Data Port (BCR34). These registers are indirect read and write registers with the register offset specified by the REGAD (BCR33, bits 3-0) portion of the MII Address Register (BCR33). The PHYAD (BCR33, bits 3-0) portion specifies the specific PHY that you wish to communicate with. The internal PHY on the Am79c971 controller has a PHYAD address of 1fh. The default external PHY has a PHYAD address of 00h. The following is a table of these indirect registers and defaulted values for the internal registers.

Register 0: MII Control Register

Bit	Name	Default	Description
15	PRPHY	Reset.	Defaults to ZERO after power-up and after hardware resets (H_RESET). This is normally only used to bring up the external PHY into a known state. Not used for the internal PHY. Use the normal Am79C971 reset mechanisms.
14	LB	NA (0)	Loop Back. Defaults to ZERO after power-up and after hardware resets (H_RESET). Loopback is supported only in an external PHY. See the “MII Configuration” section for more details..
13	SS	NA (0)	Speed Selection. Defaults to ZERO after power-up and after hardware resets (H_RESET). This is ZERO the entire time the internal PHY is active. The external PHY can be forced into 100Mbps or 10Mbps mode by manipulation of this bit. This bit is used only when the Auto-Negotiation Enable bit is ZERO.
12	ANE	Variable*	Auto-Negotiation Enable. Defaults to ONE after power-up and after hardware resets (H_RESET). This bit enables the internal or external PHY’s Auto-Negotiation capabilities. When this bit is ZERO the internal/external PHY’s must be forced into the correct state through software intervention.
11	PD	NA (0)	Power Down. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit only applies to the external PHY. Use the SLEEP# or other power saving modes in the Am79C971 Controller.

** See text on Auto Negotiation.*

10	IS	NA (0)	Isolate. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit only applies to the external PHY. This bit is used when multiple PHY's can be attached to the Am79C971 MII.
9	RAN	0	Restart Auto Negotiation. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit is used when the software needs to reset the Auto-Negotiation process. For use with an external PHY without software intervention.
8	DM	0	Duplex Mode. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit sets the full-duplex feature for the internal or external PHY's. Only valid when the Auto-Negotiation Enable bit is not set.
7	CT	0	Collision Test. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit is only useful for testing the COL pin on the MII and is not supported without software intervention.
0-6	RES	0	Reserved locations. Defaults to all ZEROs after power-up and after hardware resets (H_RESET).

Register 1: MII Status Register

Bit	Name	Default	Description
15	100T4	0	100Base-T4. Defaults to ZERO after power-up and after hardware resets (H_RESET). Indicates that the PHY is 100BASE-T4 capable. Not used on the internal PHY.
14	100XFD	0	100Base-X Full Duplex. Defaults to ZERO after power-up and after hardware resets (H_RESET). Indicates that the PHY is 100BASE-TX Full-Duplex capable.
13	100XHD	0	100Base-X Half Duplex. Defaults to ZERO after power-up and after hardware resets (H_RESET). Indicates that the PHY is 100BASE-TX Half-Duplex capable. Not used on the internal PHY
12	10FD	1	10Base Full Duplex. Defaults to ONE after power-up and after hardware resets (H_RESET). Indicates that the PHY is 10BASE-T Full-Duplex capable.
11	10HD	1	10Base Half Duplex. Defaults to ONE after power-up and after hardware resets (H_RESET). Indicates that the PHY is 10BASE-T Half-Duplex capable.
10-7	RES	0	Reserved locations. These locations should be read and written as ZEROs.

6	MFPS	0	Management Frame Preamble Suppression. Defaults to ZERO after power-up and after hardware resets (H_RESET). This is used by the Am79C971 controller to reduce the time delay overhead with processing the MII Management Frames. Strips the 32 bit preamble off of each frame sent.
5	ANC	0	Auto-Negotiation Complete. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit indicates that the Auto-Negotiation process has completed and that the contents of registers 4-8 are valid.
4	RF	0	Remote Fault. Defaults to ZERO after power-up and after hardware resets (H_RESET). This is a indication that the remote node has detected a fault with your system.
3	ANA	1	Auto-Negotiation Ability. Defaults to ONE after power-up and after hardware resets (H_RESET). This bit indicates that PHY's ability to do Auto-Negotiation. This bit is defaulted ONE on the Am79C971 controller.
2	LS0	0	Link Status. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit indicates the current status of the network link.
1	JD	0	Jabber Detect. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit indicates that a jabber event occurred on the network.
0	EC	1	Extended Capability. Defaults to ONE after power-up and after hardware resets (H_RESET). This bit indicates that the PHY has Auto-Negotiation capability. If this bit is not set then the line must be forced.

Register 2: PHY Identifier

Bit	Name	Default	Description
15-0	OUI	0000	Organizationally Unique Identifier. Administered by the IEEE.

Register 3: PHY Identifier

Bit	Name	Default	Description
15-10	OUI	68	Organizationally Unique Identifier. Administered by the IEEE.
9-4	MMN	01	Manufacturers Model Number. The Am79C971 Controller will have a default model number of 1.
3-0	RN	1	Revision Number. Initially ZERO.

Register 4: Auto Negotiation Advertisement Ability Register

Bit	Name		Description
15	NP	NA (0)	Next Page. Defaults 0 to ZERO after power-up and after hardware resets (H_RESET). This bit indicates the ability of the PHY to send and receive message pages after the initial base code word has been sent. The Am79C971 controller does not support Next Pages currently.
14	RES	0	Read as ZERO always.
13	RF	0	Remote Fault. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit indicates that we have detected a remote fault with the other node.
12-5	TAF	03	Technology Ability Field. The Am79C971 controller will only support 10BaseT or 10BaseT full duplex with the internal PHY device. When using the internal PHY the TAF will be set to 00000011b always.
4-0	SF	01	Selector Field. The Am79C971 controller will only currently support IEEE 802.3 messages. When using the internal PHY the SF will be set to 00001b always.

Register 5: Auto Negotiation Link-Partner Ability Register

Bit	Name		Description
These fields are identical to the previous and are valid for the node in which we are currently engaged.			
15	NP	0	Next Page. Defaults to ZERO after power-up and after hardware resets (H_RESET).
14	ACK	0	Acknowledge. Defaults to ZERO after power-up and after hardware resets (H_RESET).
13	RF	0	Remote Fault. Defaults to ZERO after power-up and after hardware resets (H_RESET)
12-5	TAF	03	Technology Ability Field. Defaults to all ZEROs after power-up and after hardware resets (H_RESET)
4-0	SF	01	Selector Field. Defaults to all ZEROs after power-up and after hardware resets (H_RESET)

Register 6: Auto Negotiation Expansion Register

Bit	Name		Description
This is the catch all register of odds and ends that did not fit in other registers.			
15-5	RES	0	Reserved. Read and written as ZERO's.
4	PDF	0	Parallel Detection Fault. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit indicates that more than one PHY has responded to a valid link or none have responded to a valid link. Either way

			we need to restart the process.
3	LPNP	0	Link Partner Next Page Able. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit gets set if the link partner can do the Next Page function.
2	NPA	0	Next Page Able. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit indicates that the internal/external PHY attached to the Am79C971 controller can do Next Pages. The Am79C971 controller will not support next pages without software support.
1	PR	0	Page Received. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit indicates that a valid page has been seen by the Am79C971 controller's PHY or the external PHY attached to the Am79C971 controller.
0	LPNA	0	Link Partner Auto-Negotiation Able. Defaults to ZERO after power-up and after hardware resets (H_RESET). This bit indicates whether or not the link partner can auto-negotiate.

Register 7: Auto Negotiation Next Page Transmit Register

Bit	Name		Description
This resister is only used during next pages. This is the word that will be sent by the Am79C971 controller in response to a next page.			
15	NP	0	Next Page. Defaults to ZERO after power-up and after hardware resets (H_RESET).
14	ACK	0	Acknowledge. Defaults to ZERO after power-up and after hardware resets (H_RESET).
13	MP	0	Message Page. Defaults to ZERO after power-up and after hardware resets (H_RESET).
12	ACK2	0	Acknowledge 2. Defaults to ZERO after power-up and after hardware resets (H_RESET).
11	TOG	0	Toggle. Defaults to ZERO after power-up and after hardware resets (H_RESET)
10-0	MUCF	0000	Message/Unformulated Code Field. Defaults to all ZEROs after power-up and after hardware resets (H_RESET).

Am79C971A PCnet-FAST 10/100 Mbps PCI Ethernet Controller REV A.6 ERRATA

REV A.6 STATUS

Revision A.6 silicon is the current full production silicon. Rev. A.6 has fixed rev. A.5 errata #13, #17 and #18. With revision A.6, the device part number has also been revised to Am79C971A in order to distinguish it from revision A.5 (Am79C971).

REV A.6 ERRATA SUMMARY

The device has seventeen errata to date. The system-level impact of these errata on customers is minimal. All information below should be used in conjunction with the Am79C971 PCnet-FAST preliminary data sheet, dated March 1999 (PID #20550D). This datasheet applies to all revisions of the PCnet-FAST device. These errata do not affect operation with PCnet software drivers.

Important Note: In 100 Mb mode, the Am79C971A requires the use of SRAM. Additionally, the SRAM clock (EBCLK) should be driven with a 33 MHz clock, or if pulled up, the PCI clock should be 33MHz.

PCNET-FAST REV. A.6 ERRATA

The Symptom section gives an external description of the problem. The Implication section explains how the device behaves and its impact on the system. The WORK AROUND section describes a work around for the problem. The status section indicates when and how the problem will be fixed.

1) **Symptom:** The default Inter Packet Gap (IPG) value of 60H in CSR125, bits 15-8, results in an actual IPG of 100 bit-times versus the expected 96 bit-times.

Implication: There is a 4 bit-time offset between the IPG value in CSR125 and the actual IPG at the AUI, 10BASE-T, and MII ports of the PCnet-FAST device.

Workaround: Write the value 5CH into the IPG field of CSR125 to ensure an actual, minimum IPG of 96 bit-times.

Status: *No current plan to fix this erratum*.

2) **Symptom:** The Interrupt Request pin (INTA#, pin 142), and some of the analog pins (RXD-, pin 110, RXD+, pin 111, TXP-, pin 113, TXD-, pin 114, XTAL2, pin 120, DO-, pin 122, DI-, pin 125, DI+, pin 126, CI-, pin 127, CI+, pin 128) are not accessible through the IEEE 1149.1 (JTAG) test interface.

Implication: Those pins are not included in the boundary-scan chain within the device.

Workaround: None.

Status: *No current plan to fix this erratum*.

3) **Symptom:** At high temperature and low Vcc (85 °C, 4.75V) with a minimum PCI clock low time of 12 ns, the Tval timing for some PCI interface signals exceeds the PCI spec (11 ns max.) by up to 3 ns. The measured value is less than 14 ns maximum on the production tester for this specific corner case for FRAME# de-assertion, STOP# assertion, PERR# de-assertion, and DEVSEL# assertion.

Implication: Tval timing is well within the PCI spec for nominal Vcc and 50/50 PCI clock duty cycle. Typical PCI systems should be able to tolerate Tval timings of 14 ns.

Workaround: None.

Status: *No current plan to fix this erratum*.

4) **Symptom:** Part A - On the first attempt of reading CSR30, the system will suffer a PCI Retry. A subsequent CSR30 access will be allowed to read the correct CSR30 data. Part B - If the LAAINC (BCR29, bit 14) bit is set, and if a CSR30 read access is in the middle of a series of continuous reads of BCR30 for SRAM/Flash access, the CSR30 access causes the SRAM/Flash address to auto-increment.

Implication: Part A - This is a minor erratum with no impact on the system. The PCI Retry does not cause any system errors and will automatically recover. Part B - The system-level exposure to this problem is very minimal. There is no exposure or customers of AMD's PCnet drivers, because AMD's drivers do not perform CSR30 reads. For those developing proprietary diagnostic software routines, please follow the workaround recommended below.

Workaround: Part A - None needed and none available. Part B - Disable the LAAINC bit prior to executing the specific register access sequence outlined in the symptom statement; LAAINC bit may be enabled subsequent to executing the CSR30 read access.

Status: *No current plan to fix this erratum*.

5) **Symptom:** In a high traffic network and with SRAM on and in 10Mbps mode, the babble error bit occasionally gets set erroneously; no transmitter babbles are observed on the wire.

Implication: Some diagnostics software will report false babbles. No impact for customers who utilize the PCnet family drivers since PCnet drivers do not report babble errors.

Workaround: Ignore the babble errors.

Status: *No current plan to fix this erratum*.

6) **Symptom:** In an excessively high collision network, the device occasionally set the LCOL bit erroneously in 10Mbps mode with external SRAM enabled.

Implication: No discernible performance impact. Problem has only been observed in lab set up. Some diagnostics software will report false LCOL errors.

Workaround: None.

Status: *No current plan to fix this erratum*.

7) **Symptom:** When PCnet-FAST receives a frame on the MII port with correct FCS followed by exactly one nibble with RX_ER asserted, it does not report a FCS error.

Implication: None in an 802.3 compliant network. Observable in custom diagnostic tests for End of Shell Delimiter (ESD) only.

Workaround: None.

Status: *No current plan to fix this erratum*.

8) **Symptom:** When RCVE bit (bit 2, CSR4, 5, 6, and 7) is set and the MII port is selected, the LED output does not indicate the correct receive status. This bit functions correctly for the internal 10 BASE-T and AUI ports.

Implication: LED output will not be asserted based on the receive activities at the MII port.

Workaround: Use RCVME bit (bit 5, CSR 4, 5, 6 and 7) in place of the RCVE bit for proper LED receive status indication. The RCVME bit enables the indication of all received packets which pass the address match function for this node, whereas the RCVE bit enables the indication of all packets received.

Status: *No current plan to fix this erratum*.

9) **Symptom:** During reception of a packet, if the CRS input to the device is de-asserted two RX_CLK times before the end of RX_DV, message byte counter (MCNT) will indicates one less byte and the last byte of the CRC will be corrupt. If the CRS is de-asserted three or more RX_CLK times before the end of RX_DV, the CRC error bit will be set and the packet will be lost.

Implication: Network performance might vary from normal to sluggish to no connection depending on the severity and the rate of occurrence.

Workaround: The workaround for this problem is to OR the RX_DV and CSR signals from external PHY and feed the output to the CRS input of the PCnet-FAST.

Status: *No current plan to fix this erratum*.

10) **Symptom:** When the device is connected to certain repeaters and hubs, there have been reports of receiving corrupted data and bad packets. The hubs and repeaters that are used when these problems occur do not comply with the IEEE 802.3 specification. Specifically they do not comply with the following tolerance: +/- 100 PPM. The problem is aggravated due to the fact that the device does not flag any error message as a result of this. In addition the transfers corrupted packets and continues the network activity. The received packets are corrupted but the CRC error bit does not get set.

Implications: a) In situations where there is no upper-layer protocol to detect the error, there is a possibility that data may be corrupted when the device is used in conjunction with a non-IEEE 802.3 compliant hub or repeater. b) In situations where the upper layer protocol detects the error (which is the most common situation there may be some degradation in network performance. There is also a possibility of failure of the link or an inability to maintain link status.

Workaround: It is suggested to adhere to IEEE compliant hubs and repeaters. This will avoid the circumstances that may compromise data integrity and or contribute to degradation of network performance

Status: There are no planned modifications for the device. However, modifications have been implemented in later devices in the PCnet family, such that they can tolerate non-compliant IEEE 803.2 repeaters and hub.

11) **Symptom:** When the RCVME bit in the LED registers (BCR4-7) is set, the LED output drives for one clock cycle, or longer if the pulse stretcher is on, due to reception of any packet. The expected behavior is that this LED should be on only when there is an address match

Implications: LED output will drive every time there is an incoming packet on the wire, even when there is no address match.

Workaround: None

Status: *No current plan to fix this erratum*.

12) **Symptom:** Setting the DRCVBC bit (CSR15, bit 14) does not prevent the reception of broadcast packets when the LADRF[47 (CSR10, bit 15) is set.

Implications: Broadcast packets will be received when the user wants them excluded as a Logical packet. Extra packet processing will be required.

Workaround: None

Status: *No current plan to fix this erratum*.

13) **Symptom:** Configuration Space Vendor ID (VID) Register can not a have a value other than 0x1022.

Implications: The Configuration Space VID is programmed indirectly through the BCR35. If any value other than 0x1022 is programmed in BCR35 either through software or thorough the EEPROM, the BCR35 will have the correct data but the VID register will have a corrupted value.

Workaround: Use only a value of 0x1022

Status: *No current plan to fix this erratum*.

14) **Symptom:** When using the advance parity mode, the occurrence of parity error in the last transfer of descriptor DMA write does not stop the chip as indicated in the data sheet.

Implication: If customized or proprietary software relies on the chip to stop due to parity error in the advanced parity mode, then it is possible (though highly unlikely in a typical system) to get corrupted data in descriptors. PCnet software drivers do rely on the chip to stop the device upon parity error.

Workaround: Use the interrupt generated by the SINT as an indicator for parity error.

Status: *No current plan to fix this erratum*.

15) **Symptom:** When using buffer (descriptor) chaining for transmit packets, BUFF errors are reported under certain conditions.

Implication: Transmit packets associated with this condition are truncated.

Workaround: Do not allow a chain of TX descriptors to include a descriptor which previously was an end of chain (EOP) and which is the last TX EOP descriptor for which the status has been returned. Here is a suggested method for the workaround:

The device driver's TX Write pointer (TX_W) points to first TX descriptor with OWN = 0

The device driver's TX Read pointer (TX_R) points to the last TX descriptor with OWN=0

As the TX_R pointer is advancing in the chain, if the EOP bit is set then set a device driver TX_Last EOP pointer (TX_L) = Current TX_R pointer

Then when a new frame is to be transmitted:

1) If a single descriptor is needed

if TX_R >TX_W, then a descriptor is available

2) If buffer chaining

if TX_L =TX_W, then TX_R -TX_W descriptors are available

else only TX_L - TX_W descriptors are available

Status: *No current plan to fix this erratum*.

16) Symptom: Collision LED does not show the collision status of the MII bus.

Implication: Very minor, collision LED will not show the status.

Workaround: None.

Status: *No current plan to fix this erratum*.

17) Symptom: During the automatic read of the EEPROM, such as after hardware reset or when the PREAD bit is set in the BCR19, the device drives the EECS and EESK signal pins simultaneously. The EECS is driven high at the same time when the EESK is driving low. This violates the EEPROM clock low to chip select setup time (tSKS) parameter, which should be 100ns.

Implication: None. Even though the tSKS parameter is technically violated, the relationship between EESK (clock), EECS (chip select), and EEDI (data in) signals is such that the device will not detect a false opcode. This is how these signals relate to each other. EESK starts toggling from a high level. It clocks twice before the EECS is asserted and continues toggling. The EEDI input is driven low at the same time that the EESK starts toggling and stay low for four clocks after the assertion of the EECS. Since the EEDI is low when the EECS is asserted, no false opcode is detected by the EEPROM.

Workaround: None needed. An external circuit may be used to delay the EECS signal by 100ns from the falling edge of the EESK signal

Status: *No current plan to fix this erratum*.

AM79C971A SYSTEM DESIGN HINTS

1) In 100 Mb/s mode, if the device reports excessive Transmit Underflows, set the NOUFLO bit (BCR18, bit 11) to 1. BCR18 contents are programmable either through the EEPROM or software driver.

2) In a system which does not use the reset pulse (RST#) for a warm boot reset (also known as Ctrl-Alt-Del reset), the PCI-SIG recommends that the BIOS should disable bus mastering capability of the PCI bus mastering devices early in the reboot cycle. The disabling of the bus mastering capability can be accomplished by resetting the BMEN bit in the PCI Command Register (bit 2, Offset 04h) of the device's PCI configuration space. This recommendation should be followed to avoid possible system hang.

3) In the 100Mb/s Full Duplex mode, the expansion bus clock (which is typically connected to the PCI bus clock) speed needs to be at 33MHz. Any slower speed may cause under/over flow condition.

4) When in auto-polling mode and no receive descriptors are available, the transmitter will not transmit until either a receive descriptor is available or the transmit demand bit (TDMD, CSR0, bit 3) is set. If there are no receive descriptors available, only one packet will be transmitted for every TDMD.

5) When using Auto-polling (BCR32, bit11), the ASEL bit (BCR2, bit1) should be reset to one, in order for the MII Auto-poll logic (this is not the same as the descriptor Auto-polling) to correctly detect the link status change on the MII PHY.

6) During the assertion of RST#, the EECS output becomes tri-stated. It is possible that the EECS may float to a logic high state during this time. In order to prevent a hazardous condition due to this inadvertent selection of the EEPROM, connect a 10K pull-down resistor from EECS to ground.

7) It is recommended that each poll to the SPND bit be performed with a software implemented delay of approximately 2-3 ms intervals. When the SPND bit (CSR5, bit0) is set in the device, the time that it takes to enter the suspend mode is dependant on several factors. Some of these factors are; the number of packets queued for transmission in the internal memory, the received packets still in the internal memory, the PCI bus grant time, and the transmit channel availability in the half-duplex mode. To minimize unnecessary PCI Bus activity and allow access to the PCI bus, it is recommended that each poll to the SPND bit be performed with a software implemented delay of approximately 2-3 ms intervals. Successive polling or polling implemented with a hardware delay will inhibit the device from completing its RX/TX DMA, causing longer delays before the device to enter suspend mode.

INDEX

A

B

C

D

E

N

O

P

R

S

T

U

Trademarks

AMD, the AMD logo and combinations thereof, HIMIB, IMR+, Magic Packet, PCnet, and PCnet-*FAST*, are trademarks of Advanced Micro Devices, Inc.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.