PRELIMINARY

AMD

Am79C973/Am79C975

PCnet™-*FAST III*
Single-Chip 10/100 Mbps PCI Ethernet Controller with Integrated PHY

DISTINCTIVE CHARACTERISTICS

- **Single-chip PCI-to-Wire Fast Ethernet controller**
  - **— 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 2.2 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**
  - **— Big endian and little endian byte alignments supported**
- **Fully Integrated 10/100 Mbps Physical Layer Interface (PHY)**
  - **— Conforms to IEEE 802.3 standard for 10BASE-T, 100BASE-TX, and 100BASE-FX interfaces**
  - **— Integrated 10BASE-T transceiver with on-chip filtering**
  - **— Fully integrated MLT-3 encoder/decoder for 100BASE-TX**
  - **— Provides a PECL interface for 100BASE-FX fiber implementations**
  - **— Full-duplex capability for 10BASE-T and 100BASE-TX**
  - **— IEEE 802.3u Auto-Negotiation between 10 Mbps and 100 Mbps, half- and full-duplex operation**
- **Dual-speed CSMA/CD (10 Mbps and 100 Mbps) Media Access Controller (MAC) compliant with IEEE/ANSI 802.3 and Blue Book Ethernet standards**
- **Supports PC98/PC99 and Wired for Management baseline specifications**
  - **— Full OnNow support including pattern matching and link status wake-up events**
  - **— Implements AMD's patented Magic Packet™ technology for remote wake-up & power-on**
  - **— Magic Packet mode and the physical address loaded from EEPROM at power up without requiring PCI clock**
  - **— Supports PCI Bus Power Management Interface Specification Revision 1.1**
  - **— Supports Advanced Configuration and Power Interface (ACPI) Specification Version 1.0**
  - **— Supports Network Device Class Power Management Specification Version 1.0a**
- **Serial Management Interface enables remote alerting of system management events**
  - **— Inter-IC (I^2C) compliant electrical interface**
  - **— System Management Bus (SMBus) compliant signaling interface and register access protocol**
  - **— Optional interrupt pin simplifies software interface**
- **Large independent internal TX and RX FIFOs**
  - **— 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 extensive programmability of device operation through EEPROM mapping**
- **Supports up to 1 megabyte (Mbyte) optional Boot PROM and Flash for diskless node application**
- **Extensive programmable internal/external loopback capabilities**
- **Extensive programmable LED status support**

Publication# **21510** Rev: **E** Amendment/**0**
Issue Date: **August 2000**

Refer to AMD's Website (www.amd.com) for the latest information.

- **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**
- **Compatible with the existing PCnet Family driver/diagnostic software**
- **Software compatible with AMD PCnet Family and LANCE™/C-LANCE™ register and descriptor architecture**
- **Available in 160-pin PQFP and 176-pin TQFP packages**
- **Advanced +3.3 V CMOS process technology for low power operation**

GENERAL DESCRIPTION

The Am79C973 and Am79C975 controllers are single-chip 32-bit full- duplex, 10/100-Megabit per second (Mbps) fully integrated PCI-to-Wire Fast Ethernet system solution, designed to address high-performance system application requirements. They are 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 and low CPU and system bus utilization. The Am79C973 and Am79C975 controllers are fabricated with advanced low-power 3.3-V CMOS process to provide low operating current for power sensitive applications.

The third generation Am79C973 and Am79C975 Fast Ethernet controllers also have several enhancements over their predecessors, the Am79C971 and Am79C972 devices. Besides integrating the complete 10/100 Physical Layer (PHY) interface, they further reduce system implementation cost by integrating the SRAM buffers on chip.

The Am79C973 and Am79C975 controllers contain 12-kilobyte (Kbyte) buffers, the largest of their class in 10/100 Mbps Ethernet controllers. The large internal buffers are fully programmable between the RX and TX queues for optimal performance.

The Am79C973 and Am79C975 controllers are also compliant with PC98/PC99 and Wired for Management specifications. They fully support Microsoft's OnNow and ACPI specifications, which are backward compatible with Magic Packet technology and compliant with the PCI Bus Power Management Interface Specification by supporting the four power management states (D0, D1, D2, and D3), the optional $\overline{\text{PME}}$ pin, and the necessary configuration and data registers.

The Am79C973 and Am79C975 controllers are complete Ethernet nodes 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, and an IEEE 802.3-compliant 10/100 Mbps PHY.

The integrated 10/100 PHY unit of the Am79C973 and Am79C975 controllers implement the complete physical layer for 10BASE-T and the Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA), and Physical Medium Dependent (PMD) functionality for 100BASE-TX, including MLT-3 encoding/decoding. It also supports 100BASE-FX operation by providing a Pseudo-ECL (PECL) interface for direct connection to a fiber optic transceiver module. The internal 10/100 PHY implements Auto-Negotiation for twisted-pair (10T/100TX) operation by using a modified 10BASE-T link integrity test pulse sequence as defined in the IEEE 802.3u specification. The Auto-Negotiation function automatically configures the controller to operate at the maximum performance level supported across the network link.

The Am79C975 controller also implements a Serial Management Interface in addition to the advanced management features offered with the Am79C973 controller. The Serial Management Interface is based on the industry standard Inter-IC (I^2C) and System Management Bus (SMBus) specifications and enables a system to communicate with another network station for remote monitoring and alerting of local system management parameters and events. This simple yet powerful Serial Management Interface is capable of communicating within the system and over the network during normal operation or in low-power modes, even if the device is not initialized or set up for transmit or receive operation by the network software driver.

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 Am79C973 and Am79C975 controllers provide 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 their built-in support for both little and big endian byte alignment, the

controllers also address non-PC applications. The Am79C973 and Am79C975 controllers' 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.

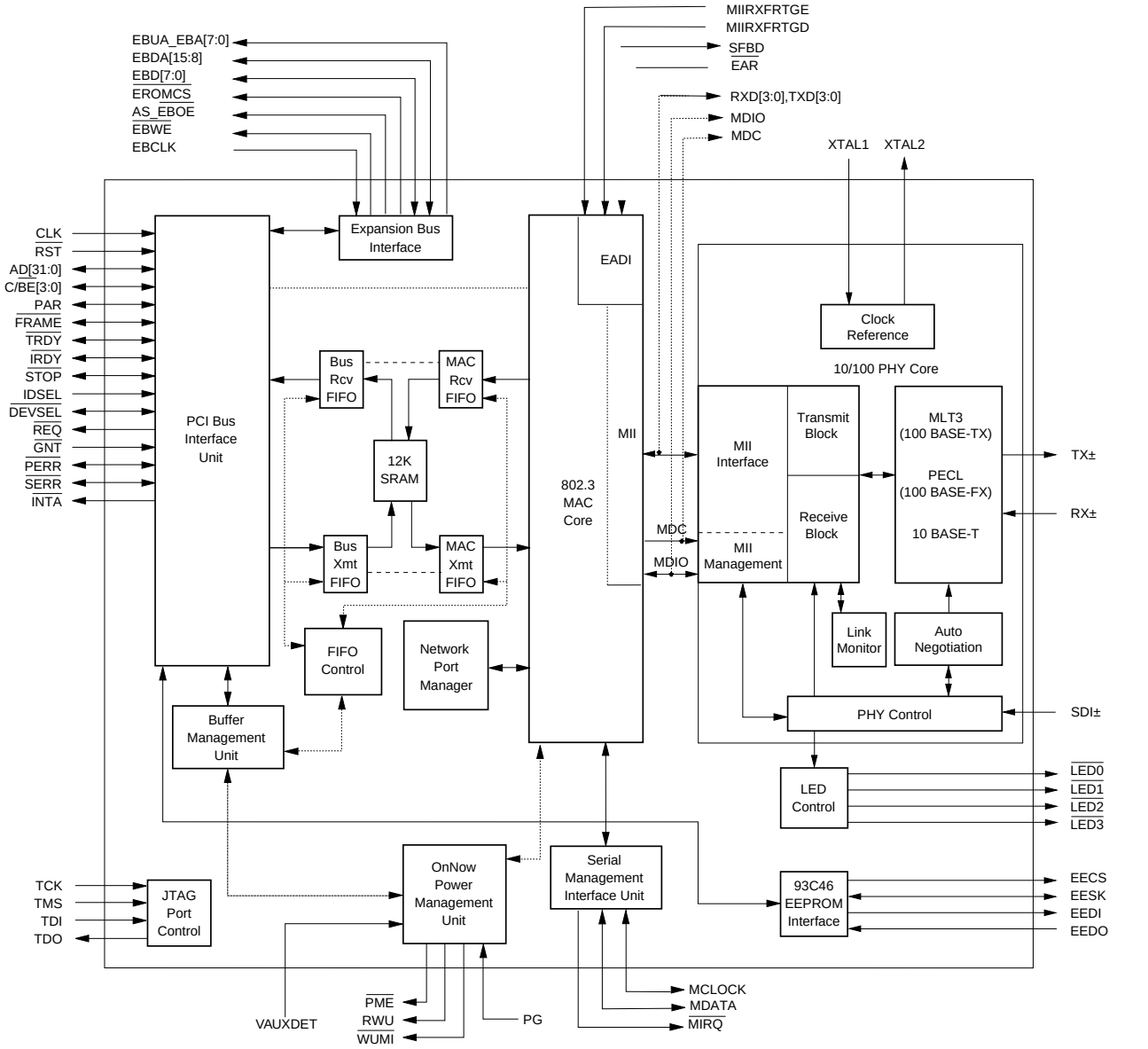
The Am79C973 and Am79C975 controllers support auto-configuration in the PCI configuration space. Additional Am79C973 and Am79C975 controller configuration parameters, including the unique IEEE physical address, can be read from an external non-volatile memory (EEPROM) immediately following system reset.

In addition, the Am79C973 and Am79C975 controllers provide programmable on-chip LED drivers for transmit, receive, collision, link integrity, Magic Packet status, activity, link active, address match, full-duplex, 10 Mbps or 100 Mbps, or jabber status.

The Am79C973 and Am79C975 controllers are 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™ (Am79C960), PCnet-ISA+™ (Am79C961), PCnet-ISA II™ (Am79C961A), PCnet-32™ (Am79C965), PCnet-PCI™ (Am79C970), PCnet-PCI II™ (Am79C970A), PCnet-*FAST*™ (Am79C971), and PCnet-*FAST*+™ (Am79C972). The Buffer Management Unit supports the LANCE and PCnet descriptor software models.

The Am79C973 and Am79C975 controllers are ideally suited for LAN on the motherboard, network adapter card, and embedded designs. It is available in a 160-pin Plastic Quad Flat Pack (PQFP) package and also in a 176-pin Thin Quad Flat Pack (TQFP) package for form factor sensitive designs.

BLOCK DIAGRAM



21510D-1

TABLE OF CONTENTS

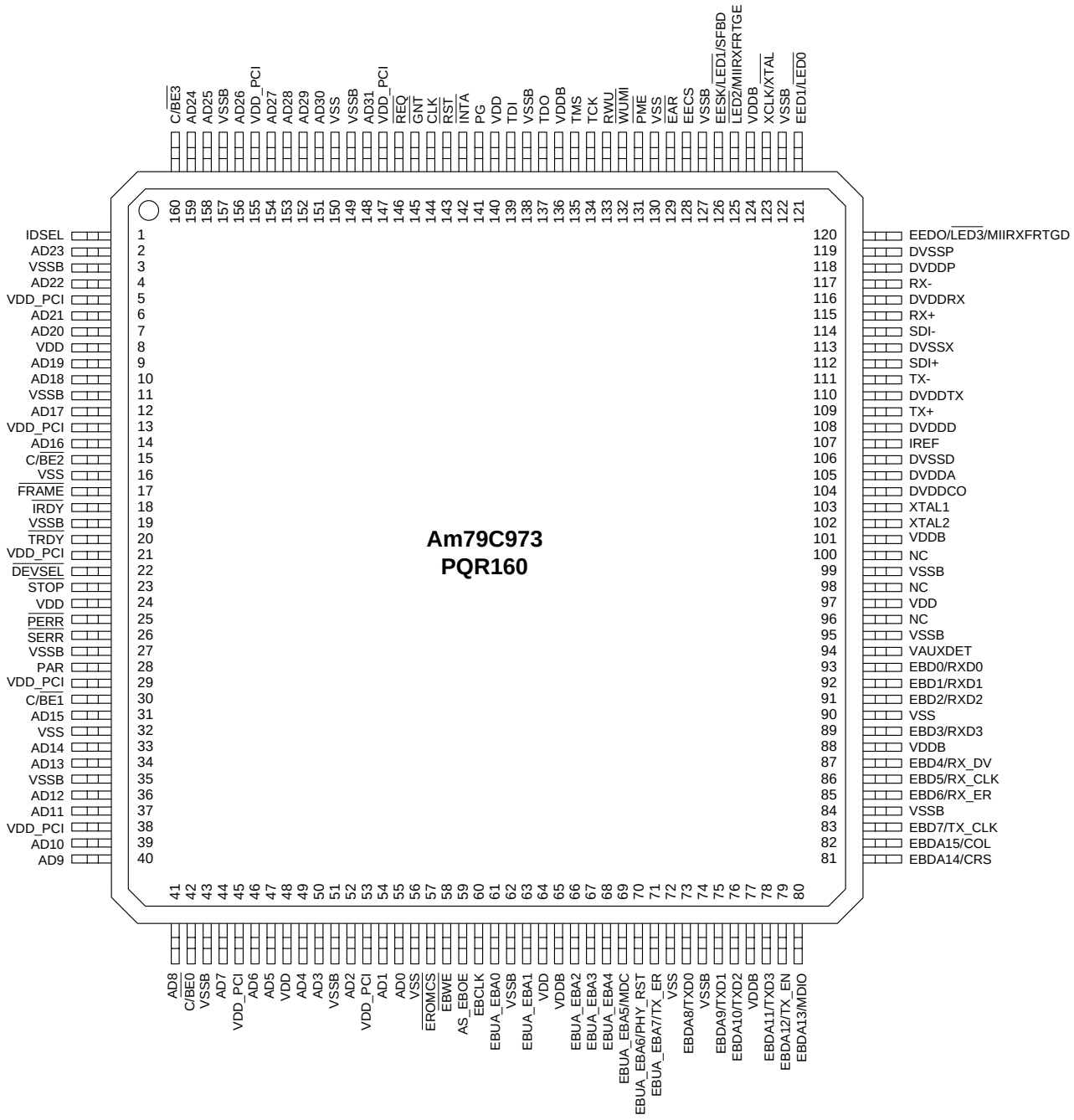
LIST OF FIGURES

LIST OF TABLES

RELATED AMD PRODUCTS

Part No.	Description
Controllers	
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE™)
Integrated Controllers	
Am79C930	PCnet™-Mobile Single Chip Wireless LAN Media Access Controller
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C961A	PCnet-ISA II Full Duplex Single-Chip Ethernet Controller for ISA Bus
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller for 486 and VL Buses
Am79C970A	PCnet-PCI II Full Duplex Single-Chip Ethernet Controller for PCI Local Bus
Am79C971	PCnet-FAST Single-Chip Full-Duplex 10/100 Mbps Ethernet Controller for PCI Local Bus
Am79C972	PCnet-FAST+ Enhanced 10/100 Mbps PCI Ethernet Controller with OnNow Support
Manchester Encoder/Decoder	
Am7992B	Serial Interface Adapter (SIA)
Physical Layer Devices (Single-Port)	
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am79761	Physical Layer 10-Bit Transceiver for Gigabit Ethernet (GigaPHY™-SD)
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Physical Layer Devices (Multi-Port)	
Am79C871	Quad Fast Ethernet Transceiver for 100BASE-X Repeaters (QFEXr™)
Am79C988A	Quad Integrated Ethernet Transceiver (QuIET™)
Am79C989	Quad Ethernet Switching Transceiver (QuEST™)
Integrated Repeater/Hub Devices	
Am79C981	Integrated Multiport Repeater Plus (IMR+)
Am79C982	Basic Integrated Multiport Repeater (bIMR)
Am79C983	Integrated Multiport Repeater 2 (IMR2™)
Am79C984A	Enhanced Integrated Multiport Repeater (eIMR™)
Am79C985	Enhanced Integrated Multiport Repeater Plus (eIMR+™)
Am79C987	Hardware Implemented Management Information Base (HIMIB™)

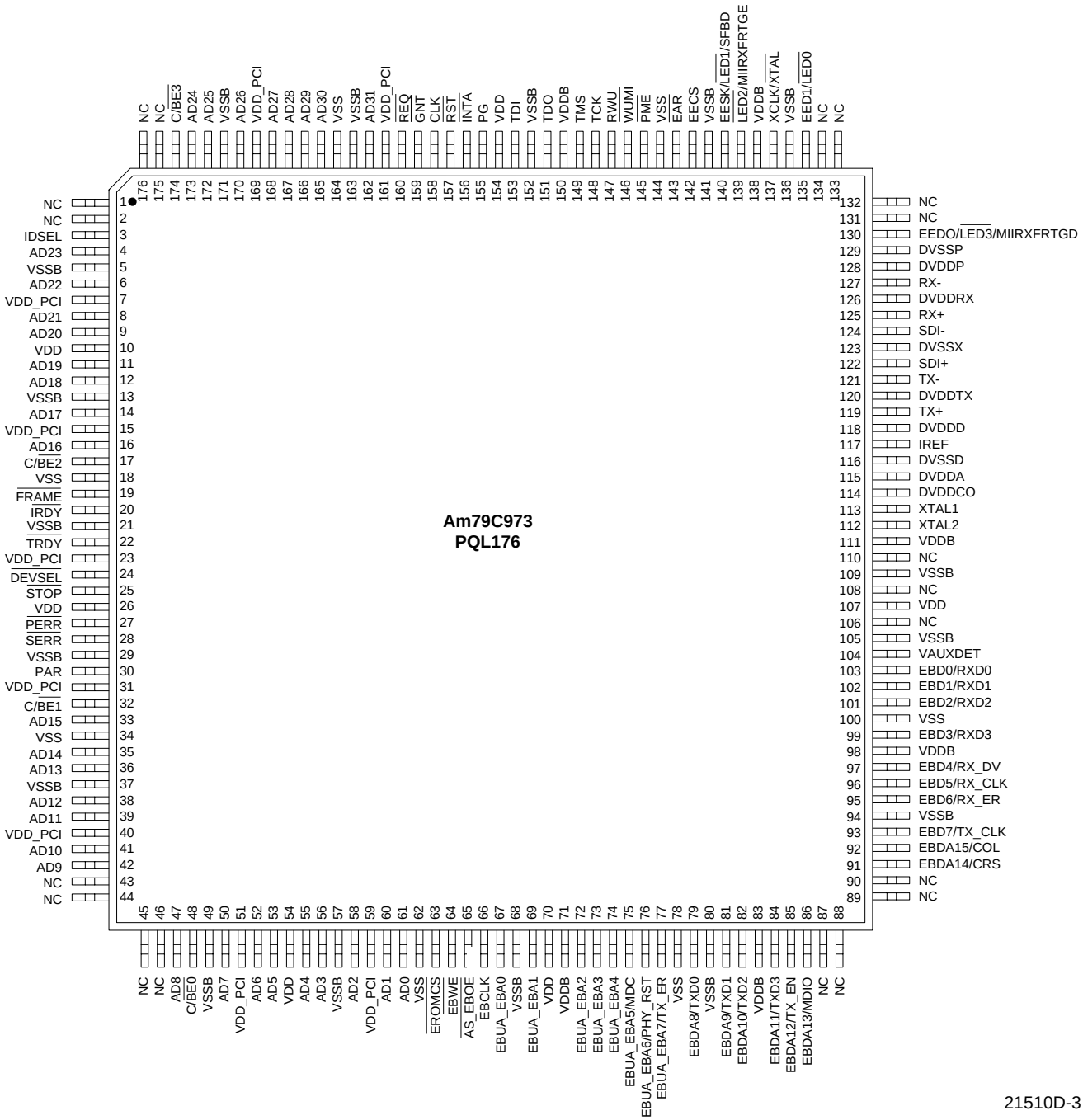
CONNECTION DIAGRAM (PQR160) - Am79C973



21510D-2

Pin 1 is marked for orientation.

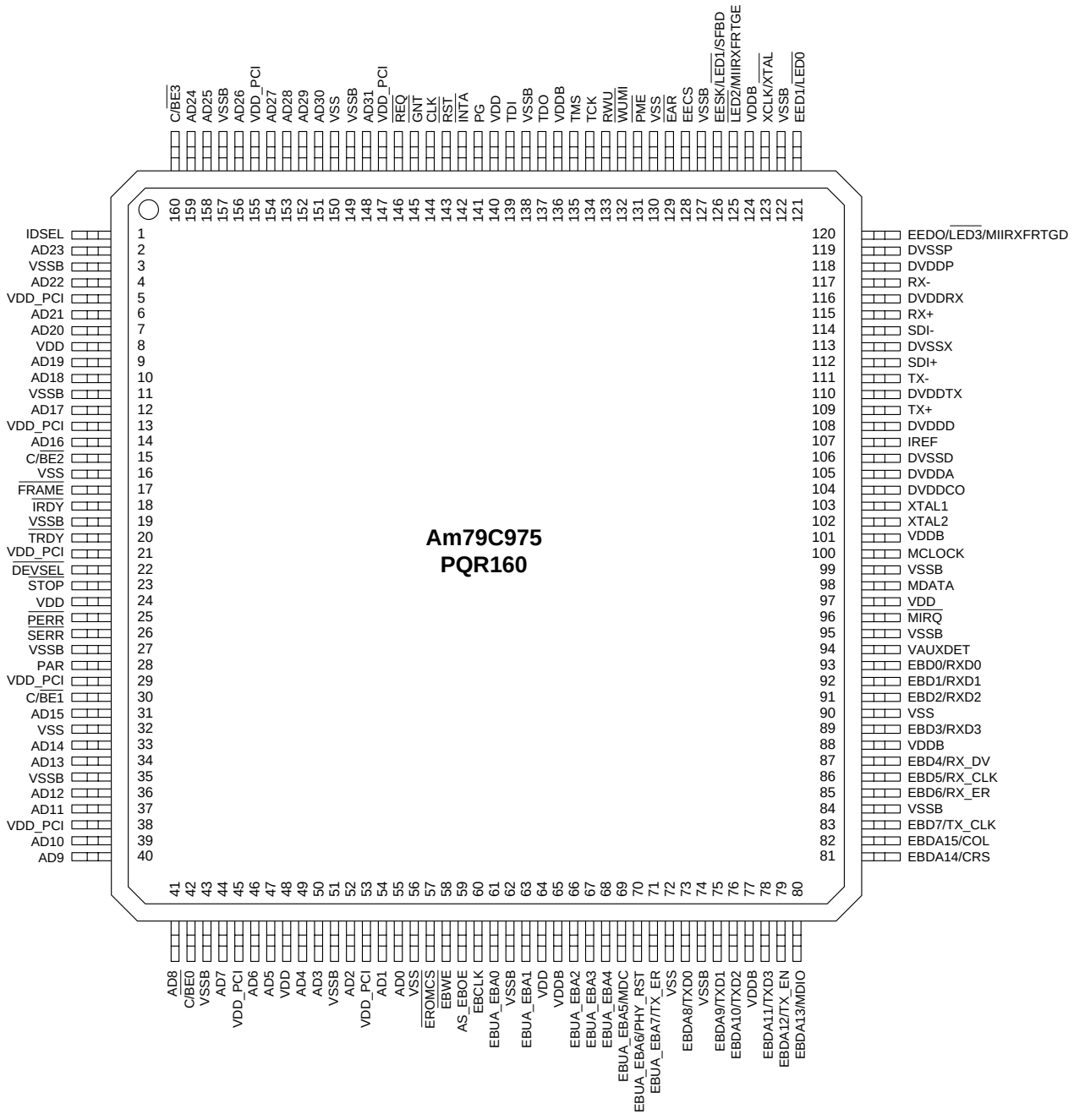
CONNECTION DIAGRAM (PQL176) Am79C973



21510D-3

Pin 1 is marked for orientation.

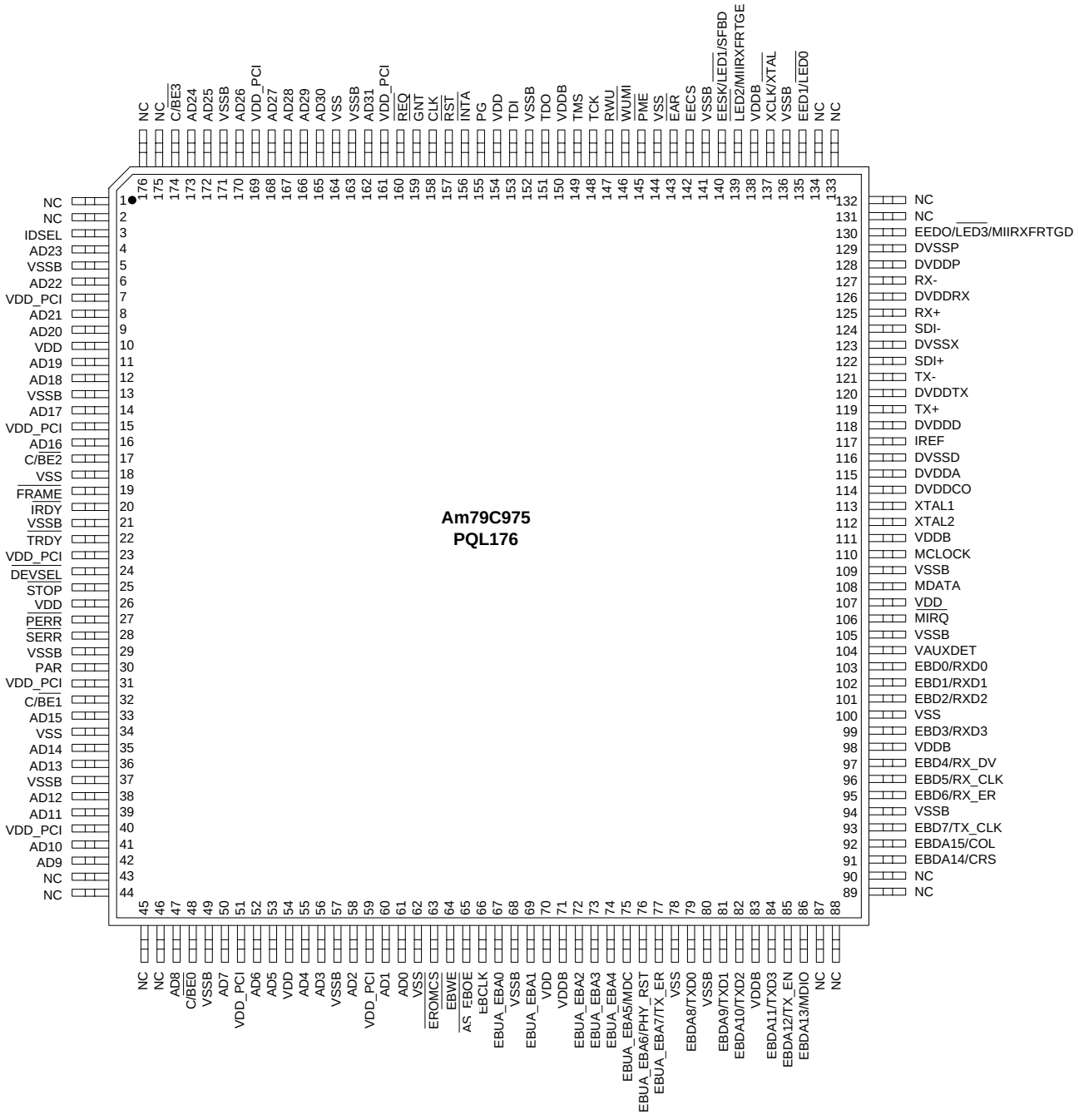
CONNECTION DIAGRAM (PQR160) - Am79C975



21510D-4

Pin 1 is marked for orientation.

CONNECTION DIAGRAM (PQL176) - Am79C975



21510D-5

Pin 1 is marked for orientation.

PIN DESIGNATIONS (PQR160) (Am79C973/Am79C975)

Listed By Pin Number

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

Note: *For the Am79C973 controller, pins 96, 98, and 100 are no connects (NC).*

PIN DESIGNATIONS (PQL176) (Am79C973/Am79C975)

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	AD8	91	EBDA14/CRS	135	EEDI/LED0
4	AD23	48	C/BE0	92	EBDA15/COL	136	VSSB
5	VSSB	49	VSSB	93	EBD7/TX_CLK	137	XCLK/XTAL
6	AD22	50	AD7	94	VSSB	138	VDDB
7	VDD_PCI	51	VDD_PCI	95	EBD6/RX_ER	139	LED2/MIIRXFRTGE
8	AD21	52	AD6	96	EBD5/RX_CLK	140	EESK/LED1/SFDB
9	AD20	53	AD5	97	EBD4/RX_DV	141	VSSB
10	VDD	54	VDD	98	VDDB	142	EECS
11	AD19	55	AD4	99	EBD3/RXD3	143	EAR
12	AD18	56	AD3	100	VSS	144	VSS
13	VSSB	57	VSSB	101	EBD2/RXD2	145	PME
14	AD17	58	AD2	102	EBD1/RXD1	146	WUMI
15	VDD_PCI	59	VDD_PCI	103	EBD0/RXD0	147	RWU
16	AD16	60	AD1	104	VAUXDET	148	TCK
17	C/BE2	61	AD0	105	VSSB	149	TMS
18	VSS	62	VSS	106	MIRQ (see Note)	150	VDDB
19	FRAME	63	EROMCS	107	VDD	151	TDO
20	IRDY	64	EBWE	108	MDATA (see Note)	152	VSSB
21	VSSB	65	AS_EBOE	109	VSSB	153	TDI
22	TRDY	66	EBCLK	110	MCLOCK (see Note)	154	VDD
23	VDD_PCI	67	EBUA_EBA0	111	VDDB	155	PG
24	DEVSEL	68	VSSB	112	XTAL2	156	INTA
25	STOP	69	EBUA_EBA1	113	XTAL1	157	RST
26	VDD	70	VDD	114	DVDDCO	158	CLK
27	PERR	71	VDDB	115	DVDDA	159	GNT
28	SERR	72	EBUA_EBA2	116	DVSSD	160	REQ
29	VSSB	73	EBUA_EBA3	117	IREF	161	VDD_PCI
30	PAR	74	EBUA_EBA4	118	DVDDD	162	AD31
31	VDD_PCI	75	EBUA_EBA5/MDC	119	TX+	163	VSSB
32	C/BE1	76	EBUA_EBA6/ PHY_RST	120	DVDDTX	164	VSS
33	AD15	77	EBUA_EBA7/TX_ER	121	TX-	165	AD30
34	VSS	78	VSS	122	SDI+	166	AD29
35	AD14	79	EBDA8/TXD0	123	DVSSX	167	AD28
36	AD13	80	VSSB	124	SDI-	168	AD27
37	VSSB	81	EBDA9/TXD1	125	RX+	169	VDD_PCI
38	AD12	82	EBDA10/TXD2	126	DVDDRX	170	AD26
39	AD11	83	VDDB	127	RX-	171	VSSB
40	VDD_PCI	84	EBDA11/TXD3	128	DVDDP	172	AD25
41	AD10	85	EBDA12/TX_EN	129	DVSSP	173	AD24
42	AD9	86	EBDA13/MDIO	130	EEDO/LED3/ MIIRXFRTGD	174	C/BE3
43	NC	87	NC	131	NC	175	NC
44	NC	88	NC	132	NC	176	NC

Note: *For the Am79C973 controller, pins 106, 108, and 110 are no connects (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	O	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
XCLK	External Clock Source Select	I	NA	1
$\overline{XTAL}$	Crystal Select	I	NA	1
XTAL1	Crystal Input -25 MHz Clock Reference	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
$\overline{EROMCS}$	Expansion Bus ROM Chip Select	O	O6	1

1. Not including test features

PIN DESIGNATIONS

Listed By Group

Pin Name	Pin Function	Type[1]	Driver	No. of Pins
Physical Layer Interface (PHY)				
IREF	Internal Current Reference	I	NA	1
RX±	Serial Receive Data	I	NA	2
TX±	Serial Transmit Data	O	NA	2
SDI±	Signal Detect Input	I	NA	2
Power Management Interface				
RWU	Remote Wake Up	O	O6	1
$\overline{\text{PME}}$	Power Management Event	O	OD6	1
$\overline{\text{WUMI}}$	Wake-Up Mode Indication	O	OD6	1
PG	Power Good	I	NA	1
VAUXDET	Auxiliary Power Detect	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
External Address Detection Interface (EADI)				
$\overline{\text{EAR}}$	External Address Reject Low	I	NA	1
SFBD	Start Frame Byte Delimiter	O	LED	1
MIIRXFRTGD	MII Receive Frame Tag Data	I	NA	1
MIIRXFRTGE	MII Receive Frame Tag Enable	I	NA	1
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	I/O	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
Serial Management Interface (SMI) - Am79C975 only				
MCLOCK	SMI Clock	I/O	OD6	1
MDATA	SMI Data	I/O	OD6	1
$\overline{\text{MIRQ}}$	SMI Interrupt	O	OD6	1

Note:

1. Not including test features.

PIN DESIGNATIONS

Listed by Group (Concluded)

Pin Name	Pin Function	Type[1]	Driver	No. of Pins
Power Supplies (MAC, PCI, Buffer, ROM)				
VDD	Digital Power	P	NA	6
VSS	Digital Ground	P	NA	7
VDDB	I/O Buffer Power	P	NA	6
VSSB	I/O Buffer Ground	P	NA	17
VDD_PCI	PCI I/O Buffer Power	P	NA	9
Power Supplies (PHY)[2]				
DVDDA	Analog PLL Power	P	NA	1
DVDDD, DVDDP	Physical Data Transceiver (PDX) Power, IREF	P	NA	2
DVSSD, DVSSP	Physical Data Transceiver (PDX) Ground	P	NA	2
DVDDTX, DVDDRX	PHY I/O Buffer Power	P	NA	2
DVSSX	PHY Ground	P	NA	1
DVDDCO	Crystal Oscillator Power	P	NA	1

***Notes**:*

1. Not including test features.

2. PHY power and ground pins require careful decoupling to ensure proper device performance.

PIN DESIGNATIONS

Listed By Driver Type

The following table describes the various types of output drivers used in the Am79C973/Am79C975 controller. All I_{OL} and I_{OH} values shown in the table apply 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.

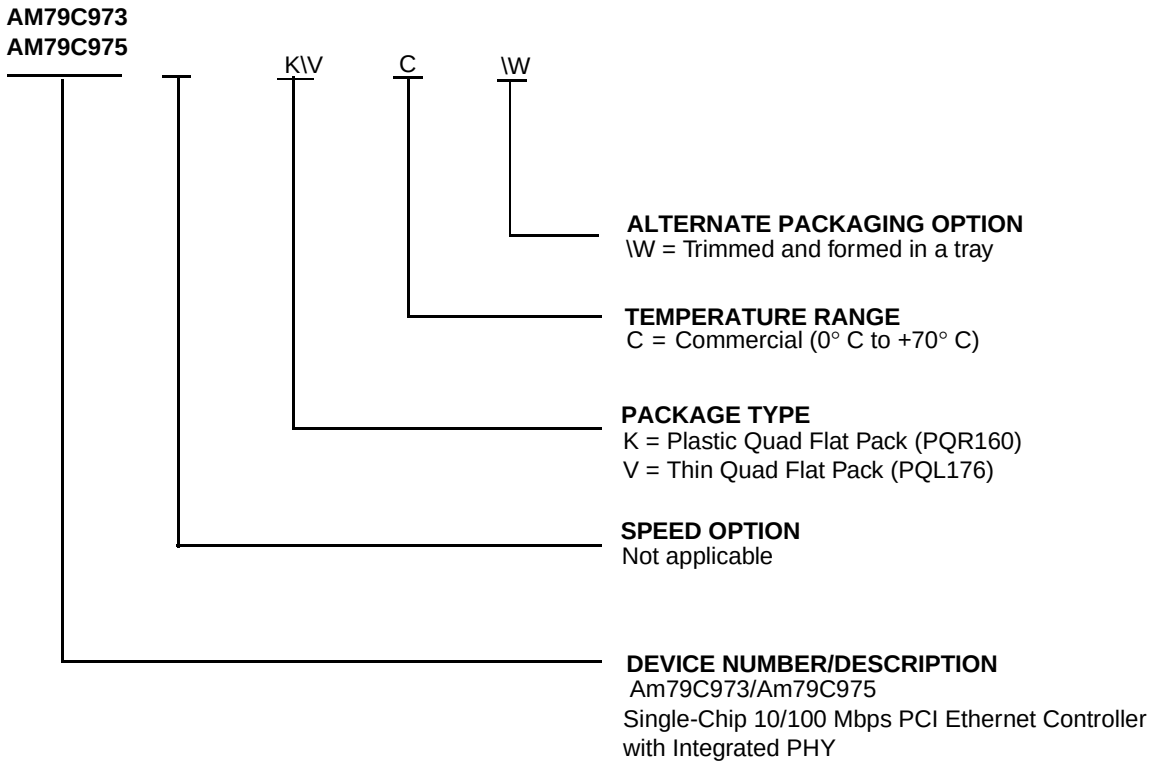
TX is a differential output driver. Its characteristics and those of XTAL2 output are described in the *DC Characteristics* section.

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

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	
AM79C973, AM79C975	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.

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[7:0] 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller is a bus master, AD[31:2] will address the active Double Word (DWord). The Am79C973/Am79C975 controller always drives AD[1:0] to '00' during the address phase indicating linear burst order. When the Am79C973/Am79C975 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 Am79C973/Am79C975 controller when performing bus master write and slave read operations. Data on AD[31:0] is latched by the Am79C973/Am79C975 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}$0 applies to byte 0 (AD[7:0]) and C/$\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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller.

The Am79C973/Am79C975 controller supports bus parking. When the PCI bus is idle and the system arbiter asserts $\overline{GNT}$ without an active $\overline{REQ}$ from the Am79C973/Am79C975 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 Am79C973/Am79C975 controller during configuration read and write transactions.

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

INTA

Interrupt Request **Output**

An attention signal which indicates that one or more of the following status flags is set: EXDINT, IDON, MERR, MISS, MFCO, MPINT, RCVCCO, RINT, SINT, TINT, TXSTRT, UINT, MCCINT, MPDTINT, MAPINT, MREINT, and STINT. Each status flag has either a mask or an enable bit which allows for suppression of INTA assertion. Table 1 shows the flag descriptions. By default INTA is an open-drain output. For applications that need a high-active edge-sensitive interrupt signal, the INTA pin can be configured for this mode by setting INTLEVEL (BCR2, bit 7) to 1.

When RST is active, INTA is the output for NAND tree testing.

IRDY

Initiator Ready **Input/Output**

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

When the Am79C973/Am79C975 controller is a bus master, it asserts IRDY during all write data phases to indicate that valid data is present on AD[31:0]. During all read data phases, the device asserts IRDY to indicate that it is ready to accept the data.

When the Am79C973/Am79C975 controller is the target of a transaction, it checks IRDY during all write data phases to determine if valid data is present on AD[31:0]. During all read data phases, the device checks IRDY to determine if the initiator is ready to accept the data.

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

PAR

Parity **Input/Output**

Parity is even parity across AD[31:0] and C/BE[3:0]. When the Am79C973/Am79C975 controller is a bus master, it generates parity during the address and write data phases. It checks parity during read data phases. When the Am79C973/Am79C975 controller operates in slave mode, it checks parity during every address phase. When it is the target of a cycle, it checks parity during write data phases and it generates parity during read data phases.

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

.

Table 1. Interrupt Flags

Name	Description	Mask Bit	Interrupt Bit
EXDINT	Excessive Deferral	CSR5, bit 6	CSR5, bit 7
IDON	Initialization Done	CSR3, bit 8	CSR0, bit 8
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
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
MCCINT	MII Management Command Complete Interrupt	CSR7, bit 4	CSR7, bit 5
MPDTINT	MII PHY Detect Transition Interrupt	CSR7, bit 0	CSR7, bit 1
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

PERR

Parity Error **Input/Output**

During any slave write transaction and any master read transaction, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller asserts REQ pin as a signal that it wishes to become a bus master. REQ is driven high when the Am79C973/Am79C975 controller does not request the bus. In Power Management 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 and the PG pin is HIGH, then the Am79C973/Am79C975 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 Am79C973/Am79C975 controller will disable or deassert all outputs. RST may be asynchronous to clock when asserted or deasserted.

When the PG pin is LOW, RST disables all of the PCI pins except the PME pin.

When RST is LOW and PG is HIGH, NAND tree testing is enabled.

SERR

System Error **Output**

During any slave transaction, the Am79C973/Am79C975 controller asserts SERR when it detects an address parity error, and reporting of the error is enabled by setting PERREN (PCI Command register, bit 6) and SERREN (PCI Command register, bit 8) to 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 Am79C973/Am79C975 controller drives the STOP signal to inform the bus master to stop the current transaction. In bus master mode, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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.

PME

Power Management Event **Output, Open Drain**

PME is an output that can be used to indicate that a power management event (a Magic Packet, an OnNow pattern match, or a change in link state) has been detected. The PME pin is asserted when either

1. PME_STATUS and PME_EN are both 1, or
2. PME_EN_OVR and MPMAT are both 1, or
3. PME_EN_OVR and LCDET are both 1.

The PME signal is asynchronous with respect to the PCI clock.

VAUXDET

Auxiliary Power Detect **Input**

VAUXDET is used to sense the presence of the auxiliary power and correctly report the capability of asserting PME signal in D3 cold. The VAUXDET pin should be connected to the auxiliary power supply or to ground through a resistor. If PCI power is used to power the device, a pull-down resistor is required. For systems that provide auxiliary power, the VAUXDET pin should be tied to auxiliary power through a pull-up resistor.

Board Interface

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

LED0

LED0 **Output**

This output is designed to directly drive an LED. By default, LED0 indicates an active link connection. This pin can also be programmed to indicate other network sta-

tus (see BCR4). The $\overline{\text{LED0}}$ pin polarity is programmable, but by default it is active LOW. When the $\overline{\text{LED0}}$ pin polarity is programmed to active LOW, the output is an open drain driver. When the $\overline{\text{LED0}}$ pin polarity is programmed to active HIGH, the output is a totem pole driver.

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

$\overline{\text{LED1}}$

LED1 **Output**

This output is designed to directly drive an LED. By default, $\overline{\text{LED1}}$ indicates receive activity on the network. This pin can also be programmed to indicate other network status (see BCR5). The $\overline{\text{LED1}}$ pin polarity is programmable, but by default, it is active LOW. When the $\overline{\text{LED1}}$ pin polarity is programmed to active LOW, the output is an open drain driver. When the $\overline{\text{LED1}}$ pin polarity 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 Am79C973/Am79C975 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.

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. This pin can be programmed to indicate various 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 MIIRXFRTGE pin.*

$\overline{\text{LED3}}$

LED3 **Output**

This output is designed to directly drive an LED. By default, $\overline{\text{LED3}}$ indicates transmit activity on the network. This pin can also be programmed to indicate other network status (see BCR7). The $\overline{\text{LED3}}$ pin polarity is programmable, but by default it is active LOW. 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 maybe required between the $\overline{\text{LED3}}$ pin and the LED circuit. If an LED circuit were directly attached to this pin, it may 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 or low current LEDs are used, 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 and MIIRXFRTGD pins.*

PG

Power Good **Input**

The PG pin has two functions: (1) it puts the device into Magic Packet™ mode, and (2) it blocks any resets when the PCI bus power is off.

When PG is LOW and either MPPEN or MPMODE is set to 1, the device enters the Magic Packet mode.

When PG is LOW, a LOW assertion of the PCI $\overline{\text{RST}}$ pin will only cause the PCI interface pins (except for $\overline{\text{PME}}$) to be put in the high impedance state. The internal logic will ignore the assertion of $\overline{\text{RST}}$.

When PG is HIGH, assertion of the PCI $\overline{\text{RST}}$ pin causes the controller logic to be reset and the configuration information to be loaded from the EEPROM.

PG input should be kept high during the NAND tree testing.

RWU

Remote Wake Up **Output**

RWU is an output that is asserted either when the controller is in the Magic Packet mode and a Magic Packet frame has been detected, or the controller is in the Link Change Detect mode and a Link Change has been detected.

This pin can drive the external system management logic that causes the CPU to get out of a low power

mode of operation. This pin is implemented for designs that do not support the PME function.

Three bits that are loaded from the EEPROM into CSR116 can program the characteristics of this pin:

1. RWU_POL determines the polarity of the RWU signal.
2. If RWU_GATE bit is set, RWU is forced to the high impedance state when PG input is LOW.
3. RWU_DRIVER determines whether the output is open drain or totem pole.

The internal power-on-reset signal forces this output into the high impedance state until after the polarity and drive type have been determined.

WUMI

Wake-Up Mode Indicator **Output**

This output, which is capable of driving an LED, is asserted when the device is in Magic Packet mode. It can be used to drive external logic that switches the device power source from the main power supply to an auxiliary power supply.

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 Am79C973/Am79C975 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 2.

EEDI

EEPROM Data In **Output**

This pin is designed to directly interface to a serial EEPROM that uses the 93C46 EEPROM interface protocol. EEDI is connected to the EEPROM's data input pin. It is controlled by either the Am79C973/Am79C975 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 LED0 pin.*

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 Am79C973/Am79C975Am79C973/Am79C975Am79C973/Am79C975 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 LED3 and MIIRXFRTGD pins.*

EESK

EEPROM Serial Clock **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 Am79C973/Am79C975 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 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 Am79C973/Am79C975 controller interface. At the rising edge of the last CLK edge while 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.

WARNING: The input signal level of EESK must be valid for correct EEPROM detection before the deassertion of 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 [19:16] during boot device accesses) is valid on these pins at the beginning of a boot device access, at the rising edge of AS_EBOE. This upper address byte must be stored externally in a D flip-flop. During subsequent cycles of a 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.

Note: *EBUA_EBA[7:5] pins are multiplexed with the TX_ER, PHY_RST, and MDC pins.*

EBDA[15:8]

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

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.

Note: *EBDA[15:8] pins are multiplexed with the TXD[3:0], TX_EN, MDIO, CRS, and COL pins.*

EBD[7:0]

Expansion Bus Data [7:0] Input/Output

The EBD[7:0] pins provide data bits [7:0] for EPROM/FLASH 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.

Note: *EBD[7:0] pins are multiplexed with the RXD[3:0], RX_DV, RX_CLK, RX_ER, and TX_CLK pins.*

$\overline{\text{EROMCS}}$

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.

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 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.

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

$\overline{\text{EBWE}}$

Expansion Bus Write Enable Output

$\overline{\text{EBWE}}$ provides the write enable for write accesses to the Flash device.

EBCLK

Expansion Bus Clock Input

EBCLK may be used as the fundamental clock to drive the Expansion Bus and internal SRAM 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 kW resistor.

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

Media Independent Interface (MII)

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 Am79C973/Am79C975 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 EBD7 pin.*

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[3:0] pins are multiplexed with the EBDA[11:8] pins.*

TX_EN

Transmit Enable Output

TX_EN indicates when the Am79C973/Am79C975 device is presenting valid transmit nibbles on the MII. While TX_EN is asserted, the Am79C973/Am79C975 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 EBDA12 pin.*

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 Am79C973/Am79C975 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.

Note: *The TX_ER pin is multiplexed with the EBUA_EBA7 pin.*

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 EBDA15 pin.*

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 EBDA14 pin.*

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 EBD6 pin.*

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 Time Base Clock.

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

Note: *The MDC pin is multiplexed with the EBUA_EBD5 pin.*

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 Am79C973/Am79C975 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 $\pm$5% resistor. If the PHY is on board, then the MDIO pin should be externally pulled up to V_{CC} with a 10-kΩ $\pm$5% resistor.

Note: *The MDIO pin is multiplexed with the EBDA13 pin.*

PHY_RST

Physical Layer Reset **Output**

PHY_RST is an output pin that is used to reset an external PHY. This output eliminates the need for a fan out buffer for the PCI RST signal, provides polarity for the specific external PHY used, and prevents the resetting of the external PHY when the PG input is LOW. The output polarity is determined by RST_POL bit (CSR 116, bit 0).

Note: *The PHY_RST pin is multiplexed with the EBUA_EBA6 pin.*

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 Am79C973/Am79C975 device. RX_CLK must provide a nibble rate clock (25% of the network data rate). Hence, when the Am79C973/Am79C975 device is operating at 10 Mbps, it provides an RX_CLK frequency of 2.5 MHz, and at 100 Mbps it provides an RX_CLK frequency of 25 MHz.

Note: *The RX_CLK pin is multiplexed with the EBD5 pin.*

RXD[3:0]

Receive Data **Input**

RXD[3:0] is the nibble-wide MII-compatible 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.

Note: *The RXD[3:0] pin is multiplexed with the EBD[3:0] pins.*

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 Am79C973/Am79C975 device, 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.

Note: *The RX_DV pin is multiplexed with the EBD4 pin.*

External Address Detection Interface

$\overline{\text{EAR}}$

External Address Reject Low **Input**

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

The $\overline{\text{EAR}}$ pin ***must not*** be left unconnected, it should be tied to VDD through a 10-kΩ ±5% resistor.

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

SFBD

Start Frame-Byte Delimiter **Output**

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.*

MIIRXFRTGD

MII 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 Snoop mode 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 $\overline{MIIRXFRTGD}$ pin is multiplexed with the EEDO and $\overline{LED3}$ pins.*

MIIRXFRTGE

MII 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 Snoop mode 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 $\overline{LED2}$ pin.*

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 Am79C973/ Am79C975 controller. The pin has an internal pull up resistor.

TDO

Test Data Out **Output**

TDO is the test data output path from the Am79C973/ Am79C975 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.

Network Interfaces

TX+, TX-
Serial Transmit Data
MLT-3/PECL **Output**

These pins are the 10BASE-T/100BASE-X differential drivers. For 100BASE-FX, these transmit outputs carry differential PECL-level NRZI data for direct connection to an external fiber optic transceiver. They can be forced to logical 0 (TX+ low, TX- high) by programming the TX_DISABLE bit (bit 3 of the internal PHY Control/ Status Register, Register 17). For 100BASE-TX, these pins carry MLT-3 data and are connected to the primary side of the magnetics module. For 10BASE-T, these

pins carry the transmit output data and are connected to the transmit side of the magnetics module.

RX+, RX-
Serial Receive Data
MLT-3/PECL **Input**

These pins are the 10BASE-T/100BASE-X port differential receiver pairs. They receive MLT-3 data and are connected to the receive side of the magnetics module in 100BASE-TX operation. They receive PECL NRZI data from an external fiber optic transceiver in 100BASE-FX application. For 10BASE-T, these pins accept the receive input data from the magnetics module.

SDI+, SDI-
Signal Detect **Input**

These pins control the selection between PECL and MLT-3 data for the TX± and RX± pins. For 100BASE-TX or 10BASE-T, both of these pins may be tied to ground or left floating. This enables transmission and reception of MLT-3 or 10BASE-T signals at the TX± and RX± pins. For 100BASE-FX, these pins are biased at PECL levels. They are connected to the SDI pin from the optical transceiver module to indicate whether the received signal is above the required threshold. If signal detect is not available, these pins should be tied to a PECL logical 1 (SDI+ = PECL 1, SDI- = PECL 0). See Table 2.

Table 2. SDI± Settings for Transceiver Operation

SDI+	SDI-	Port Mode
TTL LOW (<0.8 V)	TTL LOW (<0.8 V)	MLT-3/10BASE-T Mode
TTL HIGH (>2.0 V)	TTL HIGH (>2.0 V)	PECL Mode

IREF

Internal Current Reference **Input**

This pin serves as a current reference for the integrated 10/100 PHY. It must be connected to ground via a 12 kΩ resistor (1%).

Clock Interface

XTAL1

Crystal Input **Input**

The internal clock generator uses a 25-MHz (50 ppm-100 ppm) crystal that is attached to the XTAL1 and XTAL2 pins. XTAL1 may alternatively be driven using an external 25 MHz (50 ppm - 100 ppm) CMOS-level clock signal when XTAL2 is left floating. The XTAL1 pin is not 5 V tolerant and must only be driven by a 3.3 V clock source.

XTAL2

Crystal Output **Output**

The internal clock generator uses a 25 MHz (50 ppm - 100 ppm) crystal that is attached to the pins XTAL1 and XTAL2. XTAL1 may alternatively be driven using an external 25 MHz (50 ppm - 100 ppm) CMOS-level clock signal when XTAL2 is left floating.

XCLK/$\overline{\text{XTAL}}$

External Clock/Crystal Select **Input**

When HIGH, an External Clock Source is selected bypassing the Crystal circuit. When LOW, a Crystal is used instead. The following table illustrates how this pin works.

Input Pin	Output Pin	XCLK/$\overline{\text{XTAL}}$	Clock Source
XATL1	XTAL2	0	Crystal
XTAL1	Don't Care	1	Oscillator/ External CLK Source

Serial Management Interface (SMI) (Am79C975 only)

MCLOCK

SMI Clock **Input/Output**

MCLOCK is the clock pin of the serial management interface. MCLOCK is typically driven by an external I^2C/SMBus master. The Am79C975 controller will drive the clock line low in order to insert wait states before it starts sending out data in response to a read. The frequency of the clock signal can vary between 10 kHz and 100 kHz, and it can change from cycle to cycle.

Note: *MCLOCK is also capable of running at a frequency as high as 2.5 MHz to allow for shorter production test time.*

MDATA

SMI Data **Input/Output**

MDATA is the data pin of the serial management interface. MDATA can be driven by an external I^2C/SMBus master or by the Am79C975 controller. The interface protocol defines exactly at what time the Am79C975 controller has to listen to the MDATA pin and at what time the controller must drive the pin.

$\overline{\text{MIRQ}}$

SMI Interrupt **Output**

$\overline{\text{MIRQ}}$ is an asynchronous attention signal that the Am79C975 controller provides to indicate that a management frame has been transmitted or received. The assertion of the $\overline{\text{MIRQ}}$ signal can be controlled by a global mask bit (MIRQEN) or individual mask bits

(MRX_DONEM, MTX_DONEM) located in the Command register.

Note: *The SMI interrupt acknowledge does not follow the SMBus alert protocol, but simply requires clearing the interrupt bit.*

Power Supply

VDDB

I/O Buffer Power (6 Pins) **+3.3 V Power**

There are seven power supply pins that are used by the input/output buffer drivers. All VDDB pins must be connected to a +3.3 V supply.

VDD_PCI

PCI I/O Buffer Power (9 Pins) **+3.3 V Power**

There are nine power supply pins that are used by the PCI input/output buffer drivers (except $\overline{\text{PME}}$ driver). All VDD_PCI pins must be connected to a +3.3 V supply.

VSSB

I/O Buffer Ground (17 Pins)
Ground

There are 17 ground pins that are used by the input/output buffer drivers.

VDD

Digital Power (6 Pins) **+3.3 V Power**

There are six power supply pins that are used by the internal digital circuitry. All VDD pins must be connected to a +3.3 V supply.

VSS

Digital Ground (8 Pins) **Ground**

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

DVDDD, DVDDP

PDX Block Power **+3.3 V Power**

These pins supply power to the 100 Mbps Physical Data Transceiver (PDX) block. They must be connected to a +3.3 V ±5% source. These pins require careful decoupling to ensure proper device performance.

DVDDRX, DVDDTX

I/O Buffer Power **+3.3 V Power**

These pins supply power to the MLT-3/PECL/10BASE-T input/output buffers. They also supply the MLT-3 circuits (equalizer, etc.) of the network port. They must be connected to a +3.3 V ±5% source. These pins require careful decoupling to ensure proper device performance.

DVDDA

Analog PLL Power **+3.3 V Power**

This pin supplies power to the IREF current reference circuit and the 10BASE-T analog PLL. They must be connected to a +3.3 V ±5% source. These pins require careful decoupling to ensure proper device performance.

DVSSX

All Blocks **Ground**

These pins are the ground connection for all blocks of the device except the PDX block. They must be directly connected to the common external ground plane.

DVSSD, DVSSP

PDX Ground **Ground**

These pins are the ground connection for the Physical Data Transceiver (PDX) block. They must be directly connected to the common external ground plane.

DVDDCO

Crystal **+3.3 V Power**

This pin supplies the power to the Crystal circuit.

BASIC FUNCTIONS

System Bus Interface

The Am79C973/Am79C975 controllers are designed to operate as a bus master during normal operations. Some slave I/O accesses to the Am79C973/Am79C975 controllers are required in normal operations as well. Initialization of the Am79C973/Am79C975 controllers are 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 Am79C973/Am79C975 controllers. 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controllers support 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 Am79C973/Am79C975 controllers are divided into three parts. One part is the PCI configuration registers used to identify the Am79C973/Am79C975 controllers 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controllers. The Am79C973/Am79C975 controllers occupy 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 Am79C973/Am79C975 controller operating mode, to enable and disable various features, to monitor operating status, and to request particular functions to be executed by the Am79C973/Am79C975 controllers.

The third portion of the software interface is the descriptor and buffer areas that are shared between the software and the Am79C973/Am79C975 controllers 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 Am79C973/Am79C975 controllers 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 Am79C973/Am79C975 controllers provide all of the PHY layer functions for 10 Mbps (10BASE-T) or 100 Mbps (100BASE-TX). It also provides a Pseudo ECL (PECL) interface for 100BASE-FX fiber networks. The Am79C973/Am79C975 controllers support both half-duplex and full-duplex operation on network interfaces.

Serial Management Interface (Am79C975)

The Am79C975 controller provides a 3-pin interface based on the I^2C and SMBus standards that enables a system to monitor the status of the system hardware and report the results to the management station or system administrator. Monitored information may include critical system parameters, such as voltage, temperature, and fan speed, as well as system management events, such as chassis intrusion, operating system errors and power-on failures.

MII Interface

The Am79C973/Am79C975 supports an MII interface mode that makes the device operate like a PCnet-FAST+ device. The MII pins are multiplexed with the expansion bus pins, which means the device will only support either an EPROM/Flash or an external PHY but not both. To enter this mode, set PHYSELEN (BCR2, bit 13) = 1 and PHYSEL (BCR18, bit 4 and bit 3) = 10. This mode isolates the internal PHY to allow interface with an external PHY. For a more detailed description of the MII interface including timing diagrams see Appendix C. Refer to the connection diagram to see how the pins are multiplexed.

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 3 shows the response of the Am79C973/Am79C975 controllers to each of the PCI commands in slave mode.

Table 3. 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 Am79C973/Am79C975 PCI configuration space with a configuration read or write command. The Am79C973/Am79C975 controllers 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 Am79C973/Am79C975 controllers ignore 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 Am79C973/Am79C975 controllers will assert TRDY on the third clock of the data phase.

The Am79C973/Am79C975 controllers do not support burst transfers for access to configuration space. When the host keeps FRAME asserted for a second data phase, the Am79C973/Am79C975 controllers 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 Am79C973/Am79C975 controllers will terminate the access on the PCI bus with a disconnect/retry response.

The Am79C973/Am79C975 controllers support 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 Am79C973/Am79C975 controllers are 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 Am79C973/Am79C975 controllers assert DEVSEL on the second clock after FRAME is asserted (medium timing).

Slave I/O Transfers

After the Am79C973/Am79C975 controllers are 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 Am79C973/Am79C975 controllers will look for an address that falls within its 32 bytes of I/O address space (starting from the I/O base address). The Am79C973/Am79C975 controllers assert DEVSEL if it detects an address match and the access is an I/O cycle. If configured for memory mapped I/O mode, the Am79C973/Am79C975 controllers 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

Am79C973/Am79C975 controllers assert 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.

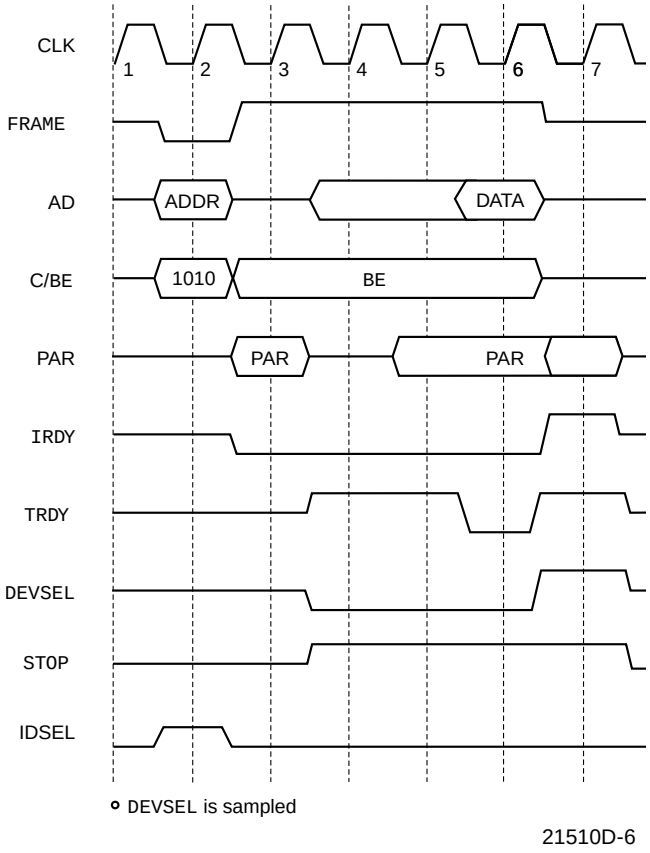


Figure 1. Slave Configuration Read

The Am79C973/Am79C975 controllers will not assert DEVSEL if it detects an address match and the PCI command is not of the correct type. In memory mapped I/O mode, the Am79C973/Am79C975 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 Am79C973/Am79C975 controllers decode 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 Am79C973/Am79C975 controllers are 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 Am79C973/Am79C975 controllers do not support burst transfers for access to its I/O resources. When the host keeps FRAME asserted for a second data phase, the Am79C973/Am79C975 controllers will disconnect the transfer.

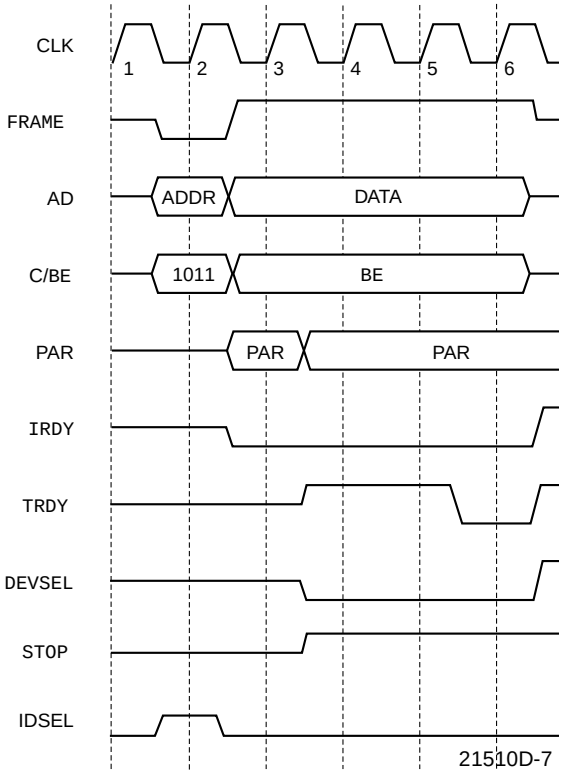


Figure 2. Slave Configuration Write

The Am79C973/Am79C975 controllers support 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 Am79C973/Am79C975 controllers are 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 Am79C973/Am79C975 controllers assert DEVSEL on the second clock after FRAME is asserted (medium timing) See Figure 3 and Figure 4.

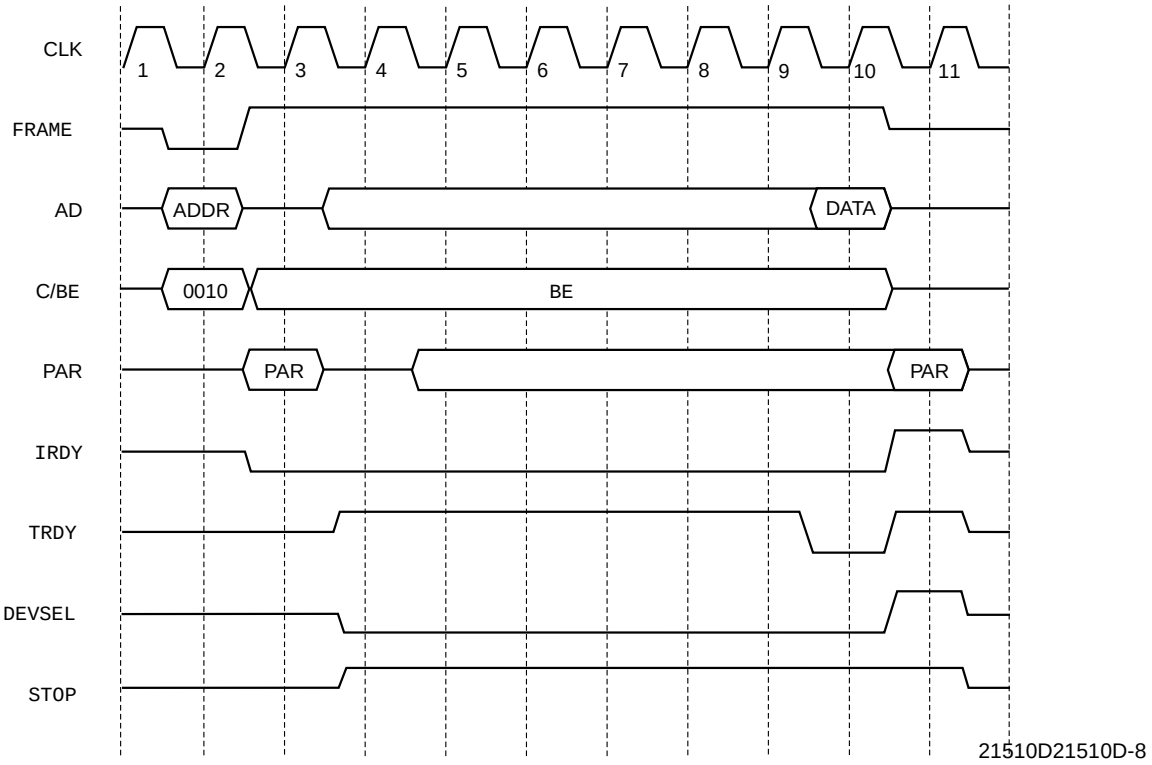


Figure 3. Slave Read Using I/O Command

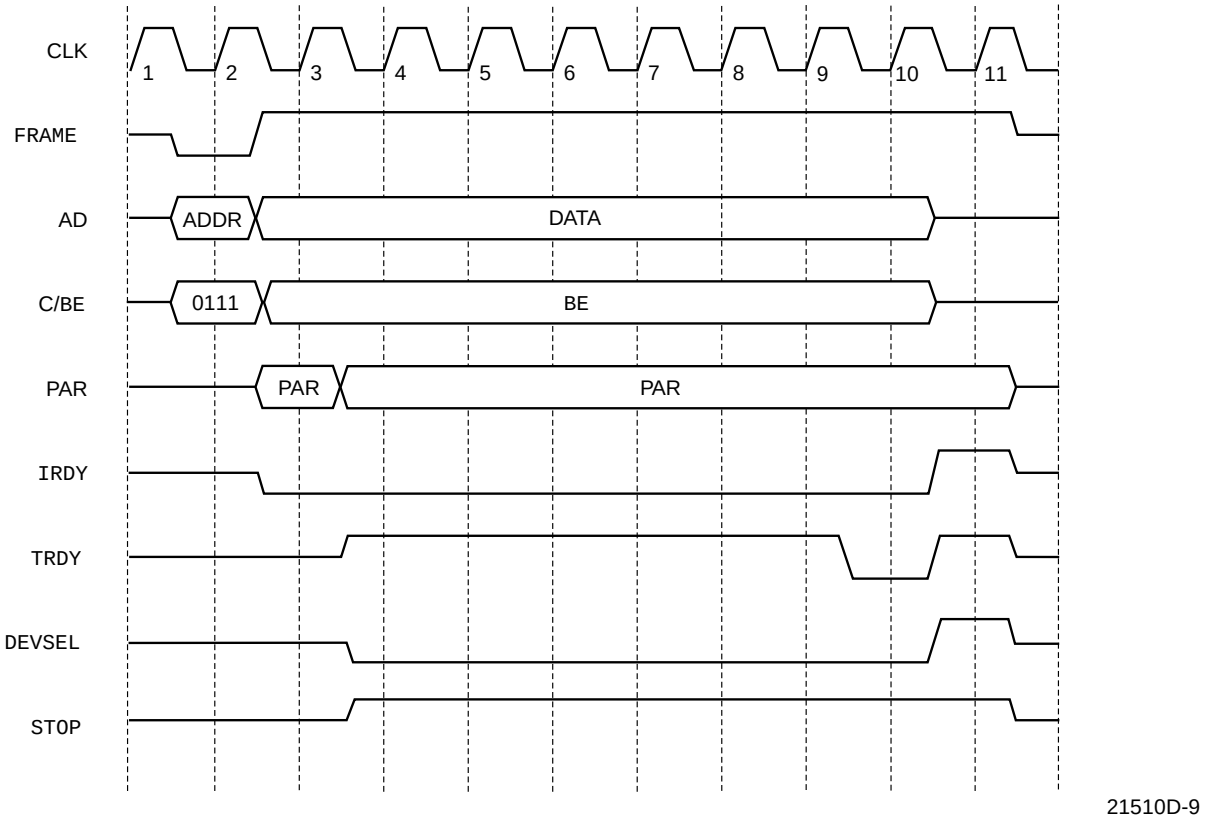


Figure 4. Slave Write Using Memory Command

Expansion ROM Transfers

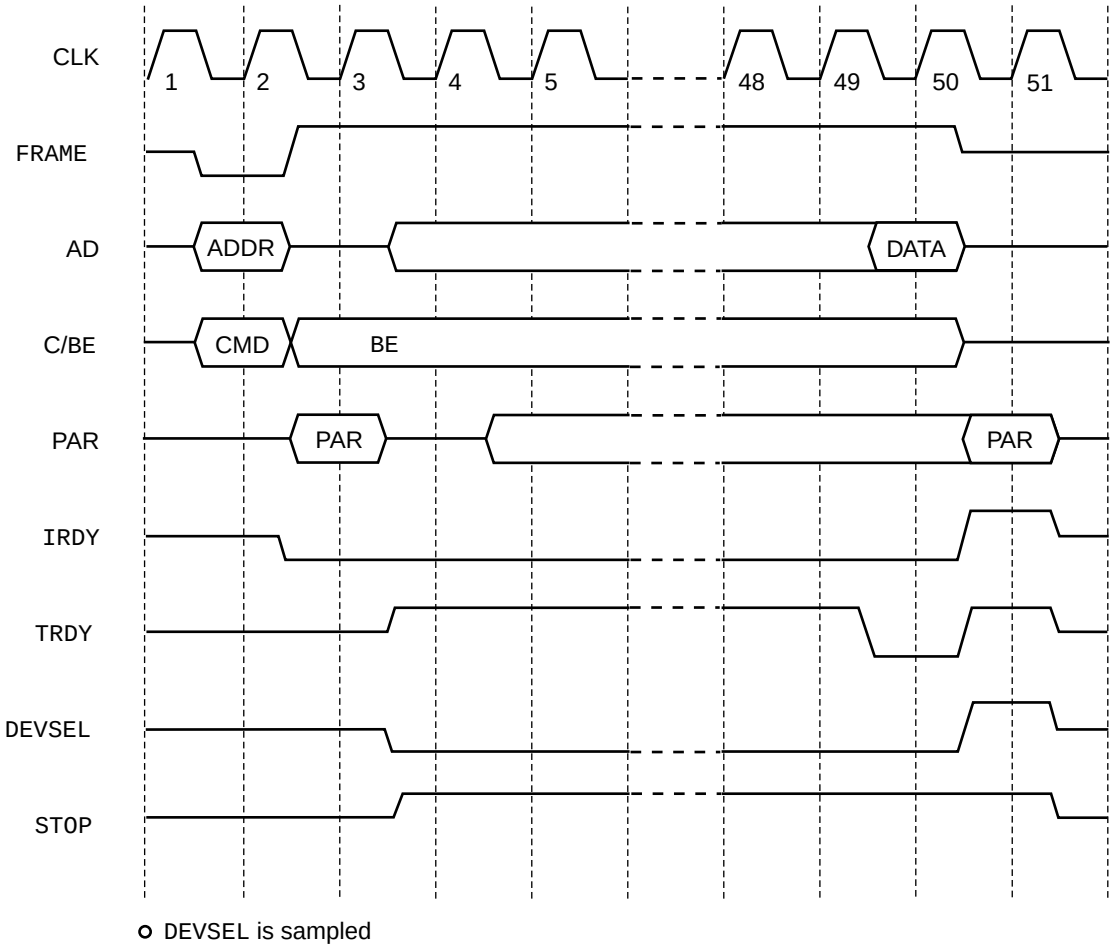
The host must initialize the Expansion ROM Base Address register at offset 30H in the PCI configuration space with a valid address before enabling the access to the device. The Am79C973/Am79C975 controllers will not react to any access to the Expansion ROM until both MEMEN (PCI Command register, bit 1) and ROMEN (PCI Expansion ROM Base Address register, bit 0) are set to 1. After the Expansion ROM is enabled, the Am79C973/Am79C975 controllers will assert $\overline{\text{DEVSEL}}$ on all memory read accesses with an address between ROMBASE and ROMBASE + 1M - 4. The Am79C973/Am79C975 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 Am79C973/Am79C975 controllers from claiming any memory cycles not intended for it.

The Am79C973/Am79C975 controllers will always read four bytes for every host Expansion ROM read access. $\overline{\text{TRDY}}$ will not be asserted until all four bytes are loaded into an internal scratch register. The cycle $\overline{\text{TRDY}}$ 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 Am79C973/Am79C975 controllers will claim the cycle by asserting $\overline{\text{DEVSEL}}$. $\overline{\text{TRDY}}$ 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. See Figure 5.



21510D-10

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).

Slave Cycle Termination

There are three scenarios besides normal completion of a transaction where the Am79C973/Am79C975 controllers are the target of a slave cycle and it will terminate the access.

Disconnect When Busy

The Am79C973/Am79C975 controllers cannot service any slave access while it is reading the contents of the EEPROM. Simultaneous access is not allowed in order to avoid conflicts, since the EEPROM is used to initialize some of the PCI configuration space locations and most of the BCRs and CSR116. 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 Am79C973/Am79C975 controllers 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 Am79C973/Am79C975 controllers 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 ms in length, all further slave accesses will be deferred until the internal reset operation is completed. See Figure 6.

Disconnect Of Burst Transfer

The Am79C973/Am79C975 controllers do 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 Am79C973/Am79C975 controllers see $\overline{FRAME}$ 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.

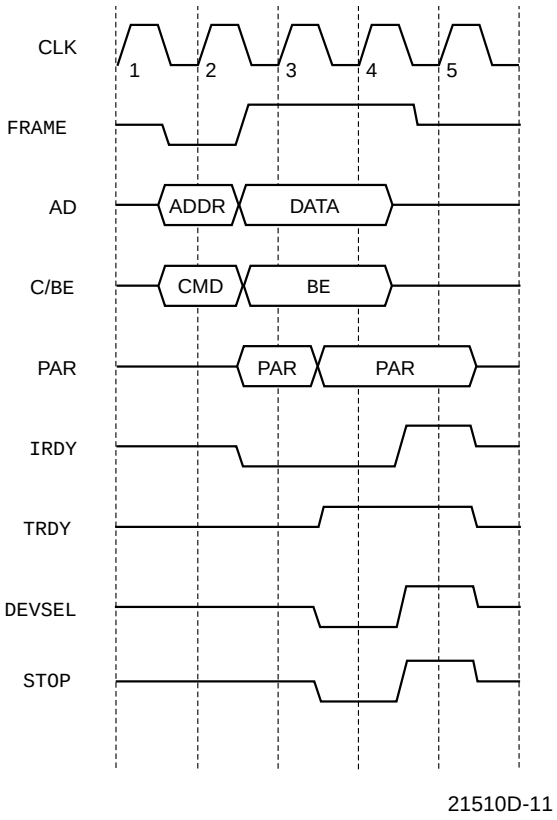


Figure 6. Disconnect Of Slave Cycle When Busy

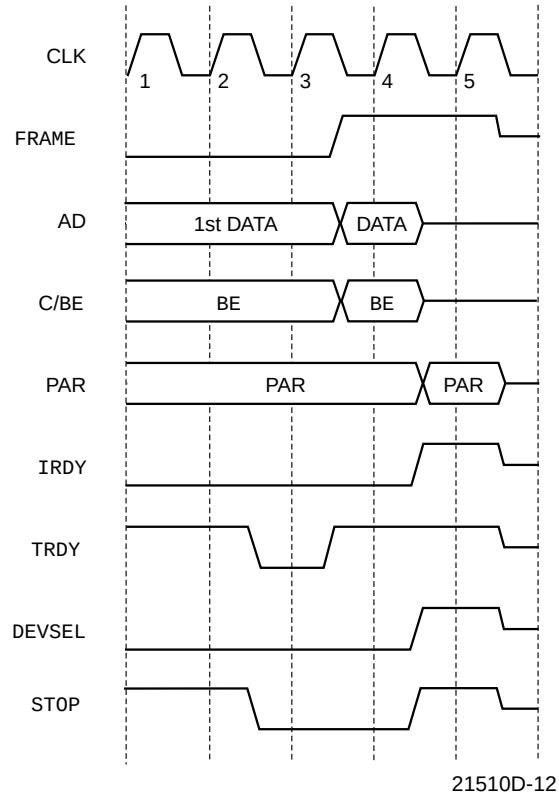


Figure 7. Disconnect Of Slave Burst Transfer - No Host Wait States

If the host is not yet ready when the Am79C973/Am79C975 controller asserts TRDY, the device will wait for the host to assert IRDY. When the host asserts IRDY and FRAME is still asserted, the Am79C973/Am79C975 controller will finish the first data phase by deasserting TRDY one clock later. At the same time, it will assert STOP to signal a disconnect to the host. STOP will stay asserted until the host removes FRAME. See Figure 8.

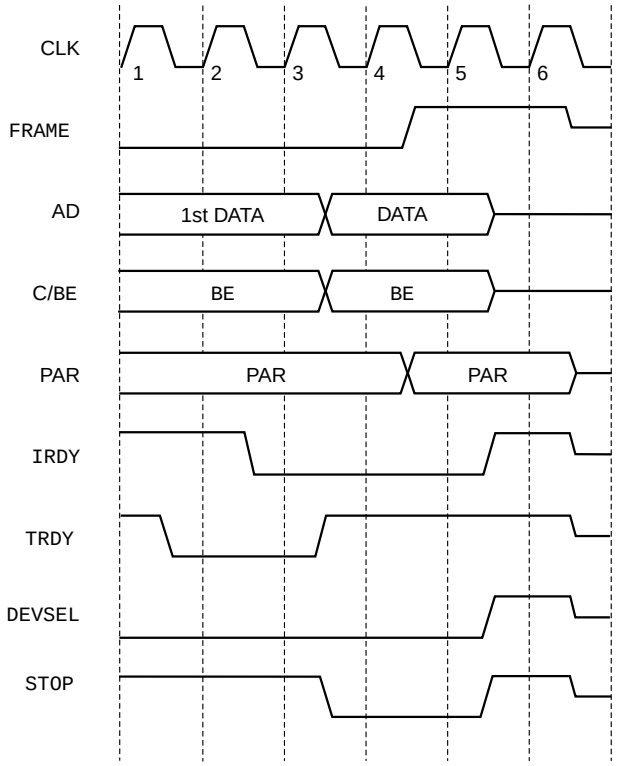


Figure 8. Disconnect Of Slave Burst Transfer - Host Inserts Wait States

Parity Error Response

When the Am79C973/Am79C975 controller 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 Am79C973/Am79C975 controller 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 Am79C973/Am79C975 controller 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.

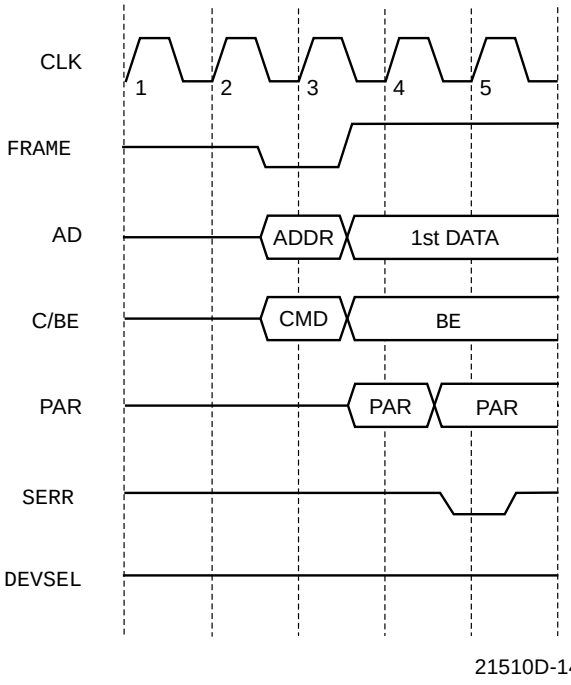


Figure 9. Address Parity Error Response

During the data phase of an I/O write, memory-mapped I/O write, or configuration write command that selects the Am79C973/Am79C975 controller 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) to 1 even when PERREN is cleared to 0. The Am79C973/Am79C975 controller 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 is driven low, the Am79C973/Am79C975 controller drives PERR high for one clock cycle, since PERR is a sustained tri-state signal.

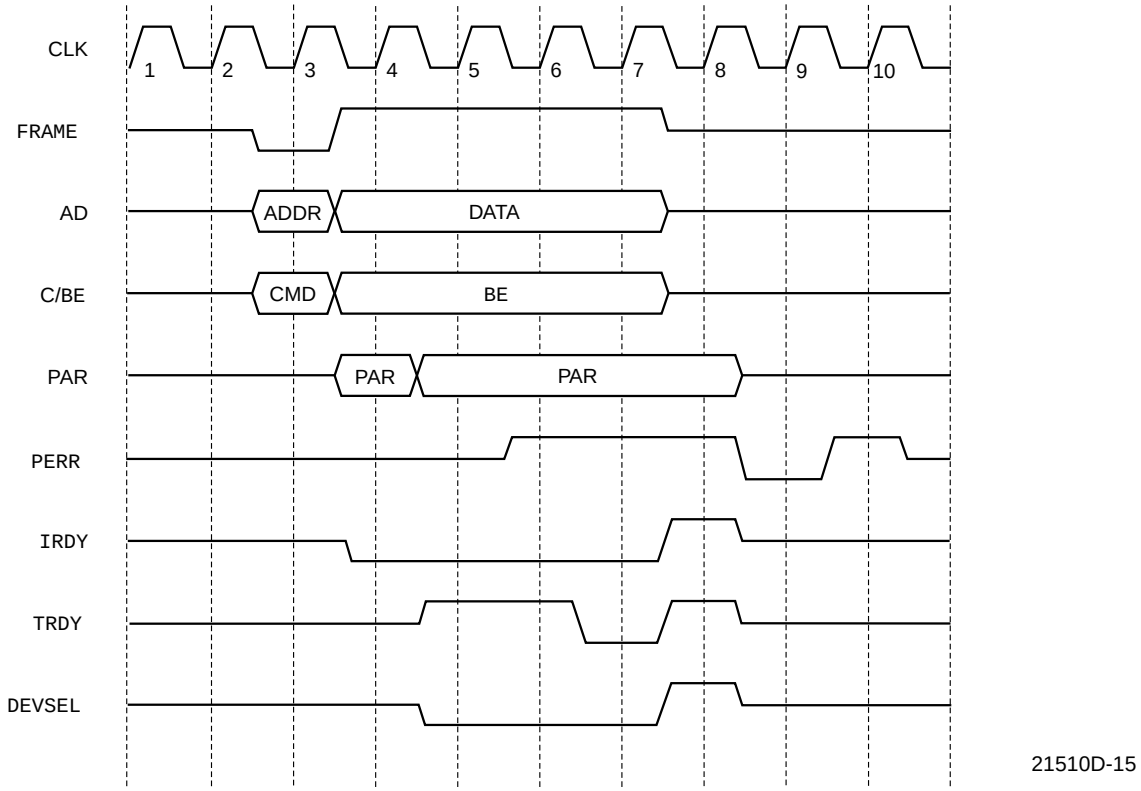


Figure 10. Slave Cycle Data Parity Error Response

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 4 shows the usage of PCI commands by the Am79C973/Am79C975 controller in master mode.

Table 4. 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 Am79C973/Am79C975 microcode will determine when a DMA transfer should be initiated. The first step in any Am79C973/Am79C975 bus master transfer is to acquire ownership of the bus. This task is handled by synchronous logic within the BIU. Bus ownership is requested with the $\overline{REQ}$ signal and ownership is granted by the arbiter through the $\overline{GNT}$ signal.

Figure 11 shows the Am79C973/Am79C975 controller bus acquisition. $\overline{REQ}$ is asserted and the arbiter returns $\overline{GNT}$ while another bus master is transferring data. The Am79C973/Am79C975 controller waits until the bus is idle ($\overline{FRAME}$ and $\overline{IRDY}$ deasserted) before it

starts driving AD[31:0] and C/$\overline{BE}$[3:0] on clock 5. $\overline{FRAME}$ is asserted at clock 5 indicating a valid address and command on AD[31:0] and C/$\overline{BE}$[3:0].

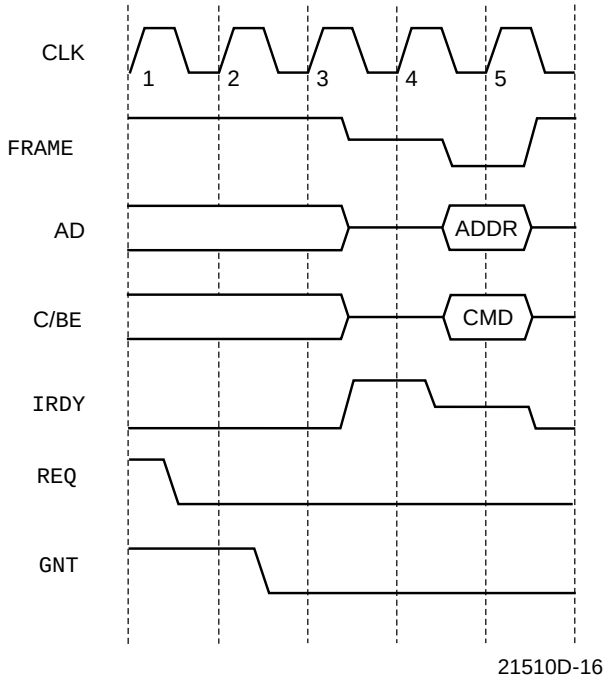


Figure 11. Bus Acquisition

In burst mode, the deassertion of $\overline{REQ}$ depends on the setting of EXTREQ (BCR18, bit 8). If EXTREQ is cleared to 0, $\overline{REQ}$ is deasserted at the same time as $\overline{FRAME}$ is asserted. (The Am79C973/Am79C975 controller never performs more than one burst transaction within a single bus mastership period.) If EXTREQ is set to 1, the Am79C973/Am79C975 controller does not deassert $\overline{REQ}$ until it starts the last data phase of the transaction. Once asserted, $\overline{REQ}$ remains active until $\overline{GNT}$ 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 $\overline{REQ}$ to go inactive immediately.

Bus Master DMA Transfers

There are four primary types of DMA transfers. The Am79C973/Am79C975 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 Am79C973/Am79C975 controller uses non-burst cycles in all bus master read operations. All Am79C973/Am79C975 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 Am79C973/Am79C975 controller will internally discard unneeded bytes.

The Am79C973/Am79C975 controller typically performs more than one non-burst read transaction within a single bus mastership period. $\overline{FRAME}$ is dropped between consecutive non-burst read cycles. $\overline{REQ}$ however stays asserted until $\overline{FRAME}$ is asserted for the last transaction. The Am79C973/Am79C975 controller supports zero wait state read cycles. It asserts $\overline{IRDY}$ immediately after the address phase and at the same time starts sampling $\overline{DEVSEL}$. 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 $\overline{TRDY}$ one clock later.

Basic Burst Read Transfer

The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will internally discard unneeded bytes.

The Am79C973/Am79C975 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 Am79C973/Am79C975 controller supports zero wait state read cycles. It asserts $\overline{IRDY}$ immediately after the address phase and at the same time starts sampling $\overline{DEVSEL}$. $\overline{FRAME}$ is deasserted when the next to last data phase is completed.

Figure 13 shows a typical burst read access. The Am79C973/Am79C975 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 each access by one wait state. The example assumes that EXTREQ (BCR18, bit 8) is cleared to 0, therefore, $\overline{REQ}$ is deasserted in the same cycle as $\overline{FRAME}$ is asserted.

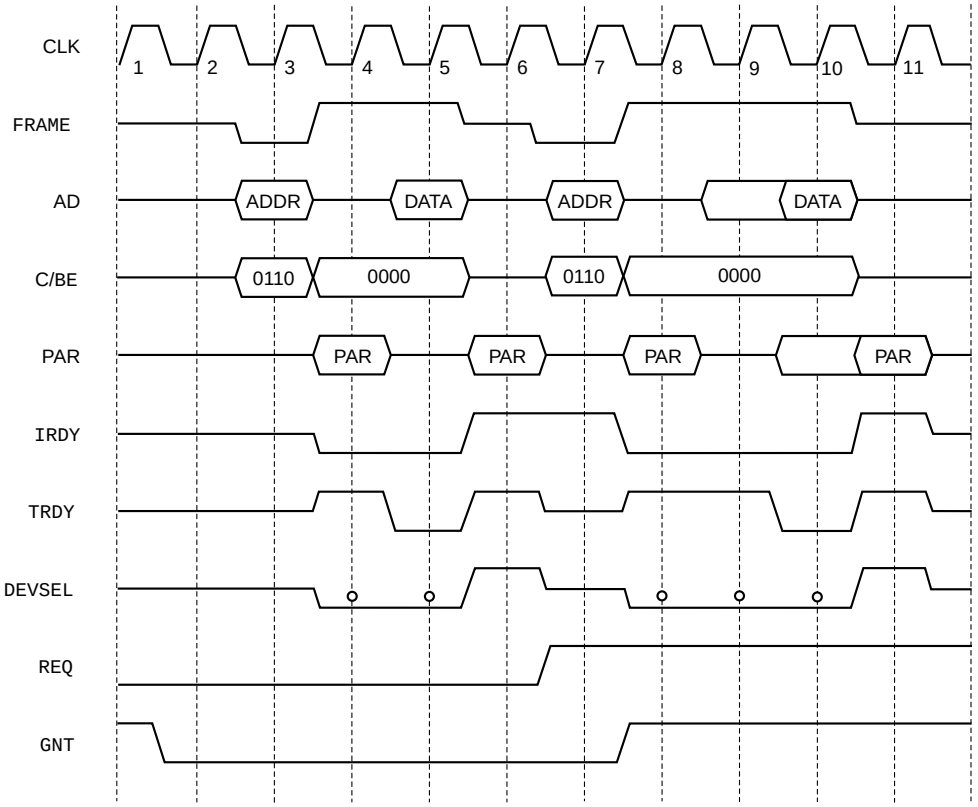


Figure 12. Non-Burst Read Transfer

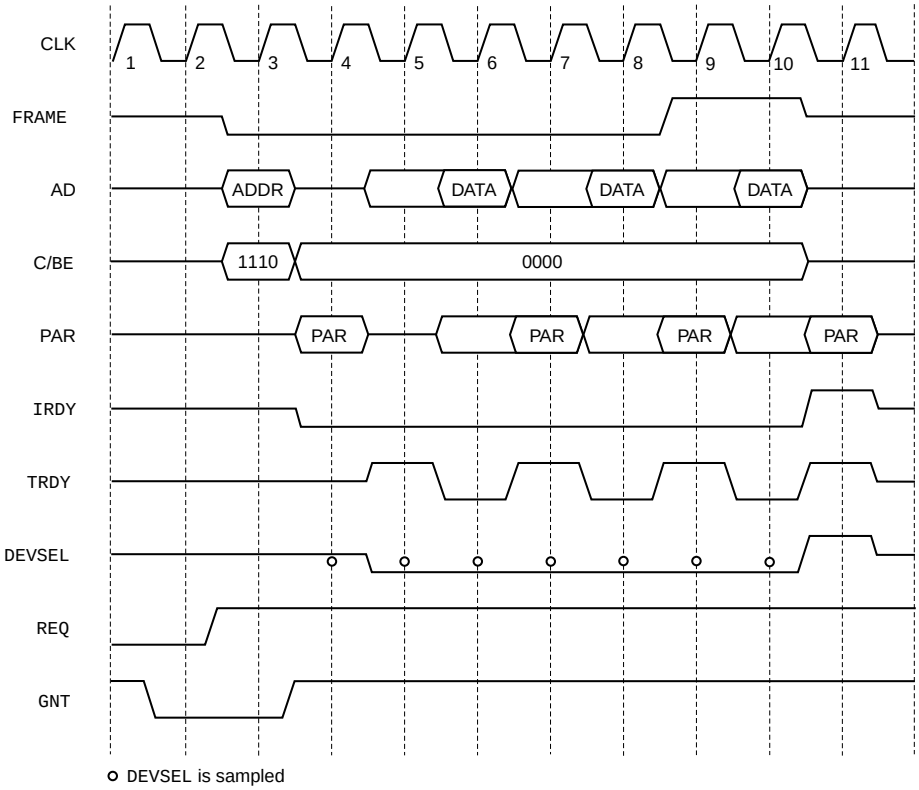


Figure 13. Burst Read Transfer (EXTREQ = 0, MEMCMD = 0)

Basic Non-Burst Write Transfer

By default, the Am79C973/Am79C975 controller uses non-burst cycles in all bus master write operations. All Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller asserts IRDY.

Basic Burst Write Transfer

The Am79C973/Am79C975 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 Am79C973/Am79C975 controller must also be programmed to use SWSTYLE 3 (BCR20, bits 7-0). All Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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.

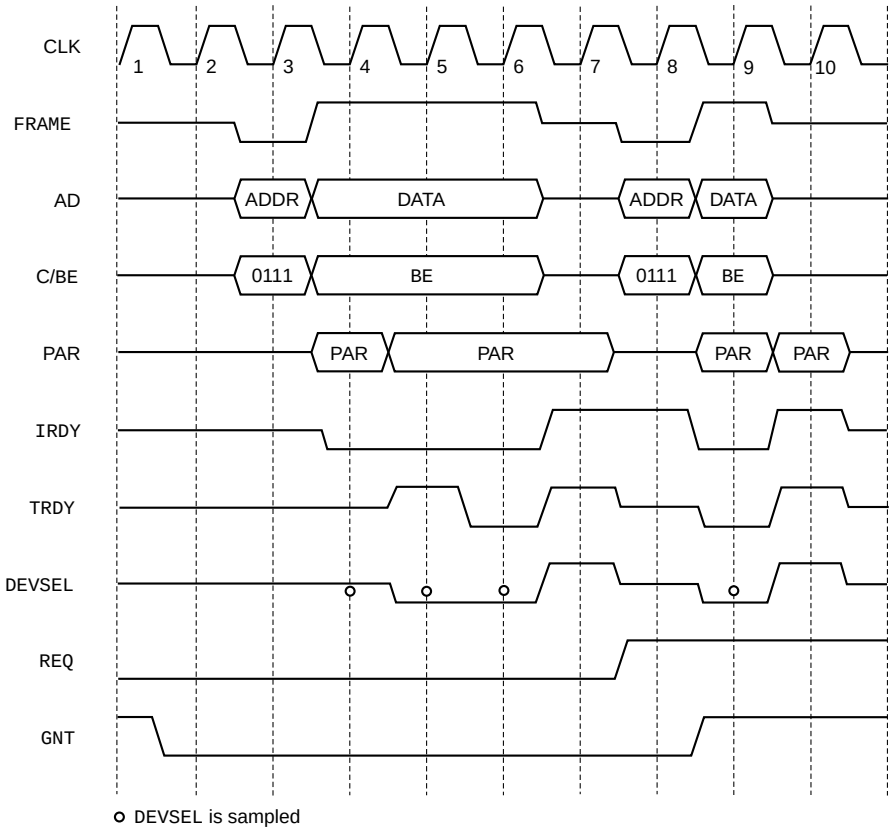


Figure 14. Non-Burst Write Transfer

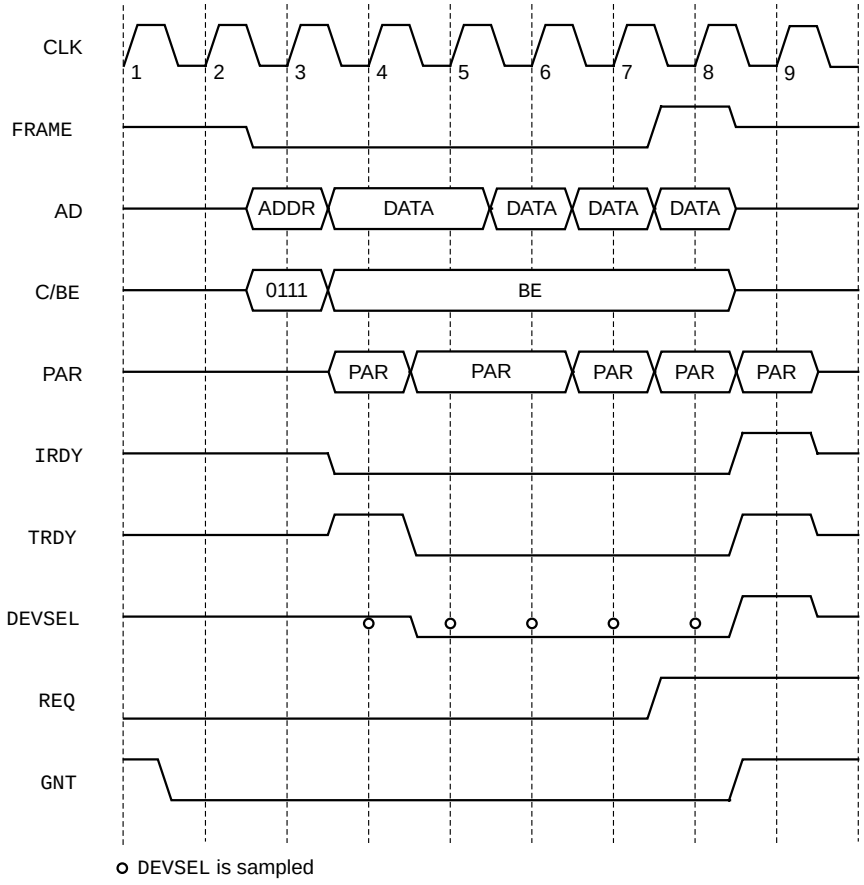
Figure 15 shows a typical burst write access. The Am79C973/Am79C975 controller arbitrates for the bus, is granted access, and writes four 32-bit words (DWords) to the system memory and then releases the bus. In this example, the memory system extends the data phase of the first access by one wait state. The following three data phases take one clock cycle each, which is determined by the timing of $\overline{\text{TRDY}}$. The example assumes that EXTREQ (BCR18, bit 8) is set to 1, therefore, $\overline{\text{REQ}}$ is not deasserted until the next to last data phase is finished.

Target Initiated Termination

When the Am79C973/Am79C975 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, disconnect without data transfer, and target abort.

Disconnect With Data Transfer

Figure 16 shows a disconnection in which one last data transfer occurs after the target asserted $\overline{\text{STOP}}$. $\overline{\text{STOP}}$ is asserted on clock 4 to start the termination sequence. Data is still transferred during this cycle, since both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. The Am79C973/Am79C975 controller terminates the current transfer with the deassertion of $\overline{\text{FRAME}}$ on clock 5 and of $\overline{\text{IRDY}}$ one clock later. It finally releases the bus on clock 7. The Am79C973/Am79C975 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.



21510D-20

Figure 15. Burst Write Transfer (EXTREQ = 1)

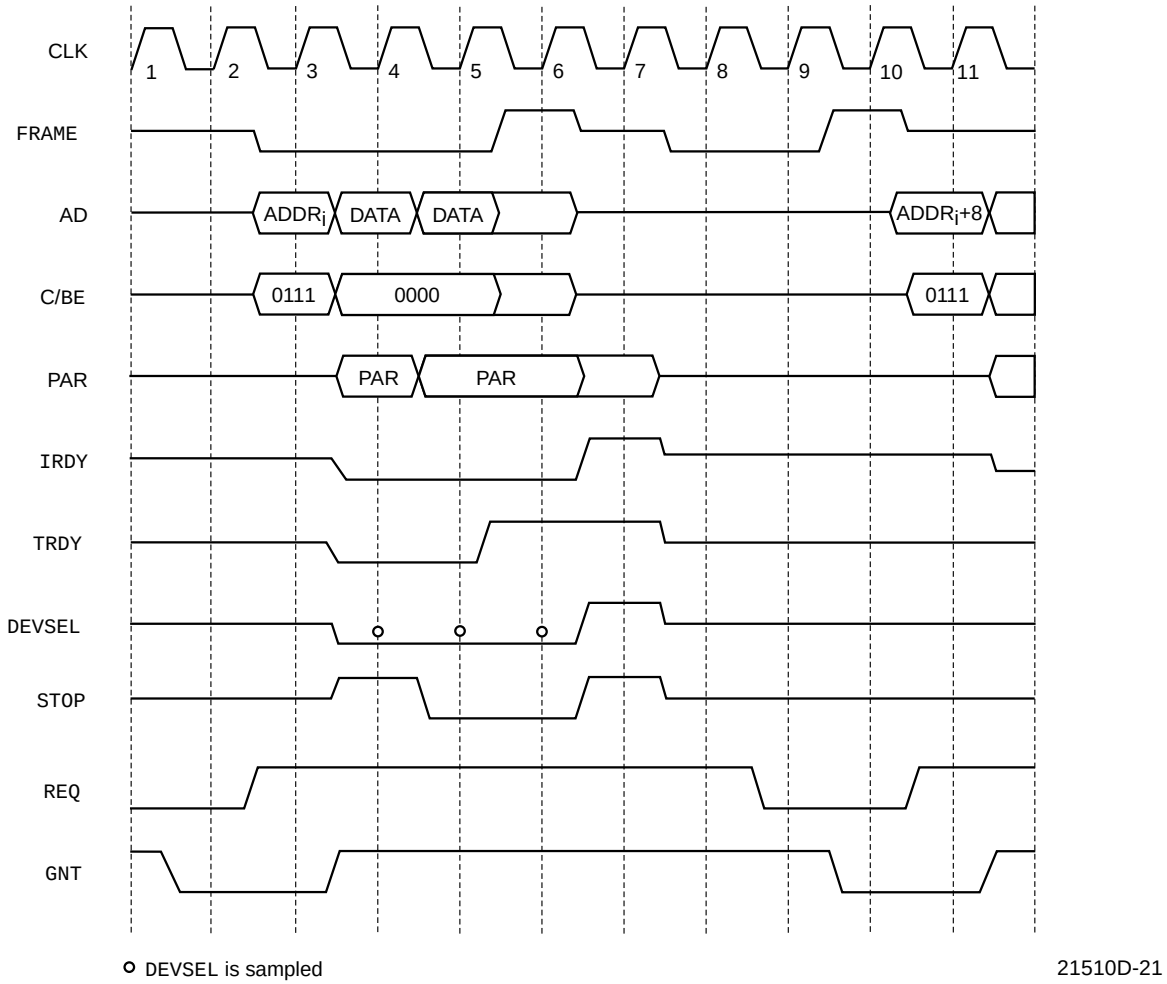


Figure 16. Disconnect With Data Transfer

Disconnect Without Data Transfer

Figure 17 shows a target disconnect sequence during which no data is transferred. $\overline{\text{STOP}}$ is asserted on clock 4 without $\overline{\text{TRDY}}$ being asserted at the same time. The Am79C973/Am79C975 controller terminates the access with the deassertion of $\overline{\text{FRAME}}$ on clock 5 and of $\overline{\text{IRDY}}$ one clock cycle later. It finally releases the bus on clock 7. The Am79C973/Am79C975 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 $\overline{\text{DEVSEL}}$ for one clock. It then deasserts $\overline{\text{DEVSEL}}$ and asserts $\overline{\text{STOP}}$ 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 Am79C973/Am79C975 controller cannot make any assumption about the success of the previous data transfers in the current transaction. The Am79C973/Am79C975 controller terminates the current transfer with the deassertion of $\overline{\text{FRAME}}$ on clock 5 and of $\overline{\text{IRDY}}$ one clock cycle later. It finally releases the bus on clock 6.

Since data integrity is not guaranteed, the Am79C973/Am79C975 controller cannot recover from a target abort event. The Am79C973/Am79C975 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.

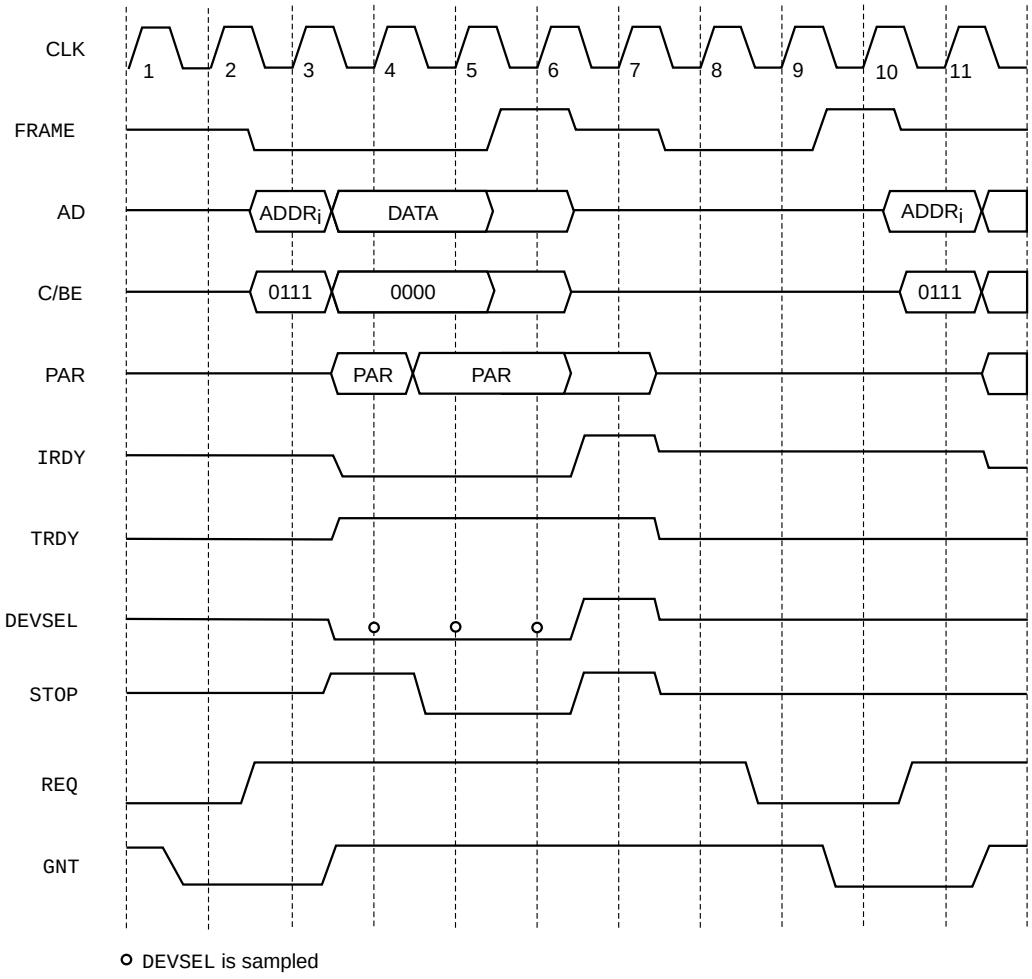


Figure 17. Disconnect Without Data Transfer

RTABORT (PCI Status register, bit 12) will be set to indicate that the Am79C973/Am79C975 controller has received a target abort. In addition, SINT (CSR5, bit 11) will be set to 1. When SINT is set, $\overline{INTA}$ 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 Am79C973/Am79C975 controller will terminate the cycles it produces on the PCI bus.

Preemption During Non-Burst Transaction

When the Am79C973/Am79C975 controller performs multiple non-burst transactions, it keeps $\overline{REQ}$ asserted until the assertion of $\overline{FRAME}$ for the last transaction. When $\overline{GNT}$ is removed, the Am79C973/Am79C975 controller will finish the current transaction and then release the bus. If it is not the last transaction, $\overline{REQ}$ will remain asserted to regain bus ownership as soon as possible. See Figure 19.

Preemption During Burst Transaction

When the Am79C973/Am79C975 controller operates in burst mode, it only performs a single transaction per bus mastership period, where *transaction* is defined as one address phase and one or multiple data phases. The central arbiter can remove $\overline{GNT}$ at any time during the transaction. The Am79C973/Am79C975 controller will ignore the deassertion of $\overline{GNT}$ and continue with data transfers, as long as the PCI Latency Timer is not expired. When the Latency Timer is 0 and $\overline{GNT}$ is deasserted, the Am79C973/Am79C975 controller will finish the current data phase, deassert $\overline{FRAME}$, finish the last data phase, and release the bus. If EXTREQ (BCR18, bit 8) is cleared to 0, it will immediately assert $\overline{REQ}$ to regain bus ownership as soon as possible. If EXTREQ is set to 1, $\overline{REQ}$ will stay asserted.

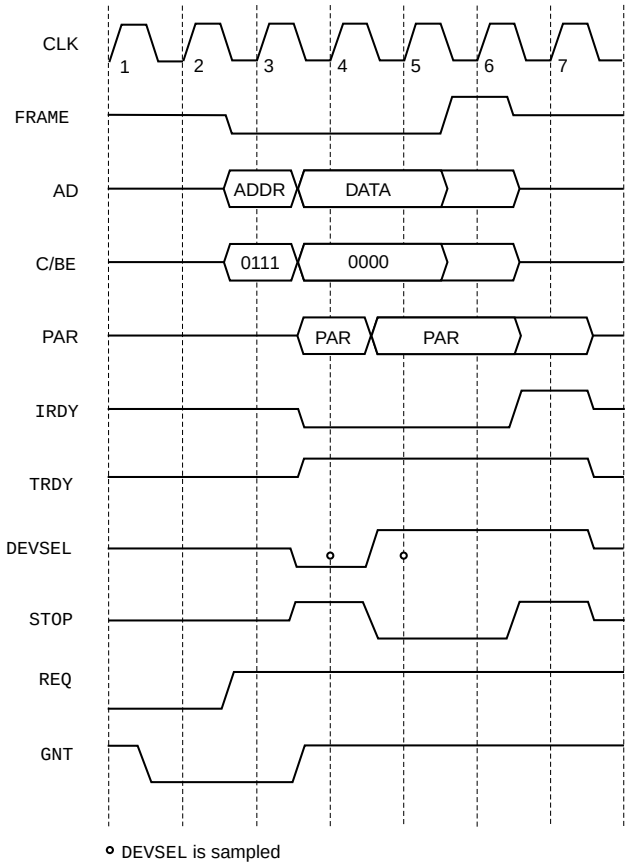


Figure 18. Target Abort

When the preemption occurs after the counter has counted down to 0, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller. The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 set 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller also sets DATAPERR (PCI Status register, bit 8) to 1.

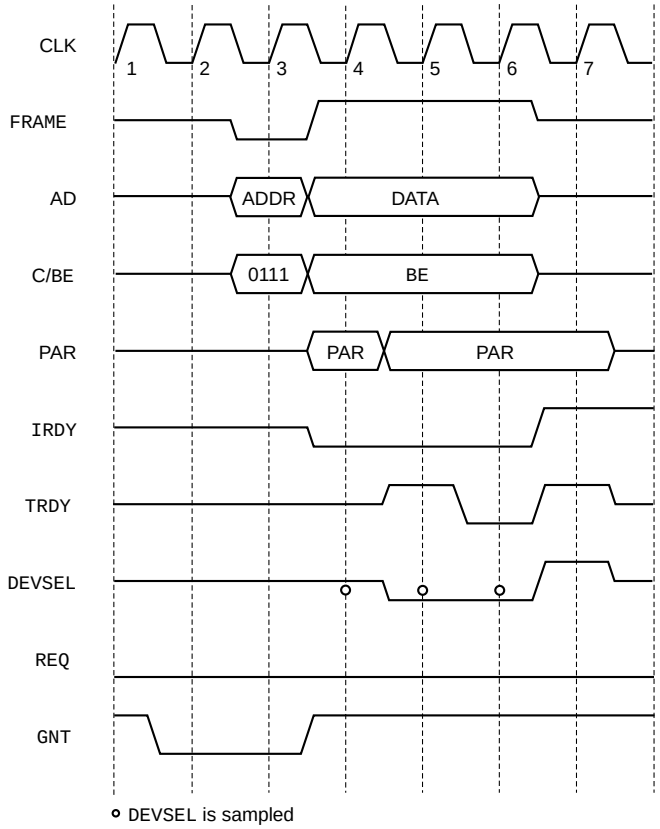


Figure 19. Preemption During Non-Burst Transaction

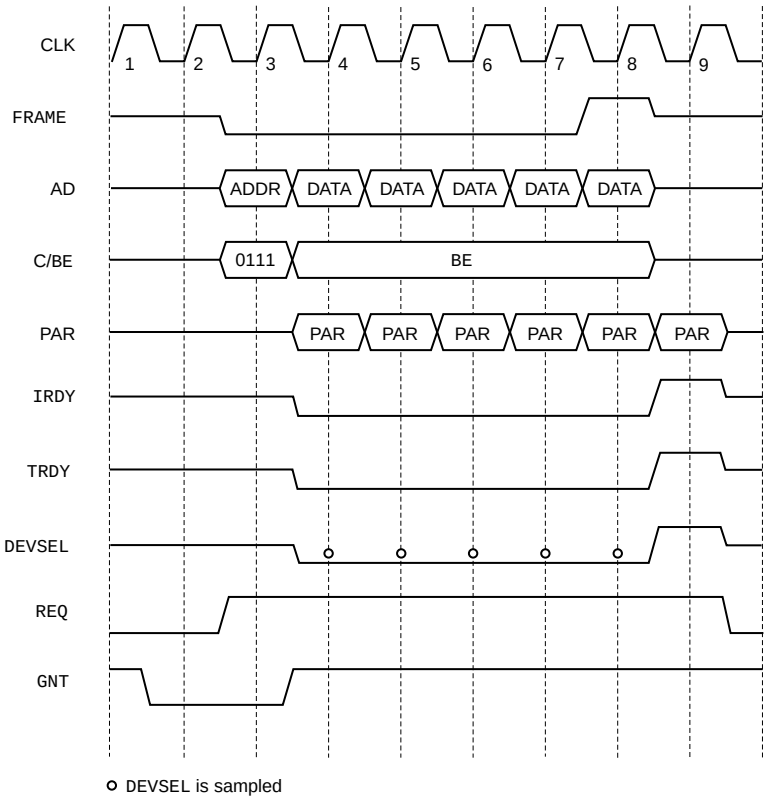
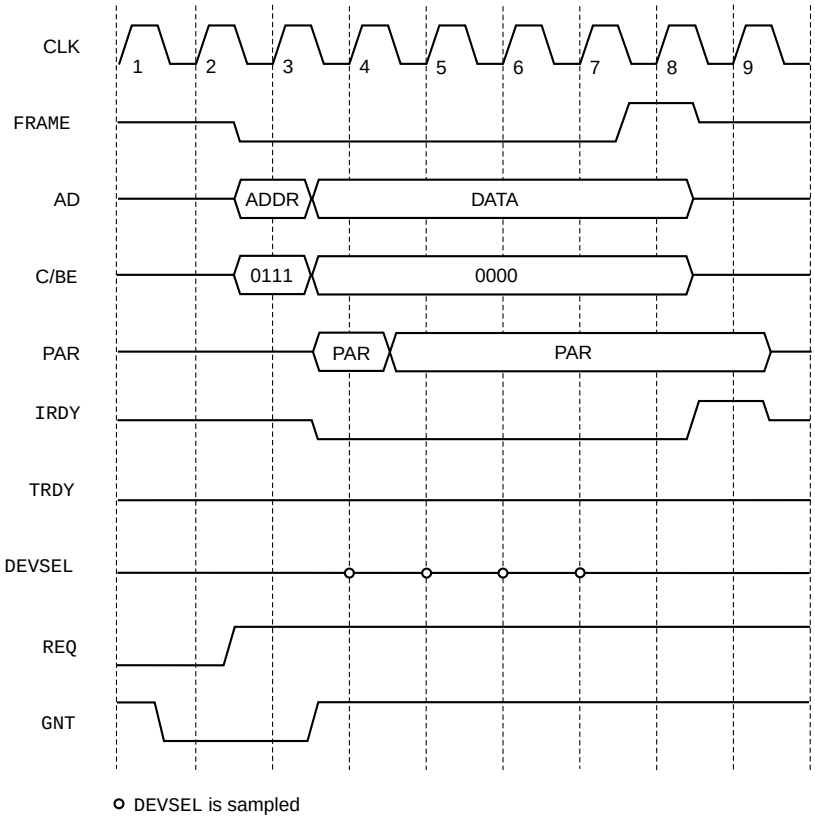
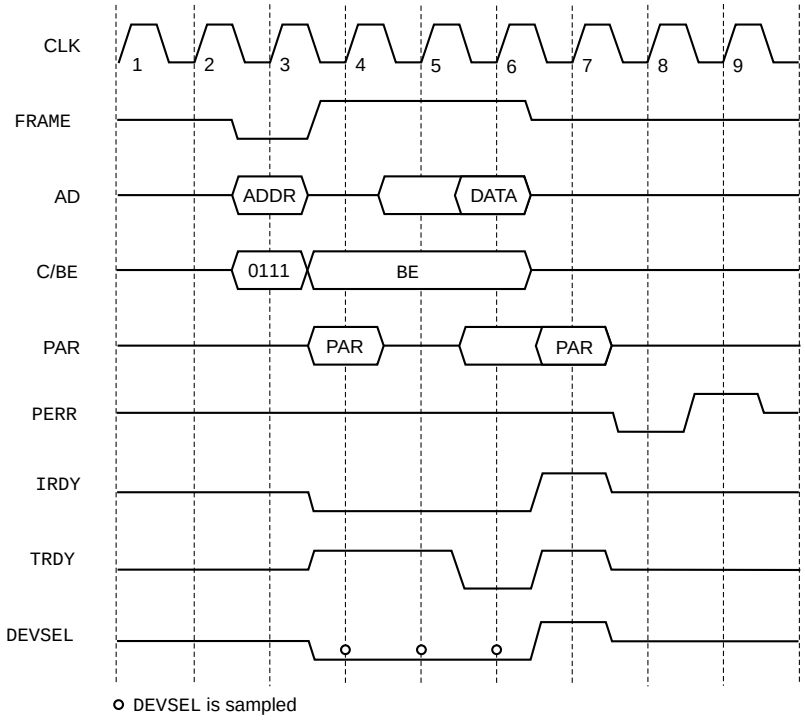


Figure 20. Preemption During Burst Transaction



21510D-26

Figure 21. Master Abort



21510D-27

Figure 22. Master Cycle Data Parity Error Response

Whenever the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller to use 32-bit software structures. The Am79C973/Am79C975 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.

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 Am79C973/Am79C975 controller is the target of the transfer.

Initialization Block DMA Transfers

During execution of the Am79C973/Am79C975 controller bus master initialization procedure, the Am79C973/Am79C975 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 master-ship 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 Am79C973/Am79C975 supports two transfer modes for reading the initialization block: non-burst and burst mode, with burst mode being the preferred mode when the Am79C973/Am79C975 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.

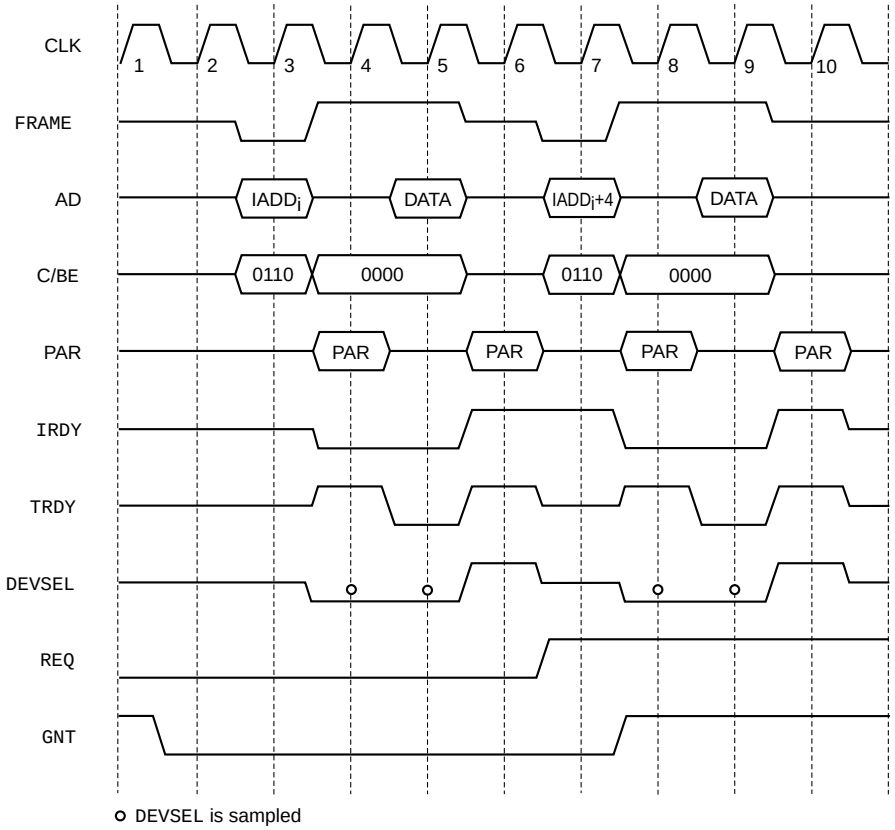


Figure 23. Initialization Block Read In Non-Burst Mode

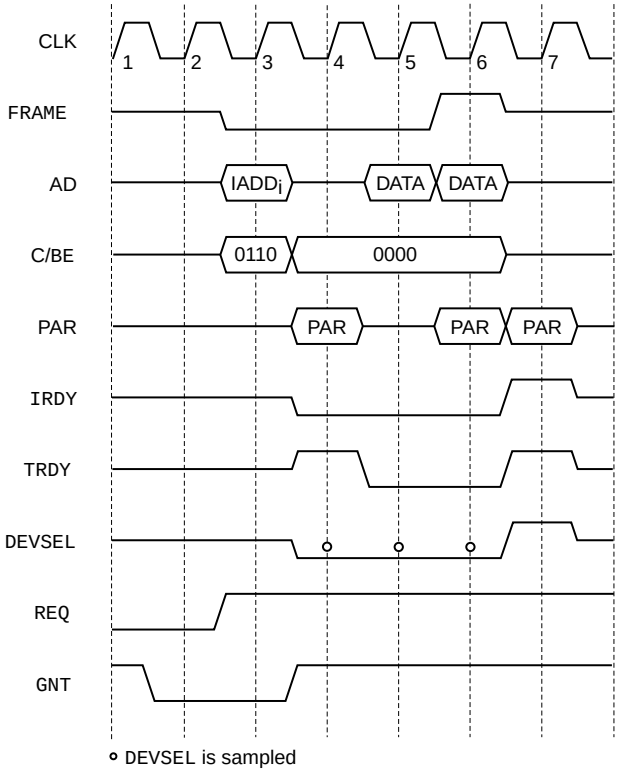


Figure 24. Initialization Block Read In Burst Mode

Descriptor DMA Transfers

Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will perform all descriptor read operations in burst mode, if BREADE is set to 1. See Figure 26.

Table 5 shows the descriptor read sequence.

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

If buffer chaining is used, accesses to the descriptors of all intermediate buffers consist of only one data transfer to return ownership of the buffer to the system. When SWSTYLE (BCR20, bits 7-0) is cleared to 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 Am79C973/Am79C975 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.

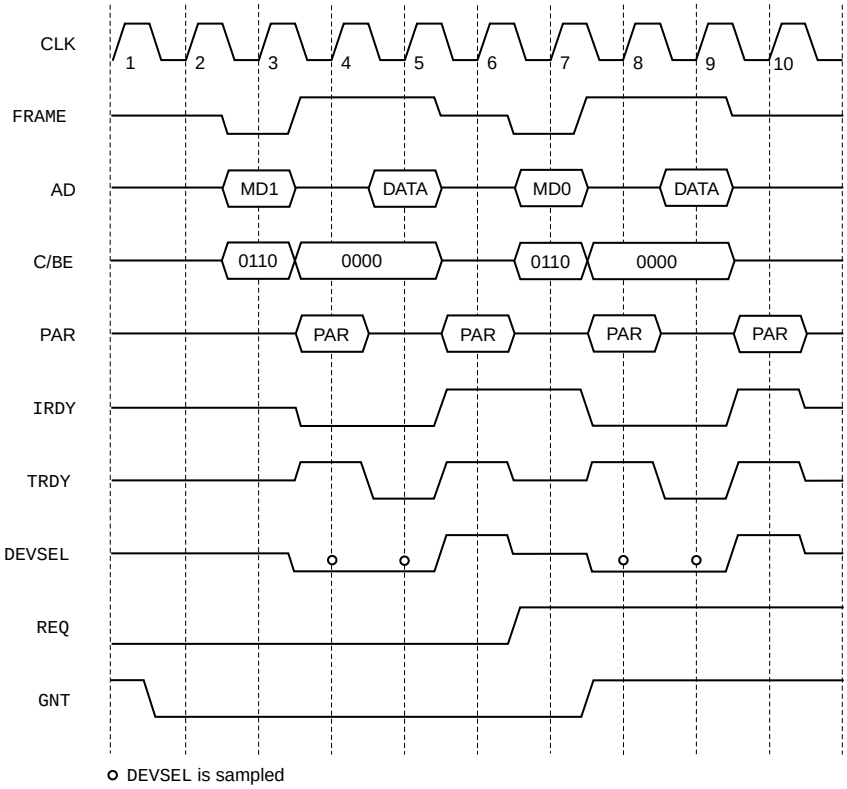
When SWSTYLE is set to 3, the descriptor entries are ordered to allow burst transfers. The Am79C973/Am79C975 controller will perform all descriptor write operations in burst mode, if BWRITE is set to 1. See Table 6 for the descriptor write sequence.

A write transaction to the descriptor ring entries is the only case where the Am79C973/Am79C975 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 Am79C973/Am79C975 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).

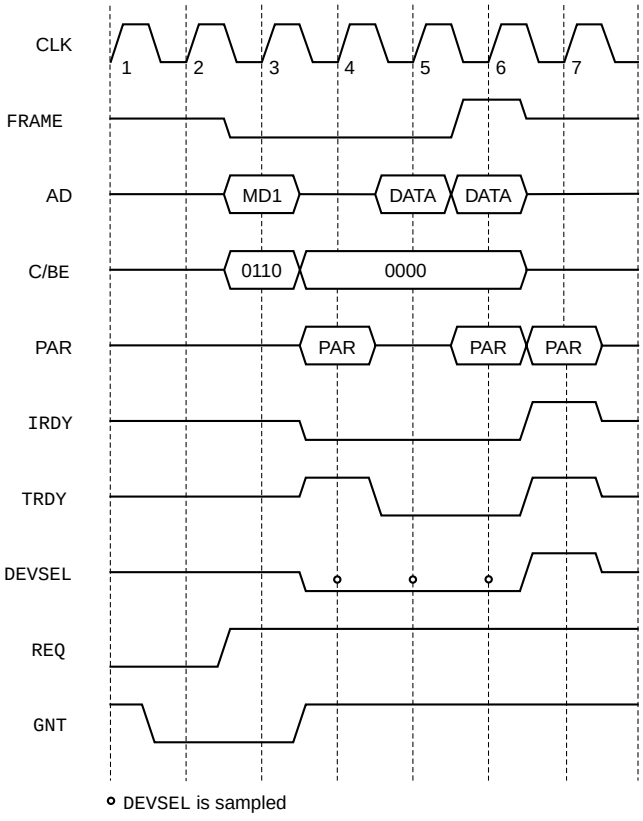
Table 5. Descriptor Read Sequence

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]



21510D-30

Figure 25. Descriptor Ring Read In Non-Burst Mode



B21510D-31

Figure 26. Descriptor Ring Read In Burst Mode

Table 6. 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

Am79C973/Am79C975 microcode will determine when a FIFO DMA transfer is required. This transfer mode will be used for transfers of data to and from the Am79C973/Am79C975 FIFOs. Once the Am79C973/Am79C975 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.

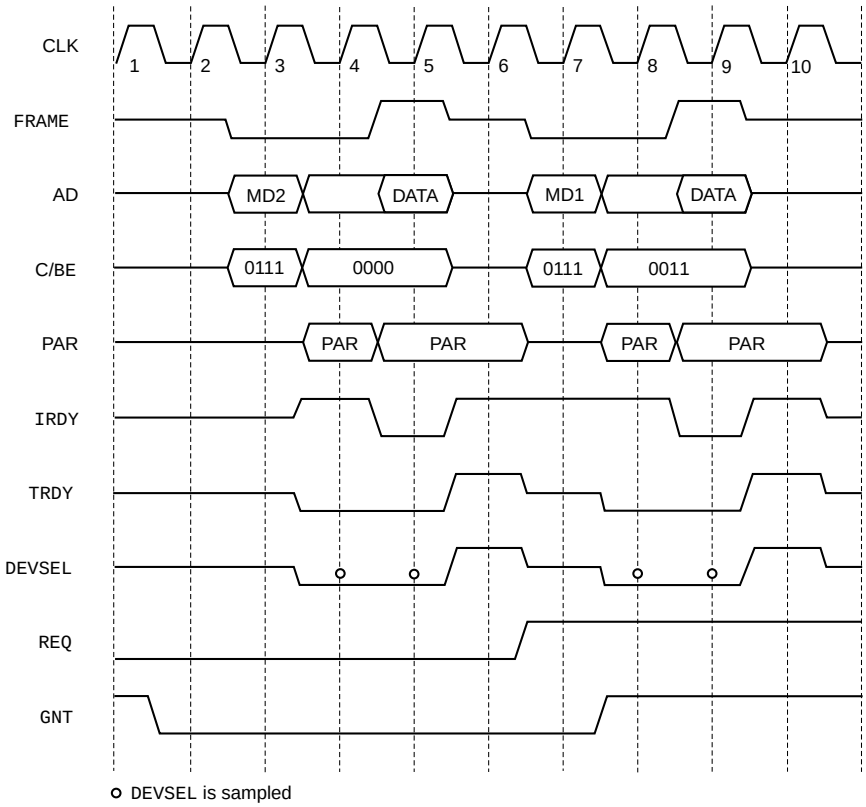
Non-Burst FIFO DMA Transfers

In the default mode, the Am79C973/Am79C975 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:

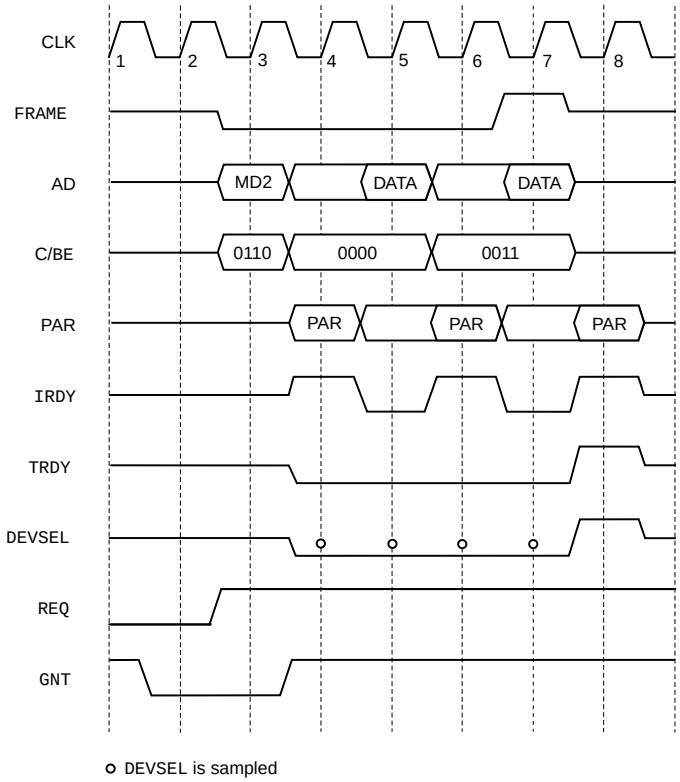
Bus cycles will continue until the transmit FIFO is filled to its high threshold (read transfers) or the receive FIFO is emptied to its low threshold (write transfers). The exact number of total transfer cycles in the bus mastership period is dependent on all of the following variables: the settings of the FIFO watermarks, the conditions of the FIFOs, the latency of the system bus to the Am79C973/Am79C975 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.*



21510D-32

Figure 27. Descriptor Ring Write In Non-Burst Mode



21510D-33

Figure 28. Descriptor Ring Write In Burst Mode

Burst FIFO DMA Transfers

Bursting is only performed by the Am79C973/ Am79C975 controller if the BREADE and/or BWRITE bits of BCR18 are set. These bits individually enable/ disable the ability of the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller can continue bursting full DWords.

If a receive buffer does not end on a DWord boundary, the Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 controller will always perform a DWord transfer as long as it owns the buffer space, even when there are less than four bytes to write. For example, if there is only one byte left for the current receive frame, the Am79C973/Am79C975 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.

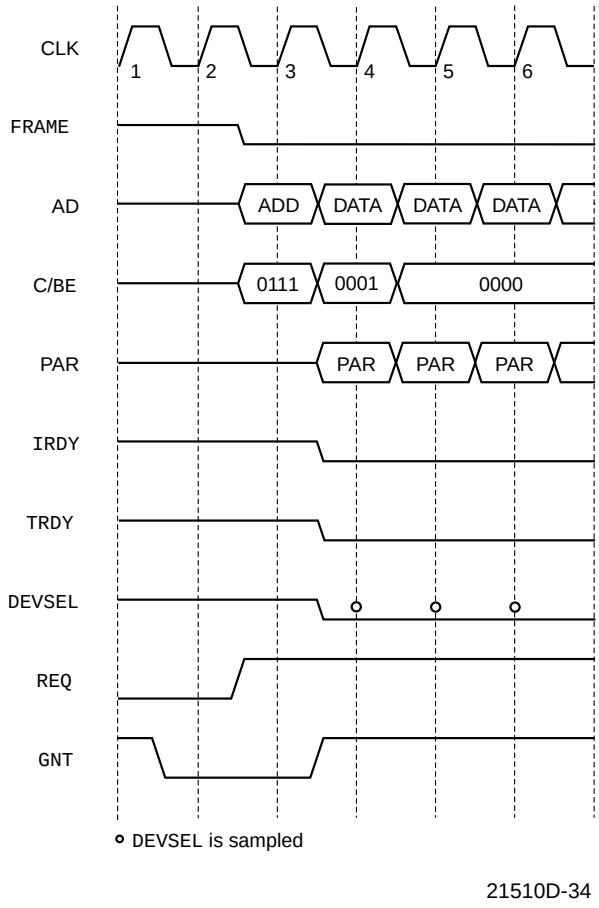


Figure 29. FIFO Burst Write At Start Of Unaligned Buffer

The Am79C973/Am79C975 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 Am79C973/Am79C975 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.

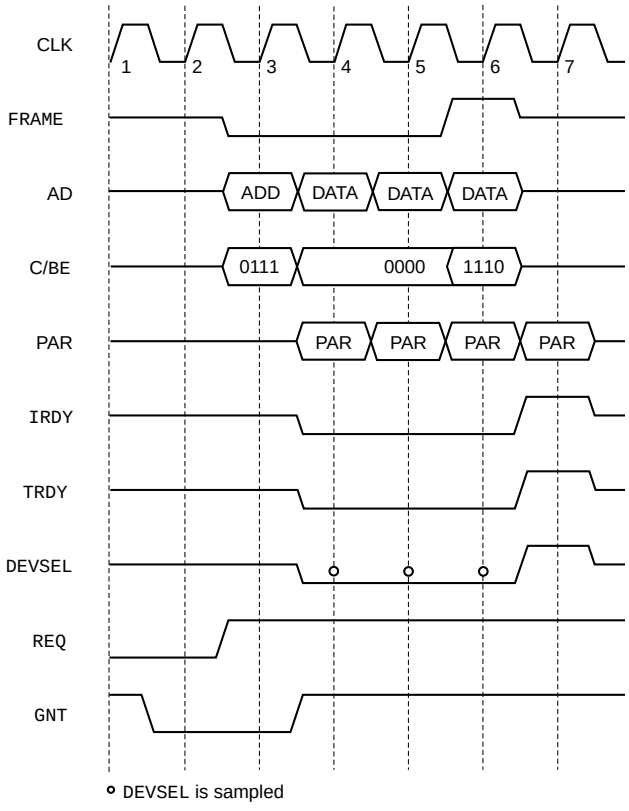


Figure 30. FIFO Burst Write At End Of Unaligned Buffer

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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

Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 operation, together with the base addresses and length information of the transmit and receive descriptor rings.

There is an alternate method to initialize the Am79C973/Am79C975 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 A, Alternative Method for Initialization* for details on this alternate method.

Re-Initialization

The transmitter and receiver sections of the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller as in the C-LANCE device. In particular, upon restart, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller offers two suspend modes that allow 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller has entered the suspend mode. When the host sets SPND to 1, the procedure taken by the Am79C973/Am79C975 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 Am79C973/Am79C975 controller performs a quick entry into the suspend mode. At the time the SPND bit is set, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller may enter the suspend mode with transmit and/or receive packets still in the FIFOs or the SRAM. This offers a worst case suspend time of a maximum length packet over the possibility of completely emptying the SRAM. Care must be exercised in this mode, because the entire memory subsystem of the Am79C973/Am79C975 controller is suspended. Any changes to either the descriptor rings or the SRAM can cause the Am79C973/Am79C975 controller to start up in an unknown condition and could cause data corruption.

When FASTSPNDE is 0 and the SPND bit is set, the Am79C973/Am79C975 controller may take longer before entering the suspend mode. At the time the SPND bit is set, the Am79C973/Am79C975 controller will complete the DMA process of a transmit packet if it had already begun and the Am79C973/Am79C975 controller will completely receive a receive packet if it had already begun. The Am79C973/Am79C975 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 SRAM (if one is present) will be transmitted, and all receive packets stored in the receive FIFOs and the receive buffer area in the SRAM (if selected) will be transferred into system memory. Since the FIFO and the SRAM contents are flushed, it may take much longer before the Am79C973/Am79C975 controller enters the suspend mode. The amount of time that it takes depends on many factors including the size of the SRAM, bus latency, and network traffic level.

Upon completion of the described operations, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will leave the suspend mode and will continue at the transmit and receive descriptor ring locations where it was when it entered the suspend mode.

See the section on *Magic Packet*™ technology for details on how that affects suspension of the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller reads the base address of both the transmit and receive descriptor rings into CSRs for use by the Am79C973/Am79C975 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.

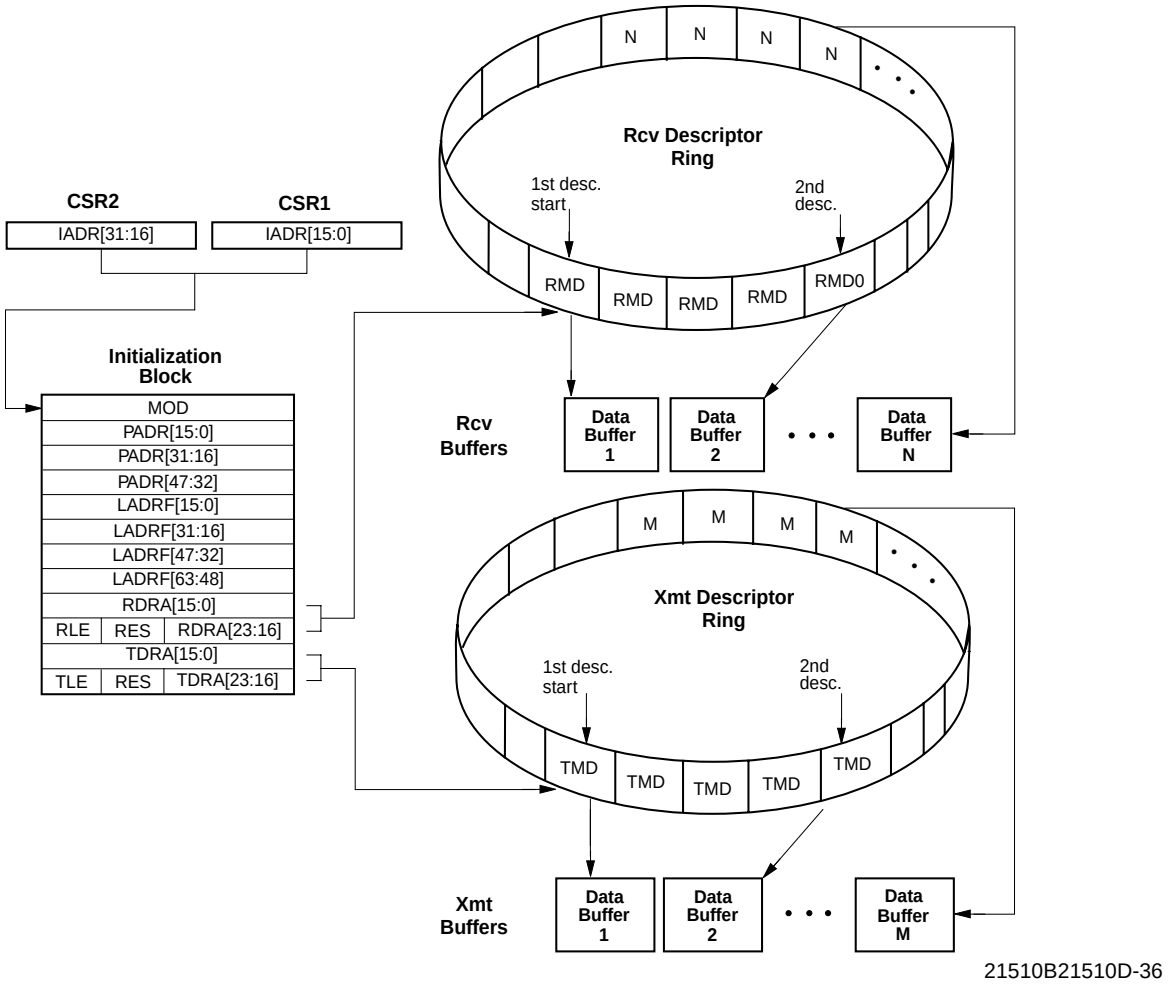


Figure 31. 16-Bit Software Model

Note: *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 Am79C973/Am79C975 controller, then the Am79C973/Am79C975 controller will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership. If the DPOLL bit in CSR4 is set, then the transmit polling function is disabled.

A typical polling operation consists of the following sequence. The Am79C973/Am79C975 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. Am79C973/Am79C975 controller does not own the current RDTE *and* the poll time has elapsed *and* RXON = 1 (CSR0, bit 5), *or*
2. Am79C973/Am79C975 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.

.

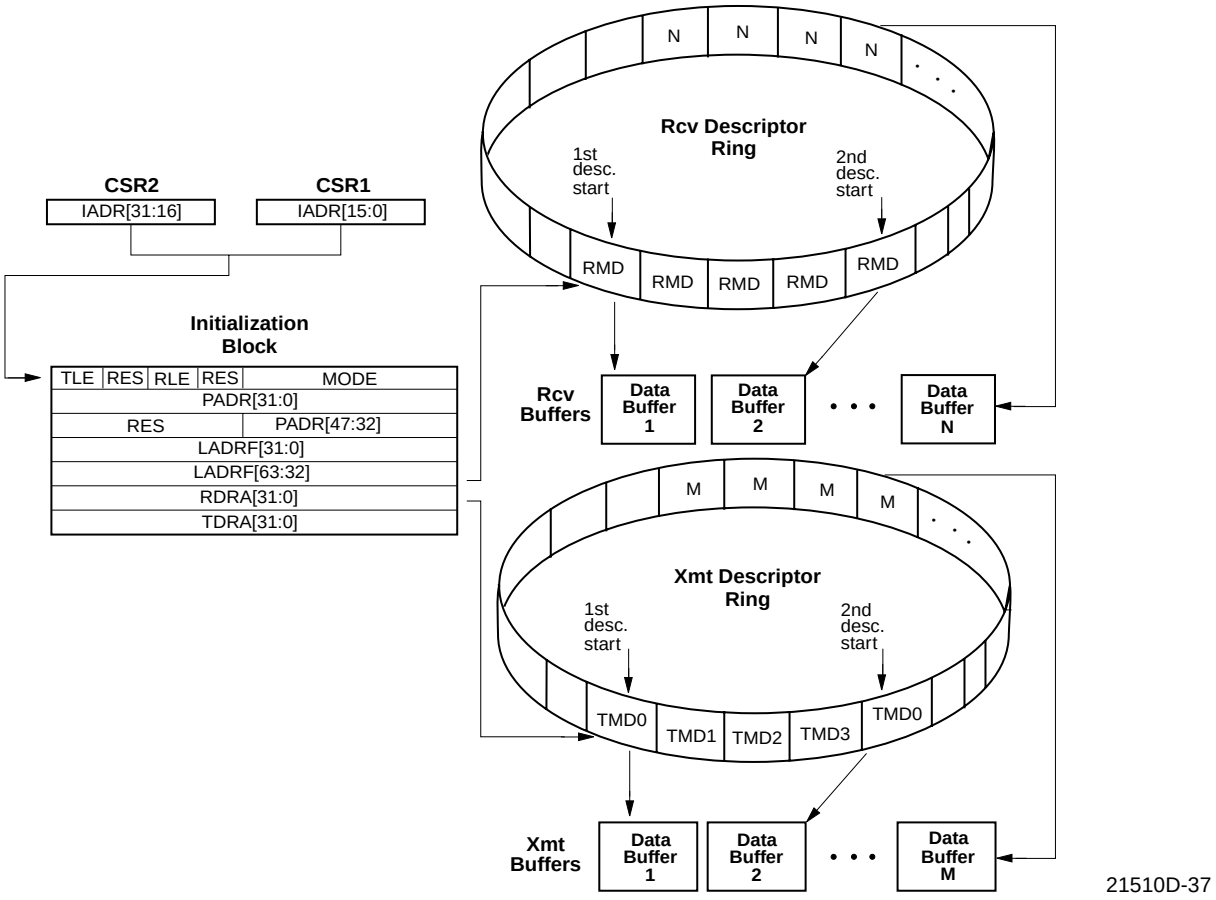


Figure 32. 32-Bit Software Model

If RXON is cleared to 0, the Am79C973/Am79C975 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. Am79C973/Am79C975 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. Am79C973/Am79C975 controller does not own the current TDTE *and* TXDPOLL = 0 *and* TXON = 1 *and* a frame has just been received, or
3. Am79C973/Am79C975 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 Am79C973/Am79C975 controller finds that the OWN bit of that TDTE is not set, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will have to be re-initialized to restore the transmit function. Setting DXSUFLO to 1 enables the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 controller normally clears OWN bits in strict FIFO order. However, the Am79C973/Am79C975 controller can queue up to two frames in the transmit FIFO. When the second frame uses buffer chaining, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will skip over the rest of the frame which experienced the error. This is done by returning to the polling microcode where the Am79C973/Am79C975 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 Am79C973/Am79C975 controller will always perform another polling operation. As described earlier, this polling operation will begin with a check of the current RDTE, unless the Am79C973/Am79C975 controller already owns that descriptor. Then the Am79C973/Am79C975 controller will poll the next TDTE. If the transmit descriptor OWN bit has a 0 value, the Am79C973/Am79C975 controller will resume incrementing the poll time counter. If the transmit descriptor OWN bit has a value of 1, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller to avoid inserting poll time counts between successive transmit frames.

By default, whenever the Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 controller does not own both the current and the next Receive Descriptor Table Entry (RDTE), then the Am79C973/Am79C975 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 Am79C973/Am79C975 controller, then additional poll accesses are not necessary. Future poll operations will not include RDTE accesses as long as the Am79C973/Am79C975 controller retains ownership of the current and the next RDTE.

When receive activity is present on the channel, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller checks the current receive buffer status register CRST (CSR41) to determine the ownership of the current buffer.

If ownership is lacking, the Am79C973/Am79C975 controller will immediately perform a final poll of the current RDTE. If ownership is still denied, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller recognizes the completion of the frame (the last byte of this receive message has been removed from the FIFO). The Am79C973/Am79C975 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 Am79C973/Am79C975 controller supports the lack of RDTEs when SRAM (SRAM SIZE in BCR 25, bits 7-0) is enabled through the Receive Frame Queuing mechanism. When the SRAM SIZE = 0, then the Am79C973/Am79C975 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 enabled, 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will also try to access the RDTE during normal descriptor accesses whether they are transmit or receive accesses. The host can force the Am79C973/Am79C975 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 Am79C973/Am79C975 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 256 Time Base Clock periods. For instance, a value of 122 ms would be programmed with a value of 9531 (253Bh), if the Time Base Clock is running at 20 MHz. 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.

10/100 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 internal PHY.

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 retransmission 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 MAC engine will detect the incoming preamble sequence when the RX_DV signal is activated by the internal PHY. The MAC will discard the preamble and begin searching for the SFD. 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 Am79C973/Am79C975 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 (ONE, 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 ms 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.

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

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 Inter-Frame-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 InterFrameSpacing-Part1 time of 6.0 ms. The InterFrameSpacingPart 2 interval is, therefore, 3.4 ms.

The Am79C973/Am79C975 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 ms InterFrameSpacing after the receive carrier is deasserted. During the first part deferral (InterFrameSpacingPart1 - IFS1), the Am79C973/Am79C975 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 ms count once again. Once the IFS1 period of 6.0 ms has elapsed, the Am79C973/Am79C975 controller will begin timing the second part deferral (InterFrameSpacingPart2 - IFS2) of 3.4 ms. Once IFS1 has completed and IFS2 has commenced, the Am79C973/Am79C975 controller will not defer to a receive frame if a transmit frame is pending. This means that the Am79C973/Am79C975 controller will not attempt to receive the receive frame, since it will start to transmit and generate a collision at 9.6 ms. The Am79C973/Am79C975 controller will complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

The Am79C973/Am79C975 controller allows the user to program the IPG and the first part deferral (Inter-Frame-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 Am79C973/Am79C975 MAC on the network. By programming a lower number of bit times than the ISO/IEC 8802-3 standard requires, the Am79C973/Am79C975 MAC engine will become more aggressive on the network. This aggressive nature will give rise to the Am79C973/Am79C975 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 Am79C973/Am79C975 MAC will become less aggressive on the network and may defer more often than normal. The performance of the Am79C973/Am79C975 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 Am79C973/Am79C975 controller uses the same IPG for back-to-back transmits and receive-to-transmit accesses. Changing IFS1 will alter the period for which the Am79C973/Am79C975 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 should generate the SQE Test message within 0.6 to 1.6 ms after the transmission ceases. During the time period in which the SQE Test message is expected, the Am79C973/Am79C975 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* ms *but no more than 8.0* ms*. During the time window the Carrier Sense Function is inhibited."*

The Am79C973/Am79C975 controller implements a carrier sense "blinding" period of 4.0 ms 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 ms to 6 ms after a transmission. How-

ever, since IPG shrinkage below 4 ms will rarely be encountered on a correctly configured network, and since the fragment size will be larger than the 4 ms blinding window, the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the Am79C973/Am79C975 controller will defer its transmission. If carrier is detected within the 4.0 to 6.0 ms IFS1 period, the Am79C973/Am79C975 controller will not restart the "blinding" period, but only restart IFS1.

Collision Handling

Collision detection is performed and reported to the MAC engine via the COL input pin.

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

The Am79C973/Am79C975 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 Am79C973/Am79C975 controller are controlled by programmable options. The Am79C973/Am79C975 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 (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 Am79C973/Am79C975 controller to compute the actual number of pad bytes to be inserted. The Am79C973/Am79C975 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 Am79C973/Am79C975 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.

.

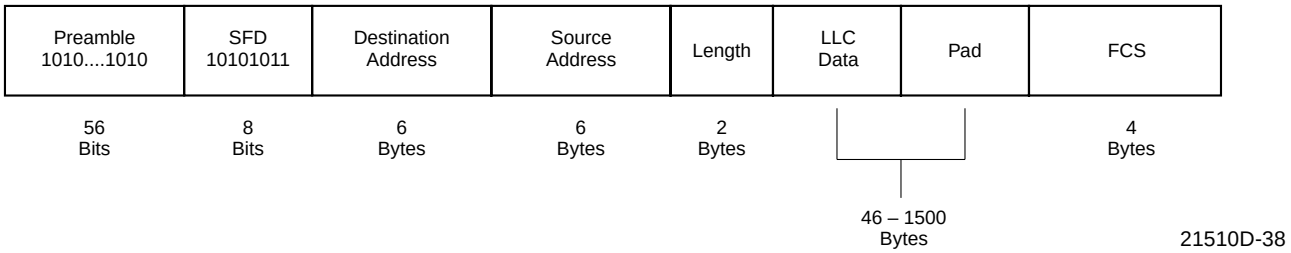


Figure 33. ISO 8802-3 (IEEE/ANSI 802.3) Data Frame

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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller regardless of the state of DXMTFCS or ADD_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 all descriptors 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 Am79C973/Am79C975 controller include collisions within the slot time with automatic retry. The Am79C973/Am79C975 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 Am79C973/Am79C975 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 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

LCAR will be reported for every frame transmitted if the controller detects a loss of carrier.

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 Am79C973/Am79C975 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

CERR will be asserted in the 10BASE-T mode 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 Am79C973/Am79C975 controller are controlled by programmable options. The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller can be programmed to accept runt packets by setting RPA in CSR124.

Address Matching

The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will not accept unicast frames.

If the incoming frame is multicast, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller hardware. Broadcast frames are always accepted, except when DRCVBC (CSR15, bit 14) is set and there is no Logical Address match.

None of the address filtering described above applies when the Am79C973/Am79C975 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 Am79C973/Am79C975 controller operates in promiscuous mode when PROM (CSR15, bit 15) is set.

In addition, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller to accept the frame. Note that these indicator bits are only available when the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller operates in promiscuous mode and none of the three match bits is set, it is an indication that the Am79C973/Am79C975 controller only accepted the frame because it was in promiscuous mode.

When the Am79C973/Am79C975 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 Am79C973/Am79C975 controller only accepted the frame because it was not rejected by driving the $\overline{EAR}$ pin LOW within 64 bytes after SFD.

See Table 7 for receive address matches.

Table 7. Receive Address Match

PAM	LAFM	BAM	DRCVBC	Comment
0	0	0	X	Frame accepted due to PROM = 1 or no EADI reject
1	0	0	X	Physical address match
0	1	0	0	Logical address filter match; frame is not of type broadcast
0	1	0	1	Logical address filter match; frame can be of type broadcast
0	0	1	0	Broadcast frame

Automatic Pad Stripping

During reception of an 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.

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.

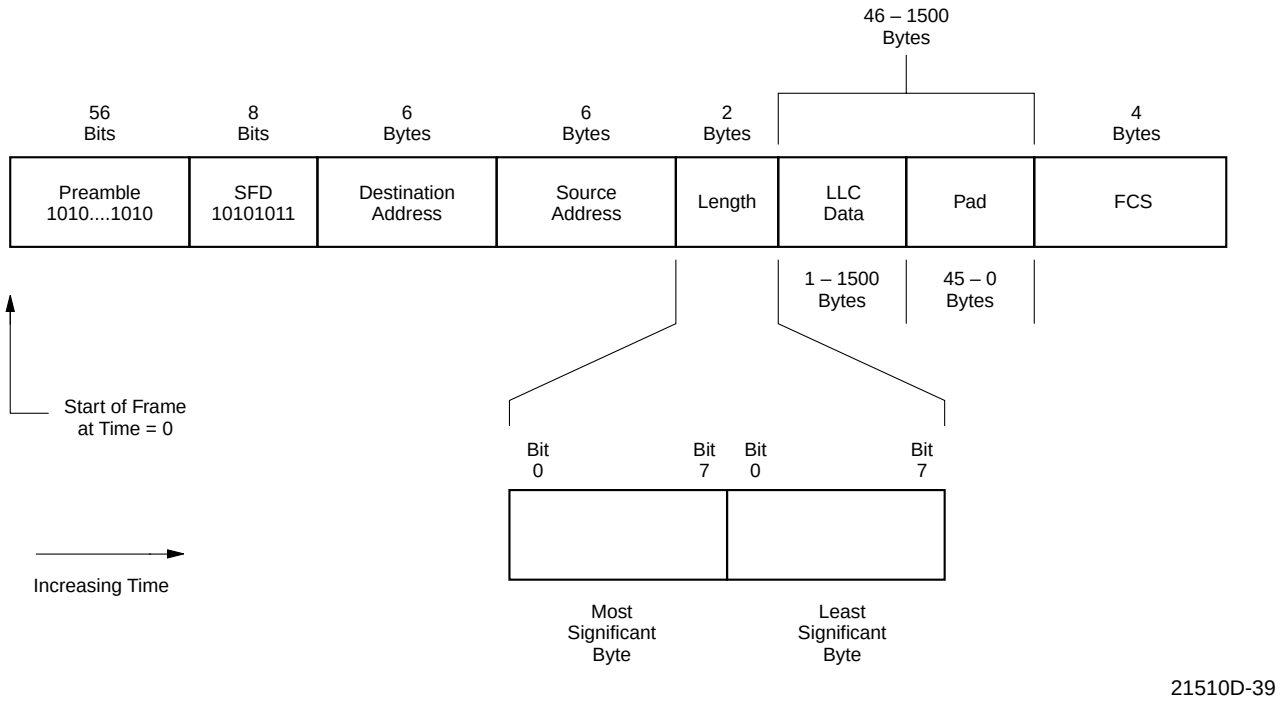


Figure 34. IEEE 802.3 Frame And Length Field Transmission Order

Since any valid Ethernet Type field value will always be greater than a normal IEEE 802.3 Length field (Š46), the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller are basically collisions within the slot time and automatic runt packet rejection. The Am79C973/Am79C975 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, or the full-duplex Runt Packet Accept Disable bit (FDRPAD, BCR9, bit 2) is set. 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.

Refer to Table 21 for various bit settings required for Loopback modes.

The external loopback requires a two-step operation. The internal PHY must be placed into a loopback mode by writing to the PHY Control Register (BCR33, BCR34). Then, the Am79C973/Am79C975 controller must be placed into an external loopback mode by setting the Loop bits.

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.

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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.

Full-Duplex Operation

The Am79C973/Am79C975 controller supports full-duplex operation on both network interfaces. Full-duplex operation allows simultaneous transmit and receive activity. Full-duplex operation is enabled by the FDEN bit located in BCR9. Full-duplex operation is also enabled through Auto-Negotiation when DANAS (BCR 32, bit 7) is not enabled and the ASEL bit is set, and its link partner is capable of Auto-Negotiation and full-duplex operation.

When operating in full-duplex mode, the following changes to the device operation are made:

Bus Interface/Buffer Management Unit changes:

- The first 64 bytes of every transmit frame are not 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.
- The 4.0 µs carrier sense blinding period after a transmission during which the SQE test normally occurs is disabled.
- The collision indication input to the MAC engine is ignored.

The internal PHY changes for full-duplex operation are as follows:

- The collision detect (COL) pin is disabled.
- The SQE test function is disabled (10 Mbps).
- Loss of Carrier (LCAR) reporting is disabled.

- PHY Control Register (ANR0) bit 8 is set to 1 if Auto-Negotiation is disabled.

Full-Duplex Link Status LED Support

The Am79C973/Am79C975 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 in Full-Duplex.

10/100 PHY Unit Overview

The 10/100 PHY unit implements the complete physical layer for 10BASE-T and the Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA), and Physical Medium Dependent (PMD) functionality for 100BASE-TX. The 10/100 PHY implements Auto-Negotiation allowing two devices connected across a link segment to take maximum advantage of their capabilities. Auto-Negotiation is performed using a modified 10BASE-T link integrity test pulse sequence as defined in the IEEE 802.3u specification.

The internal 10/100 PHY consists of the following functional blocks:

- 100BASE-X Block including:
  - — Transmit and Receive State Machines
  - — 4B/5B Encoder and Decoder
  - — Stream Cipher Scrambler and Descrambler
  - — Link Monitor State Machine
  - — Far End Fault Indication (FEFI) State Machine
  - — MLT-3 Encoder
  - — MLT-3 Decoder with adaptive equalization
- 10BASE-T Block including:
  - — Manchester Encoder/Decoder
  - — Collision Detection
  - — Jabber
  - — Receive Polarity Detect
  - — Waveshaping and Filtering
- Auto-Negotiation
- Physical Data Transceiver (PDX)
- PHY Control and Management

100BASE-TX Physical Layer

The functions performed by the device include encoding of 4-bit data (4B/5B), decoding of received code groups (5B/4B), generating carrier sense and collision detect indications, serialization of code groups for transmission, de-serialization of serial data from reception, mapping of transmit, receive, carrier sense, and collision at the PHY/MAC interface, and recovery of clock from the incoming data stream. It offers stream cipher scrambling and descrambling capability for 100BASE-TX applications.

In the transmit data path for 100 Mbps, the 10/100 PHY receives 4-bit (nibble) wide data across the internal MII at 25 million nibbles per second. For 100BASE-TX applications, it encodes and scrambles the data, serializes it, and transmits an MLT-3 data stream to the media via an isolation transformer.

The 10/100 PHY receives an MLT-3 data stream from the network for 100BASE-TX. It then recovers the clock from the data stream, de-serializes the data stream, and descrambles/decodes the data stream (5B/4B) before presenting it to the internal MII interface.

100BASE-FX (Fiber Interface)

The Am79C973/Am79C975 device supports a Pseudo-ECL (PECL) interface for Fiber applications. The mode is enabled when BCR2 bit 14 (DISSCR_SFEX) is set to 1 and the Signal Detect pins SDI± are connected to the optical transceiver.

For 100BASE-FX receive operation, the PHY unit receives a PECL data stream from the optical transceiver and decodes the data stream. For transmit operation, the PHY unit encodes and serializes the data and transmits a pseudo-ECL data stream to the fiber optic transceiver. See Figure 35.

The Fiber Interface (100BASE-FX) does not support Auto-Negotiation, 10 BASE-FL, and data scrambling. When the device is set to operate in PECL mode, the 100BASE-TX operation will be disabled.

10BASE-T Physical Layer

The 10/100 PHY incorporates 10BASE-T physical layer functions, including both clock recovery (ENDEC) and transceiver functions. Data transmission over the 10BASE-T medium requires an integrated 10BASE-T MAU. The transceiver will meet the electrical requirements for 10BASE-T as specified in IEEE 802.3i. The transmit signal is filtered on the transceiver to reduce harmonic content per IEEE 802.3i. Since filtering is performed in silicon, external filtering modules are not needed. The 10/100 PHY receives 10-Mbps data from the MAC across the internal MII at 2.5 million nibbles per second for 10BASE-T. It then Manchester encodes the data before transmission to the network.

The RX± pins are differential twisted-pair receivers. When properly terminated, each receiver will meet the electrical requirements for 10BASE-T as specified in IEEE 802.3i. Each receiver has internal filtering and does not require external filter modules. The 10/100 PHY receives a Manchester coded 10BASE-T data stream from the medium. It then recovers the clock and decodes the data.

PHY/MAC Interface

The internal MII-compatible interface provides the data path connection between the 10/100 PHY and 10/100 Media Access Control (MAC). The interface is compatible with Clause 22 of the IEEE 802.3 standard specification.

Transmit Process

The transmit process generates code-groups based on TXD[3:0], TX_EN, TX_ER signals on the internal MII. These code-groups are transmitted by the PDX block. This process is also responsible for frame encapsulation into a Physical Layer Stream, generating the collision signal based on whether a carrier is received simultaneously with transmission and generating the Carrier Sense (CRS) and Collision (COL) signals at the internal MII. The transmit process is implemented in compliance with the transmit state diagram as defined in Clause 24 of the IEEE 802.3u specification. Figure 38 shows the transmit process.

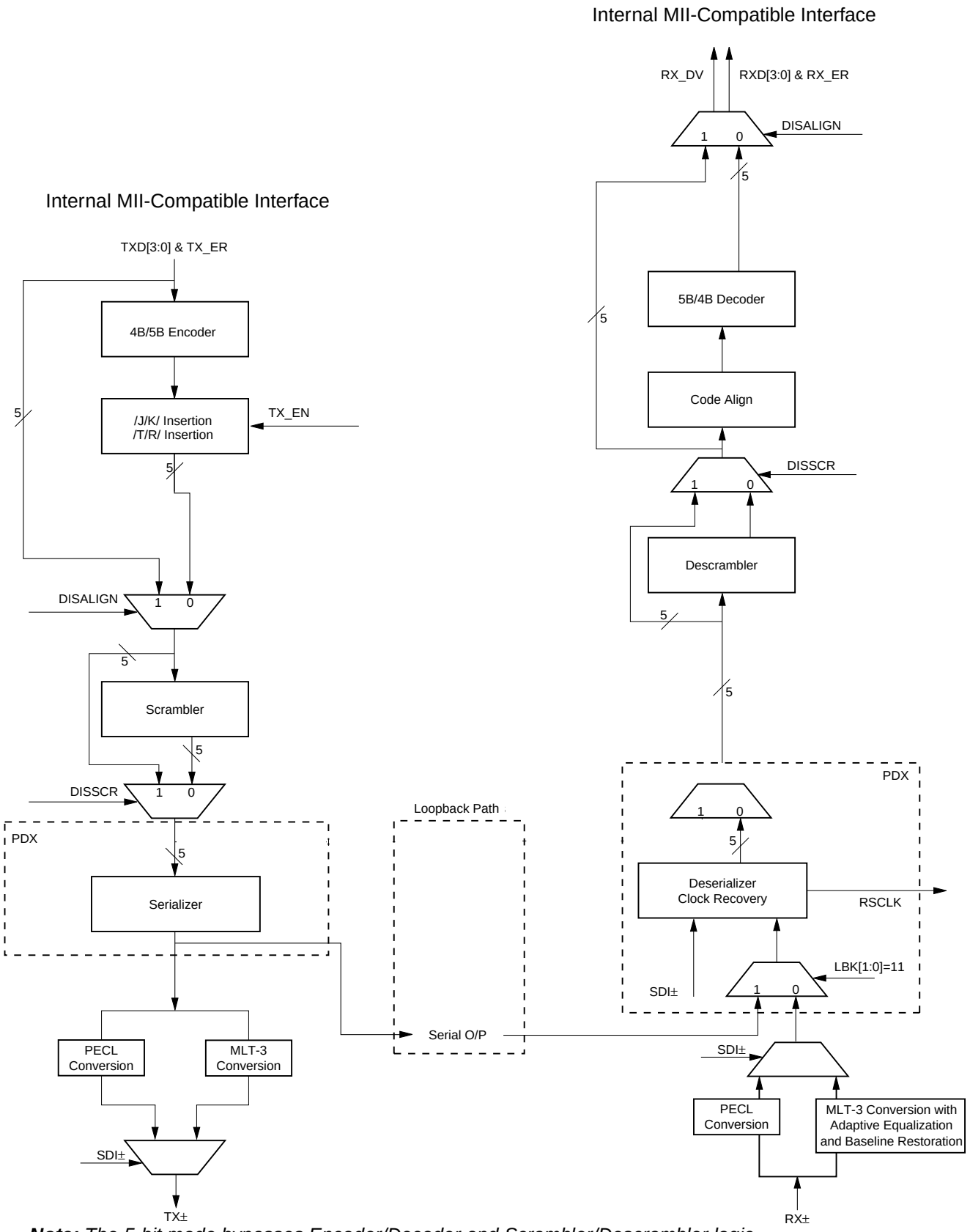
Receive Process

The receive process passes to the internal MII a sequence of data nibbles derived from the incoming code-groups. Each code-group is comprised of five code-bits. This process detects channel activity and then aligns the incoming code bits in code-group boundaries for subsequent data decoding. The receive process is responsible for code-group alignment and also generates the Carrier Sense (CRS) signal at the internal MII. The receive process is implemented in compliance with the receive state diagram as defined in Clause 24 of the IEEE 802.3u specification. The False Carrier Indication as specified in the standard is also generated by this block, and communicated to the Reconciliation layer. Figure 38 shows the receive process.

Internal PHY Loopback Path

As shown in Figure 35, the 10/100 PHY provides an internal loopback path for system testing purposes. The loopback option utilizes the serial loopback path from the PDX serial output to the PDX serial input and can be programmed via the LBK[1:0] bits in the PHY Control/Status Register (ANR17).

For the corresponding LBK setting, refer to the description for the PHY Control/Status Register.



Note: *The 5-bit mode bypasses Encoder/Decoder and Scrambler/Descrambler logic.*

21510D-40

Figure 35. 100BASE-X Transmit and Receive Data Paths of the Internal PHY

Encoder

The encoder converts the 4-bit nibble from the MII into five-bit code-groups, using a 4B/5B block coding scheme. The encoder operates on the 4-bit data nibble independent of the code-group boundary. The 100BASE-X physical protocol data unit is called a *stream*. The encoding method used provides the following:

- Adequate codes (32) to provide for all data code-groups (16) plus necessary control code-groups.
- Appropriate coding efficiency (4 data bits per 5 code-bits; 80%) to implement a 100-Mbps physical layer interface on a 125-Mbps physical channel.
- Sufficient transition density to facilitate clock recovery (when not scrambled).

The code-group mapping is defined in Table 8.

Table 8. Encoder Code-Group Mapping

TXD[3:0]	Name	PCS Code-Group	Interpretation
0 0 0 0	0	1 1 1 1 0	Data 0
0 0 0 1	1	0 1 0 0 1	Data 1
0 0 1 0	2	1 0 1 0 0	Data 2
0 0 1 1	3	1 0 1 0 1	Data 3
0 1 0 0	4	0 1 0 1 0	Data 4
0 1 0 1	5	0 1 0 1 1	Data 5
0 1 10	6	0 1 1 1 0	Data 6
0 1 1 1	7	0 1 1 1 1	Data 7
1 0 0 0	8	1 0 0 1 0	Data 8
1 0 0 1	9	1 0 0 1 1	Data 9
1 0 1 0	A	1 0 1 1 0	Data A
1 0 1 1	B	1 0 1 1 1	Data B
1 1 0 0	C	1 1 0 1 0	Data C
1 1 0 1	D	1 1 0 1 1	Data D
1 1 1 0	E	1 1 1 0 0	Data E
1 1 1 1	F	1 1 1 0 1	Data F
Undefined	I	1 1 1 1 1	IDLE; used as inter-Stream fill code
0 1 0 1	J	1 1 0 0 0	Start-of-Stream Delimiter, Part 1 of 2; always used in pairs with K
0 1 0 1	K	1 0 0 0 1	Start-of-Stream Delimiter, Part 2 of 2; always used in pairs with J
Undefined	T	0 1 1 0 1	End-of-Stream Delimiter, Part 1 of 2; always used in pairs with R
Undefined	R	0 0 1 1 1	End-of-Stream Delimiter, Part 2 of 2; always used in pairs with T
Undefined	H	0 0 1 0 0	Transmit Error; used to force signaling errors
Undefined	V	0 0 0 0 0	Invalid Code
Undefined	V	0 0 0 0 1	Invalid Code
Undefined	V	0 0 0 1 0	Invalid Code
Undefined	V	0 0 0 1 1	Invalid Code
Undefined	V	0 0 1 0 1	Invalid Code
Undefined	V	0 0 1 1 0	Invalid Code
Undefined	V	0 1 0 0 0	Invalid Code
Undefined	V	0 1 1 0 0	Invalid Code
Undefined	V	1 0 0 0 0	Invalid Code
Undefined	V	1 1 0 0 1	Invalid Code

Decoder

The decoder performs the 5B/4B decoding of the received code-groups. The five bits of data are decoded into four bits of nibble data. The decoded nibble is then forwarded to the PCS Control block to be sent across the internal MII to the MAC unit. The code-group decoding is shown in Table 9.

Table 9. Decoder Code-Group Mapping

PCS Code-Group	Name	RXD[3:0]	Interpretation
1 1 1 1 0	0	0 0 0 0	Data 0
0 1 0 0 1	1	0 0 0 1	Data 1
1 0 1 0 0	2	0 0 1 0	Data 2
1 0 1 0 1	3	0 0 1 1	Data 3
0 1 0 1 0	4	0 1 0 0	Data 4
0 1 0 1 1	5	0 1 0 1	Data 5
0 1 1 1 0	6	0 1 10	Data 6
0 1 1 1 1	7	0 1 1 1	Data 7
1 0 0 1 0	8	1 0 0 0	Data 8
1 0 0 1 1	9	1 0 0 1	Data 9
1 0 1 1 0	A	1 0 1 0	Data A
1 0 1 1 1	B	1 0 1 1	Data B
1 1 0 1 0	C	1 1 0 0	Data C
1 1 0 1 1	D	1 1 0 1	Data D
1 1 1 0 0	E	1 1 1 0	Data E
1 1 1 0 1	F	1 1 1 1	Data F
1 1 1 1 1	I	Undefined	IDLE; used as Inter-Stream fill code
1 1 0 0 0	J	0 1 0 1	Start-of-Stream Delimiter, Part 1 of 2; always used in pairs with K
1 0 0 0 1	K	0 1 0 1	Start-of-Stream Delimiter, Part 2 of 2; always used in pairs with J
0 1 1 0 1	T	Undefined	End-of-Stream Delimiter, Part 1 of 2; always used in pairs with R
0 0 1 1 1	R	Undefined	End-of-Stream Delimiter, Part 2 of 2; always used in pairs with T
0 0 1 0 0	H	Undefined	Transmit Error; used to force signaling errors
0 0 0 0 0	V	Undefined	Invalid Code
0 0 0 0 1	V	Undefined	Invalid Code
0 0 0 1 0	V	Undefined	Invalid Code
0 0 0 1 1	V	Undefined	Invalid Code
0 0 1 0 1	V	Undefined	Invalid Code
0 0 1 1 0	V	Undefined	Invalid Code
0 1 0 0 0	V	Undefined	Invalid Code
0 1 1 0 0	V	Undefined	Invalid Code
1 0 0 0 0	V	Undefined	Invalid Code
1 1 0 0 1	V	Undefined	Invalid Code

Scrambler/Descrambler

The 4B/5B encoded data has repetitive patterns which result in peaks in the RF spectrum large enough to keep the system from meeting the standards set by regulatory agencies such as the FCC. The peaks in the radiated signal are reduced significantly by scrambling the transmitted signal. Scramblers add the output of a random generator to the data signal. The resulting signal has fewer repetitive data patterns.

After reset, the scrambler seed will be set to the PHY address value to help improve the EMI performance of the device.

The scrambled data stream is descrambled, at the receiver, by adding it to the output of another random generator. The receiver's random generator has the same function as the transmitter's random generator.

Link Monitor

The Link Monitor process is responsible for determining whether the underlying receive channel is providing reliable data. This process takes advantage of the continuous indication of signal detect by the PMD (PDX & MLT-3). The process sets the link_status to FAIL whenever signal_status is OFF. The link is reliable whenever the signal_status has been continuously ON for 330 - 1000 ms. The implementation is in compliance with Clause 24 of the IEEE 802.3u specification.

The 10BASE-T Link Monitor monitors the line for link pulses, while the 100BASE-T Link Monitor expects 100 Mbps idle signals. When the Link Monitor detects both 10 Mbps and 100 Mbps signals, a state called Parallel Fault is entered, where the Link Monitor simply halts and fails to report a link. This condition can be caused by spurious noise on the network line. Consult the IEEE 802.3u specification for more information. The Parallel Fault Detect condition is displayed in Register 6, bit 4.

The current link status of this port is displayed In the PHY Management Status Register (Register 1, bit 2).

Far End Fault Generation and Detection

Far End Fault Generation and Detection is implemented in the 10/100 PHY for 100BASE-TX over STP and 100BASE-FX. This block generates a special Far End Fault indication to its far end peer. This indication is generated only when an error condition is detected on the receive channel. When Far End Fault Indication is detected from the far end peer, this block will cause the link monitor to transition the link_status to FAIL. This action in-turn will cause IDLE code-group bits to be automatically transmitted. This is necessary to re-establish communication when the link is repaired. The implementation is in compliance with the Clause 24 of IEEE 802.3u specification.

Far End Fault Indication can be initialized using the PHY Control/Status Register (ANR17, bit 10).

MLT-3 and Adaptive Equalization

This block is responsible for converting the NRZI data stream from the PDX block to a currently sourced MLT-3 encoded data stream. The effect of MLT-3 is the reduction of energy on the media (TX cable) in the critical frequency range of 20 MHz to 100 MHz. The receive section of this block is responsible for equalizing and amplifying the received data stream and link detection. The adaptive equalizer compensates for the amplitude and phase distortion due to the cable.

MLT-3 is a tri-level signal. All transitions are between 0 V and +1 V or 0 V and -1 V. A transition has a logical value of 1 and a lack of a transition has a logical value of 0. The benefit of MLT-3 is that it reduces the maximum frequency over the data line. The bit rate of TX data is 125 Mbps. The maximum frequency (using NRZI) is half of 62.5 MHz. MLT-3 reduces the maximum frequency to 31.25 MHz.

The implementation of this block is in compliance with ANSI X3712 TP-PMD/312, Revision 2.1 that defines a 125-Mbps, full-duplex signalling for twisted pair wiring.

A data signal stream following MLT-3 rules is illustrated in Figure 36. The data stream is 1010101.

The TX± drivers convert the NRZI serial output to MLT-3 format. The RX± receivers convert the received MLT-3 signals to NRZI. When the TX port of the 10/100 PHY is connected as in Figure 37, the transmit and receive signals will be compliant with IEEE 802.3u Section 25. The required signals (MLT-3) are described in detail in ANSI X3.263:1995 TP-PMD Revision 2.2 (1995).

The 10/100 PHY provides on-chip filtering. External filters are not required for either the transmit or receive signals.

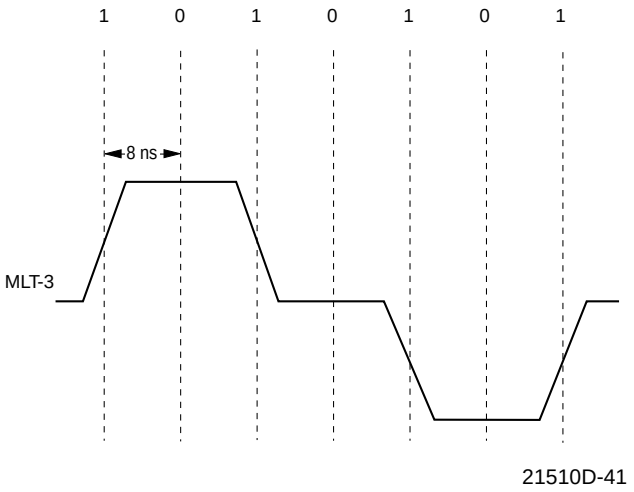


Figure 36. MLT-3 Waveform

Serializer/Deserializer and Clock Recovery

The Physical Data Transceiver (PDX) is a CMOS all digital core that is used in the 10/100 PHY. It employs new circuit techniques to achieve clock and data recovery.

Traditionally, Phase-Locked-Loops (PLLs) are used for the purpose of clock recovery in data communication areas. There are both analog and digital versions of the PLL components such as phase detector, filter, and charge pump. A traditional PLL always contains a voltage-controlled oscillator (VCO) to regenerate a clock which is synchronized in frequency to and aligned in phase with the received data.

The PDX employs techniques that are significantly different from traditional PLLs. Not only are the control functions completely digital, the VCO function is also replaced by a proprietary delay time ruler technique. The result is a highly integratible core which can be manufactured in a standard digital CMOS process.

To transmit, the PDX accepts 4B/5B encoded data symbols from the scrambler. The 5-bit symbol is clocked into the PDX by the rising edge of the 25-MHz clock, serialized and converted to NRZI format. The NRZI data is delivered to the PECL transceiver or MLT3 transceiver. The output of either of the two transceivers goes to the TX± pair.

The PDX uses a 25-MHz clock as the frequency and phase reference to generate the serial link data rate. The external clock source must be continuous. All of the internal logic of the PDX runs on an internal clock that is derived from the external reference source. The PDX's clock multiplier is referenced to the rising edges of the 25-MHz clock only.

In order to generate the serial output wave forms conforming to the specifications, the external reference clock must meet 100BASE-X frequency and stability requirements. Under normal conditions, the frequency of the 25-MHz clock multiplied by 5 must be within the 100BASE-X specified 100 ppm of the received data for the PDX to operate optimally.

Note: *The 100 ppm is the tolerance of the crystal-controlled source.*

The TX± serial output typically contains less than 0.4 ns peak-to-peak jitter at 125 Mbaud.

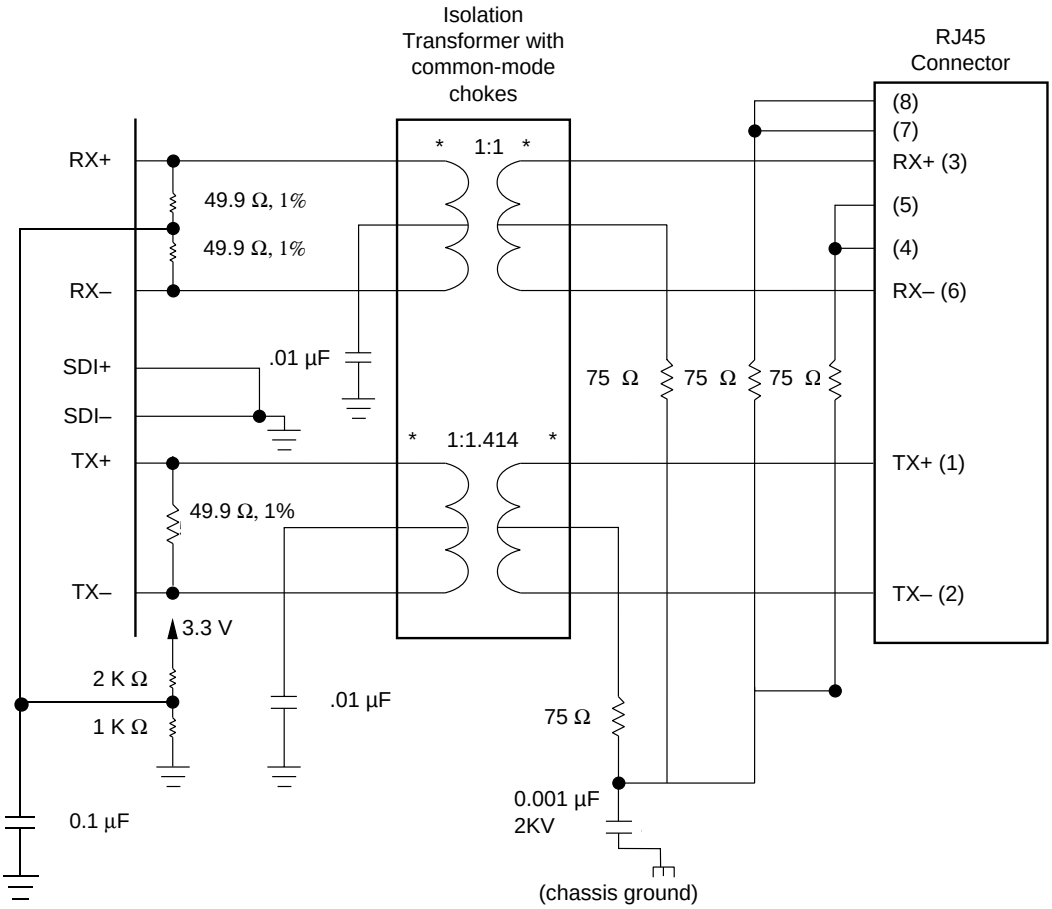
Receiving from the physical medium through the PMD device, the PDX accepts encoded PECL NRZI signal levels at the RX± inputs. The receiver circuit recovers data from the input stream by regenerating clocking information embedded in the serial stream. The recovered clock is called RSCLK (an internal signal). The PDX then clocks the unframed symbol (5 bits) to the descrambler interface on the falling edge of RSCLK.

The PDX receiver uses advanced circuit techniques to extract encoded clock information from the serial input stream and recovers the data. Its operating frequency is established by the reference clock at 25 MHz. The PDX is capable of recovering data correctly within ±1000 ppm of the 25-MHz clock signal (which exceeds the frequency range defined by the 100BASE-X specification). The 100BASE-X 4B/5B encoding scheme ensures run-length limitation and adequate transition density of the encoded data stream, while TP-PMD achieves this on a statistical basis through data scrambling. The PDX clock recovery circuit is designed to tolerate a worst-case run-length of 60-bits in order to function correctly with both fiber-optic and twisted-pair PMDs.

The PDX receiver has input jitter tolerance characteristics that meet or exceed the recommendations of Physical Layer Medium Dependent (PMD) 100BASE-X document. Typically, at 125 Mbaud (8 ns/bit), the peak-to-peak Duty-Cycle Distortion (DCD) tolerance is 1.4 ns, the peak-to-peak Data Dependent Jitter (DDJ) tolerance is 2.2 ns, and the peak-to-peak Random Jitter (RJ) tolerance is 2.27 ns. The total combined peak-to-peak jitter tolerance is typically 5 ns with a bit error rate (BER) better than 2.5×10^{-10}.

Medium Dependent Interface

The Am79C973/Am79C975 device connects directly to low cost magnetics modules for interface to twisted pair media (UTP and/or STP). The TX± and RX± pins provide the interface for both 10BASE-T and 100BASE-TX allowing the use of a 1:1.41 (transmit) and 1:1 (receive) transformer with single primary and secondary windings. No filtering is required in the magnetics module. Refer to Figure 37 for recommended termination.



Notes: *1. The isolation transformers include common-mode chokes.*
2. Consult magnetics vendors for appropriate termination schemes.

21510D-42

Figure 37. TX± and RX± Termination

10BASE-T Block

The 10BASE-T block consists of the following sub-blocks:

- Transmit Process
- Receive Process
- Interface Status
- Collision Detect Function
- Jabber Function
- Reverse Polarity Detect

Refer to Figure 38 for the 10BASE-T block diagram.

Twisted Pair Transmit Function

Data transmission over the 10BASE-T medium requires use of the integrated 10BASE-T MAU and uses the differential driver circuitry on the TX± pins.

TX± is a differential twisted-pair driver. When properly terminated, TX± will meet the transmitter electrical requirements for 10BASE-T transmitters as specified in IEEE 802.3, Section 14.3.1.2. The load is a twisted pair cable that meets IEEE 802.3, Section 14.4.

The TX± signal is filtered on the chip to reduce harmonic content per Section 14.3.2.1 (10BASE-T). Since filtering is performed in silicon, TX± can be connected directly to a standard transformer. External filtering modules are not needed.

Twisted Pair Receive Function

The RX± port is a differential twisted-pair receiver. When properly terminated, the RX± port will meet the electrical requirements for 10BASE-T receivers as specified in IEEE 802.3, Section 14.3.1.3. The receiver has internal filtering and does not require external filter modules or common mode chokes.

Signals appearing at the RX± differential input pair are routed to the internal decoder. The receiver function meets the propagation delays and jitter requirements specified by the 10BASE-T Standard. The receiver squelch level drops to half its threshold value after unsquelch to allow reception of minimum amplitude sig-

nals and to mitigate carrier fade in the event of worst case signal attenuation and crosstalk noise conditions.

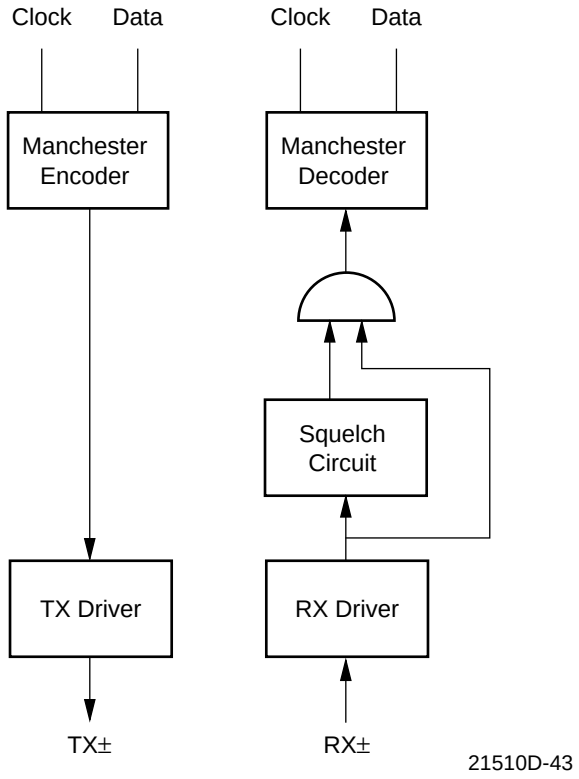


Figure 38. 10BASE-T Transmit and Receive Data Paths

Twisted Pair Interface Status

The Am79C973/Am79C975 device will power up in the Link Fail state. The Auto-Negotiation algorithm will apply to allow it to enter the Link Pass state.

In the Link Pass state, receive activity which passes the pulse width/amplitude requirements of the RX± inputs, will cause the PCS Control block to assert Carrier Sense (CRS) signal at the internal MII interface. Collision would cause the PCS Control block to assert Carrier Sense (CRS) and Collision (COL) signal at the internal MII. In the Link Fail state, this block would cause the PCS Control block to de-assert Carrier Sense (CRS) and Collision (COL).

In jabber detect mode, this block would cause the PCS Control block to assert the COL pin at the MII, and allow the PCS Control block to assert or de-assert the CRS pin to indicate the current state of the RX± pair. If there is no receive activity on RX±, this block would cause the PCS Control block to assert only the COL pin at the internal MII. If there is RX± activity, this block would cause the PCS Control block to assert both COL and CRS at the internal MII.

Collision Detect Function

Simultaneous activity (presence of valid data signals) from both the internal encoder transmit function and the twisted pair RX± pins constitutes a collision, thereby causing the PCS Control block to assert the COL pin at the internal MII.

Jabber Function

The Jabber function inhibits the 10BASE-T twisted pair transmit function of the Am79C973/Am79C975 device if the TX± circuits are active for an excessive period (20-150 ms). This prevents one port from disrupting the network due to a *stuck-on* or faulty transmitter condition. If the maximum transmit time is exceeded, the data path through the 10BASE-T transmitter circuitry is disabled (although Link Test pulses will continue to be sent). The PCS Control block also asserts the COL pin at the internal MII and sets the Jabber Detect bit in Register 1. Once the internal transmit data stream from the MENDEC stops, an *unjab* time of 250-750 ms will elapse before this block causes the PCS Control block to de-assert the COL indication and re-enable the transmit circuitry.

When jabber is detected, this block will cause the PCS control block to assert the COL pin and allow the PCS Control block to assert or de-assert the CRS pin to indicate the current state of the RX± pair. If there is no receive activity on RX±, this block causes the PCS Control block to assert only the COL pin at the internal MII. If there is RX± activity, this block will cause the PCS Control block to assert both COL and CRS at the internal MII.

Reverse Polarity Detect

The polarity for 10BASE-T signals is set by reception of Normal Link Pulses (NLP) or packets. Polarity is locked, however, by incoming packets only. The first NLP received when trying to bring the link up will be ignored, but it will set the polarity to the correct state. The reception of two consecutive packets will cause the polarity to be locked, based on the polarity of the ETD. In order to change the polarity once it has been locked, the link must be brought down and back up again.

Auto-Negotiation

The object of the Auto-Negotiation function is to determine the abilities of the devices sharing a link. After exchanging abilities, the Am79C973/Am79C975 device and remote link partner device acknowledge each other and make a choice of which advertised abilities to support. The Auto-Negotiation function facilitates an ordered resolution between exchanged abilities. This exchange allows both devices at either end of the link to take maximum advantage of their respective shared abilities.

The Am79C973/Am79C975 device implements the transmit and receive Auto-Negotiation algorithm as defined in IEEE 802.3u, Section 28. The Auto-Negotiation algorithm uses a burst of link pulses called Fast Link Pulses (FLPs). The burst of link pulses are spaced between 55 and 140 µs so as to be ignored by the standard 10BASE-T algorithm. The FLP burst conveys information about the abilities of the sending device. The receiver can accept and decode an FLP burst to learn the abilities of the sending device. The link pulses transmitted conform to the standard 10BASE-T template. The device can perform auto-negotiation with reverse polarity link pulses.

The Am79C973/Am79C975 device uses the Auto-Negotiation algorithm to select the type connection to be established according to the following priority: 100BASE-TX full duplex, 100BASE-T4, 100BASE-TX half-duplex, 10BASE-T full duplex, 10BASE-T half-duplex. The Am79C973/Am79C975 device does not support 100BASE-T4 connections.

The Auto-Negotiation algorithm is initiated when one or the following events occurs: Auto-Negotiation enable bit is set, or reset, or soft reset, or transition to link fail state (when Auto-Negotiation enable bit is set), or Auto-Negotiation restart bit is set. The result of the Auto-Negotiation process can be read from the status register (Summary Status Register, ANR24).

The Am79C973/Am79C975 device supports Parallel Detection for remote legacy devices which do not support the Auto-Negotiation algorithm. In the case that a 100BASE-TX only device is connected to the remote end, the Am79C973/Am79C975 device will see scrambled idle symbols and establish a 100BASE-TX only connection. If NLPs are seen, the Am79C973/Am79C975 device will establish a 10BASE-T connection.

By default, the link partner must be at least 10BASE-T half-duplex capable. The Am79C973/Am79C975 controller can automatically negotiate with the network and yield the highest performance possible without software support. See the section on *Network Port Manager* for more details.

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

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 the Am79C973/Am79C975 device unless the DANAS (BCR32, bit 10) is selected.*

Soft Reset Function

The PHY Control Register (ANR0) incorporates the soft reset function (bit 15). It is a read/write register and is self-clearing. Writing a 1 to this bit causes a soft reset. When read, the register returns a 1 if the soft reset is still being performed; otherwise, it is cleared to 0. *Note that the register can be polled to verify that the soft reset has terminated*. Under normal operating conditions, soft reset will be finished in 150 clock cycles.

Soft reset only resets the 10/100 PHY unit registers to default values (some register bits retain their previous values). Refer to the individual registers for values after a soft reset. Soft reset does not reset the PDX block nor the management interface.

Soft reset is required when changing the value of the SDISSCR (scrambling/descrambling) bit. After soft reset, the register will retain the previous value written.

External Address Detection Interface

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 as it arrives at the Am79C973/Am79C975 controller, to compare the captured address with a table of stored addresses or identifiers, and then to determine whether or not the Am79C973/Am79C975 controller should accept the packet.

If an address match is detected by comparison with either the Physical Address or Logical Address Filter registers contained within the Am79C973/Am79C975 controller or the frame is of the type 'Broadcast', then the frame will be accepted regardless of the condition of $\overline{\text{EAR}}$. When the EADISEL bit of BCR2 is set to 1 and the Am79C973/Am79C975 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 $\overline{\text{EAR}}$ 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 Am79C973/Am79C975 control-

ler, 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 Am79C973/Am79C975 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. In order for the $\overline{EAR}$ pin to be functional in full-duplex mode, FDRPAD bit (BCR9, bit 2) needs to be set. 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-powerdown mode in that the Am79C973/Am79C975 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 Frames
1	X	No timing requirements	All received frames
0	1	No timing requirements	All received frames
0	0	Low for two bit times plus 10 ns	Frame rejected if in address match mode

External Address Detection Interface: MII Snoop Mode

The MII Snoop mode provides all necessary data and clock signals needed for the EADI interface. Data for the EADI is the RXD[3:0] receive data provided to the internal MII. 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 data arrives in nibbles and can be at a rate of 25 MHz or 2.5 MHz.

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 Am79C973/Am79C975 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 $\overline{EAR}$ pin should be driven LOW by the external address comparison logic to reject a frame.

External Address Detection Interface: Receive Frame Tagging

The Am79C973/Am79C975 controller supports receive frame tagging in MII Snoop mode. The receive frame tagging implementation is a two-wire chip interface in addition to the existing EADI.

The Am79C973/Am79C975 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 Snoop 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.

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 Am79C973/Am79C975 controller receives the end of the current receive frame.

In the MII Snoop 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 39.

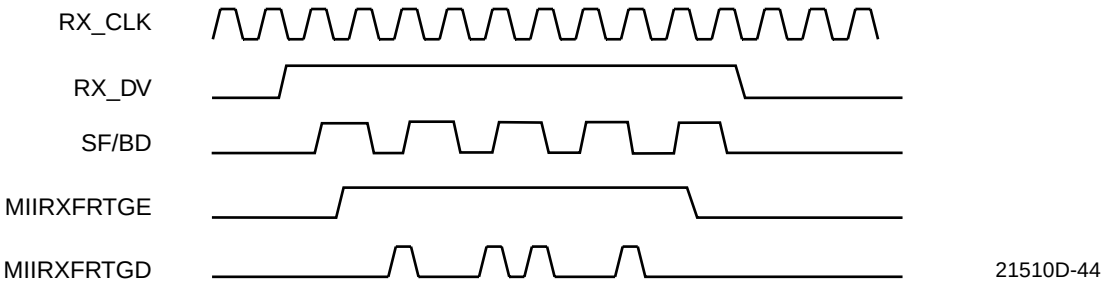


Figure 39. Receive Frame Tagging

Expansion Bus Interface

The Am79C973/Am79C975 controller contains an Expansion Bus Interface that supports Flash and EPROM devices as boot devices, as well as provides read/write access to Flash or EPROM.

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 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 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 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 $\overline{\text{EBWE}}$ is connected to the $\overline{\text{WE}}$ of the Flash device.

The Expansion Data Bus is configured for 8-bit byte access during EPROM/Flash accesses. During EPROM/Flash accesses, EBD[7:0] provides the data byte. See Figure 40, Figure 41, and Figure 42.

Expansion ROM - Boot Device Access

The Am79C973/Am79C975 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 Am79C973/Am79C975 controller is functionally equivalent to the PCnet-PCI II controller with Expansion ROM. See Figure 41 and Figure 42.

The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller starts driving the lower 8 bits of the Expansion ROM address on EBUA_EBA[7:0].

The time that the Am79C973/Am79C975 controller waits for data to be valid is programmable. ROMTMG (BCR18, bits 15-12) defines the time from when the Am79C973/Am79C975 controller drives EBUA_EBA[7:0] with the lower 8 bits of the Expansion ROM address to when the Am79C973/Am79C975 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 Ex-

pansion ROM data are stored in holding registers. One clock cycle after the last data byte is available, the Am79C973/Am79C975 controller asserts $\overline{TRDY}$.

The access time for the Expansion ROM or the EBDATA (BCR30) device (tACC) during read operations can be calculated by subtracting the clock to output delay for the EBUA_EBA[7:0] outputs (tv_A_D) and by subtracting the input to clock setup time for the EBD[7:0] inputs (ts_D) from the time defined by ROMTMG:

tACC = ROMTMG * CLK period *CLK_FAC - (tv_A_D) - (ts_D)

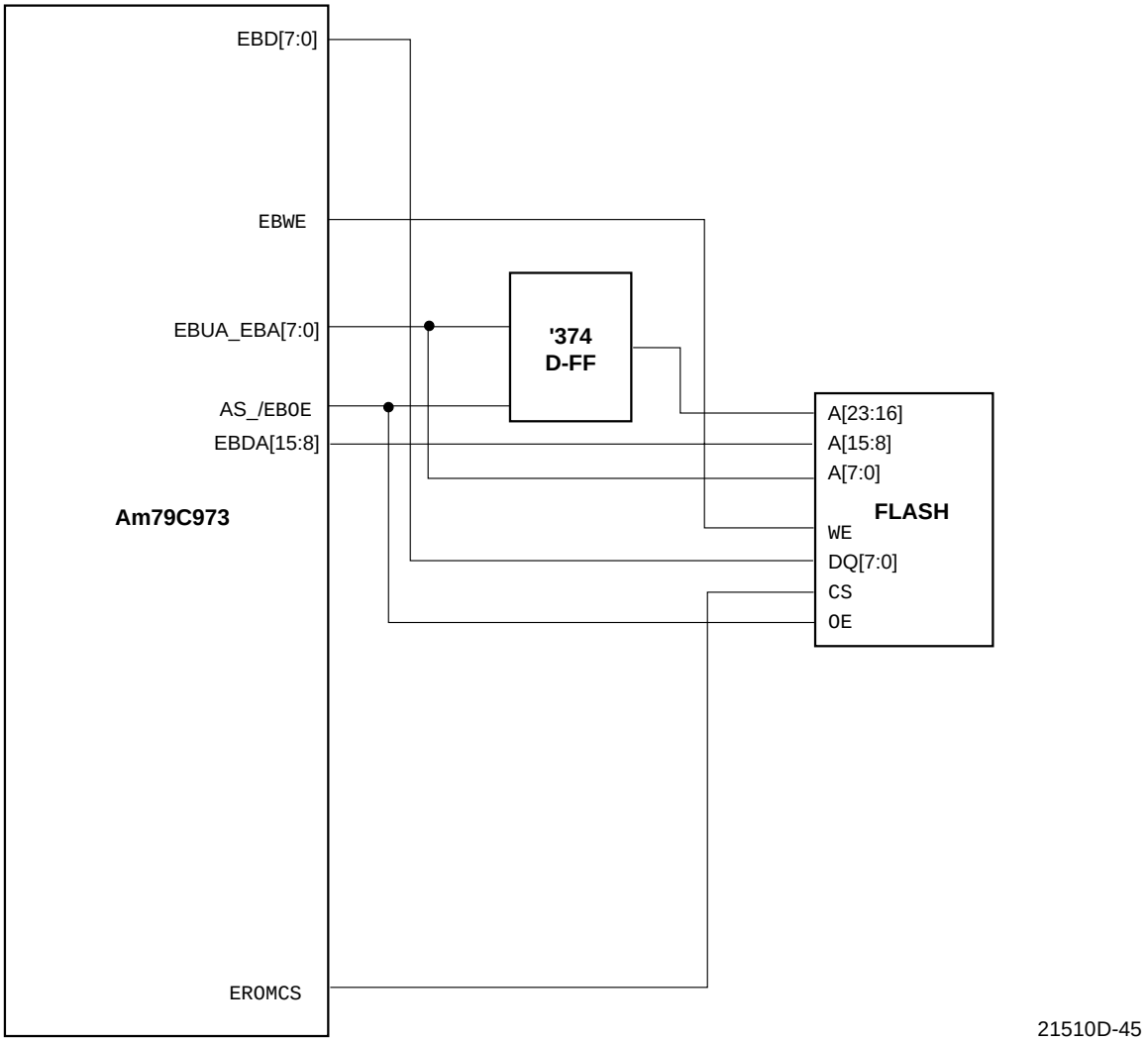


Figure 40. Flash Configuration for the Expansion Bus

The access time for the Expansion ROM or for the EBDATA (BCR30) device (tACC) during write operations can be calculated by subtracting the clock to output delay for the EBUA EBA[7:0] outputs (tv_A_D) and by adding the input to clock setup time for Flash/EPRO inputs (ts_D) from the time defined by ROMTMG:

tACC = ROMTMG * CLK period * CLK_FAC - (tv_A_D) - (ts_D)

The timing diagram in Figure 43 assumes the default programming of ROMTMG (1001b = 9 CLK). After reading the first byte, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/

Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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.

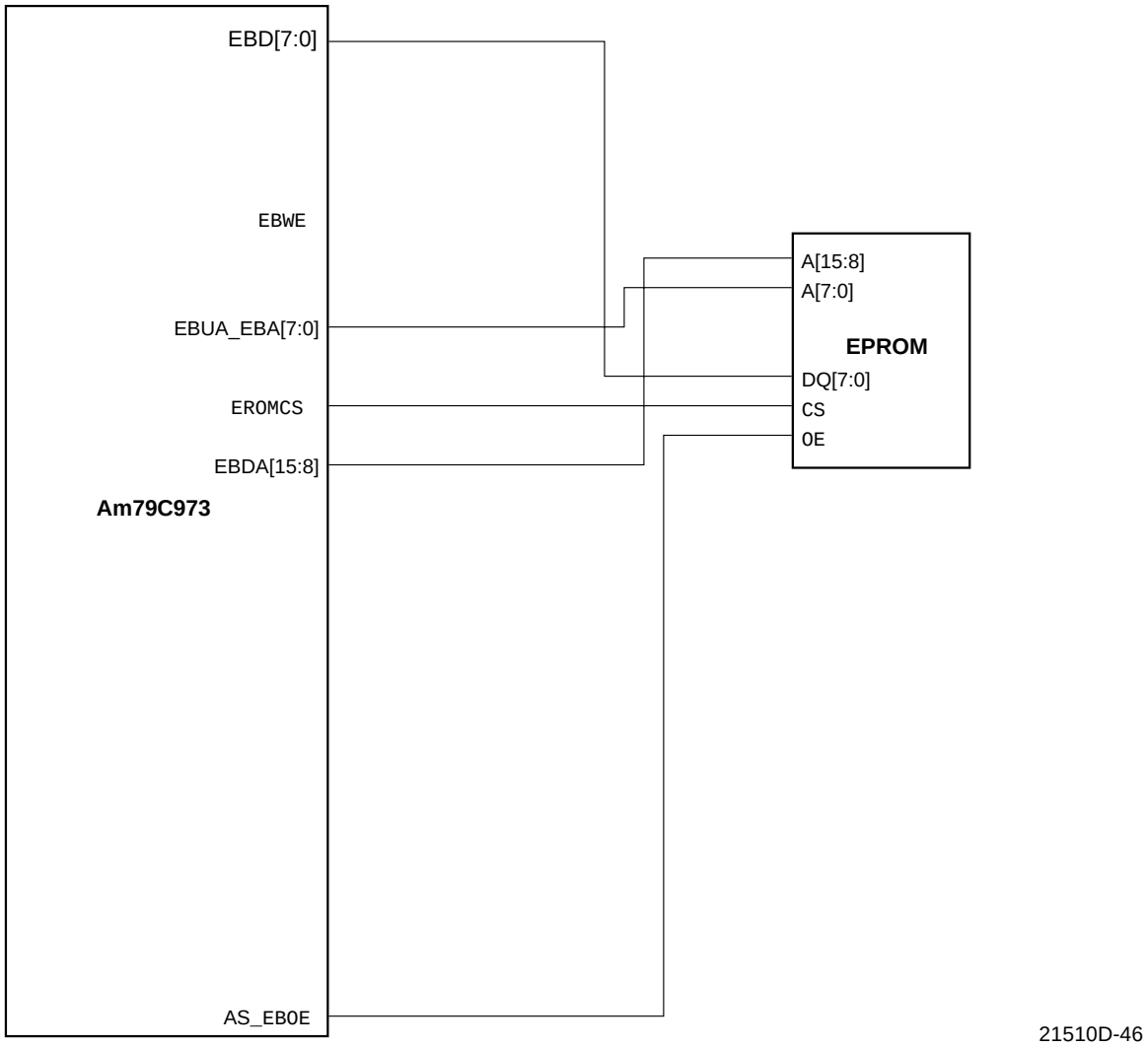


Figure 41. EPROM Only Configuration for the Expansion Bus (64K EPROM)

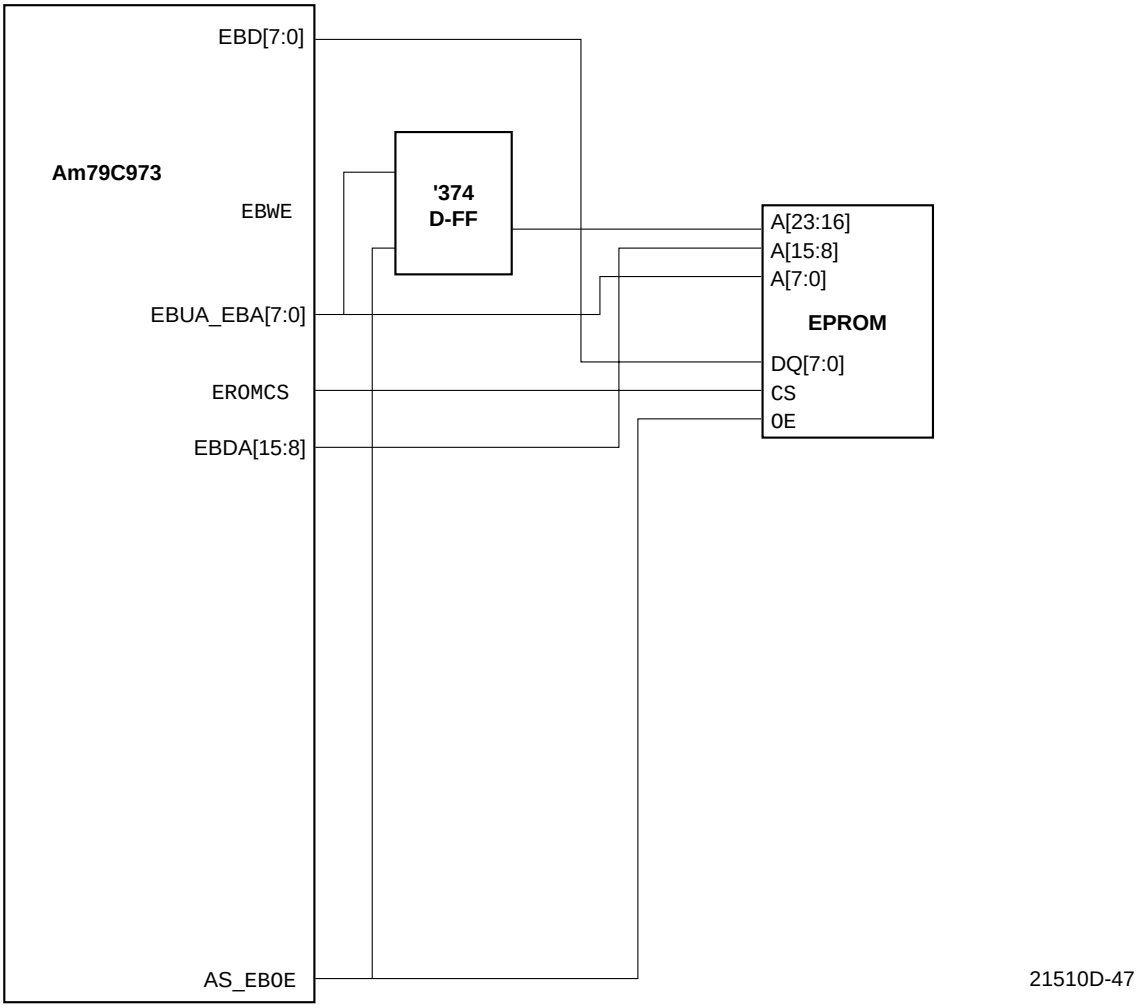


Figure 42. EPROM Only Configuration for the Expansion Bus (>64K EPROM)

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 Am79C973/ Am79C975 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).

Direct Flash Access

Am79C973/Am79C975 controller supports Flash as an Expansion ROM device, as well as providing a read/ write data path to the Flash. The Am79C973/ Am79C975 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 Am79C973/Am79C975 controller will drive EBUA_EBA[7:0] with the most significant address byte at the same time the Am79C973/Am79C975 controller will drive AS_$\overline{\text{EBOE}}$ high to strobe the address in the external ‘374 (D flip-flop). On the next clock, the Am79C973/Am79C975 controller will drive EBDA[15:8] and EBUA_EBA[7:0] with the middle and least significant address bytes.

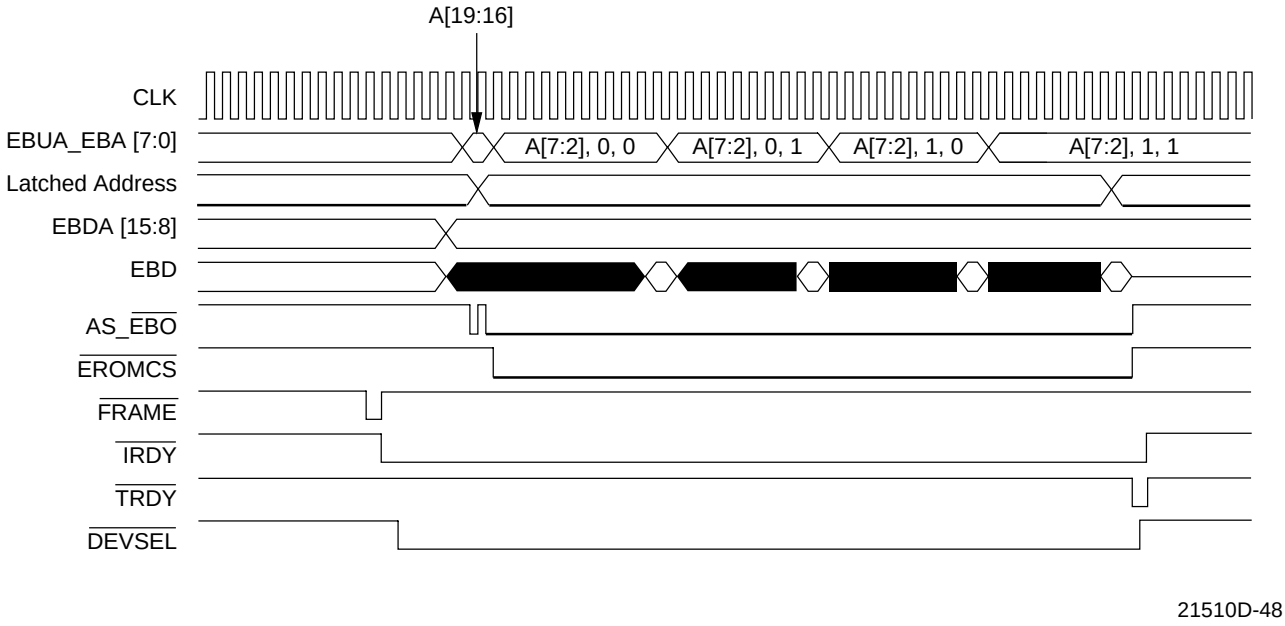


Figure 43. Expansion ROM Bus Read Sequence

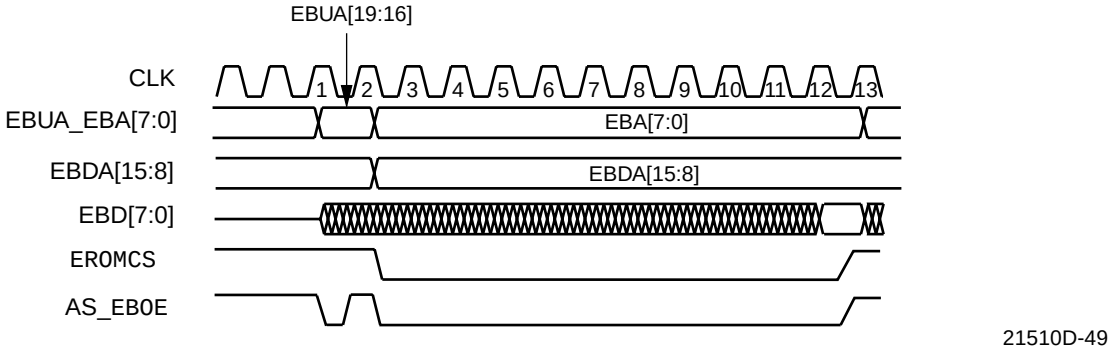


Figure 44. Flash Read from Expansion Bus Data Port

The EROMCS is driven low for the value ROMTMG + 1. Figure 44 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 Am79C973/Am79C975 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. See Figure 45.

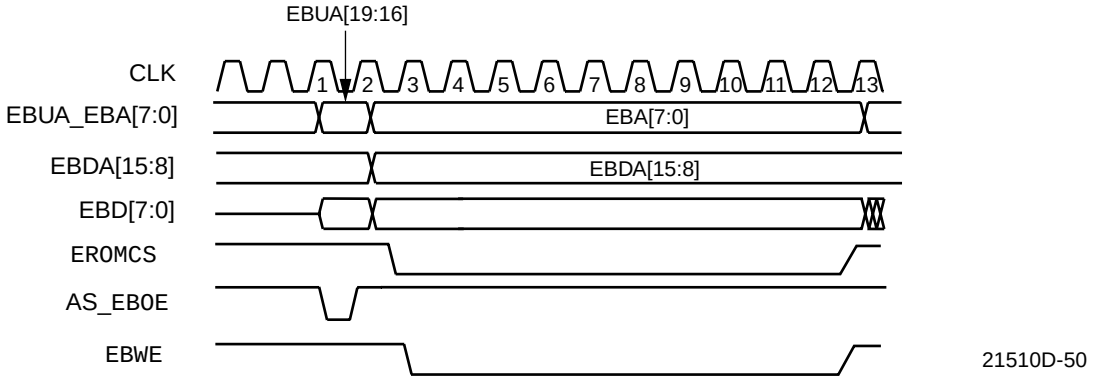


Figure 45. Flash Write from Expansion Bus Data Port

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 $\overline{\text{EBWE}}$ and the data is latched on the rising edge of $\overline{\text{EBWE}}$. The rising edge of $\overline{\text{EBWE}}$ begins programming.

Upon executing the AMD Flash Embedded Program Algorithm command sequence, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller is not required to provide any controls or timings during these operations. The automatic erase begins on the rising edge of the last $\overline{\text{EBWE}}$ 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will support only a single sector erase per command and not concurrent sector erasures. The Am79C973/Am79C975 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 Am79C973/Am79C975 controller access mechanism. The Am79C973/Am79C975 controller will also support only Flash devices that do not require data hold times after write operations. See Table 12.

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 Am79C973/Am79C975 controller supports internal SRAM as a FIFO extension as well as providing a read/write data path to the SRAM. The Am79C973/Am79C975 controller contains 12 Kbytes of SRAM.

Internal SRAM Configuration

The SRAM_SIZE (BCR25, bits 7-0) programs the size of the SRAM. SRAM_SIZE can be programmed to a smaller value than 12 Kbytes.

The SRAM should be programmed on a 512-byte boundary. However, there should be no accesses to the RAM space while the Am79C973/Am79C975 controller is running. The Am79C973/Am79C975 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 Am79C973/Am79C975 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 externally provided time base clock. When the PCI or time base 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 Am79C973/Am79C975 controller is stopped. Otherwise, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller does not support an SRAM frequency above 33 MHz regardless of the clock and clock factor used.

No SRAM Configuration

If the SRAM_SIZE (BCR25, bits 7-0) value is 0 in the SRAM size register, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller. See Figure 46.

Low Latency Receive Configuration

If the LOLATRX (BCR27, bit 4) bit is set to 1, then the Am79C973/Am79C975 controller will configure itself for a low latency receive configuration. In this mode, SRAM is required at all times. If the SRAM_SIZE (BCR25, bits 7-0) value is 0, the Am79C973/Am79C975 controller will not configure for low latency receive mode. The Am79C973/Am79C975 controller will provide a fast path on the receive side bypassing the SRAM. 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 Am79C973/Am79C975 controller has received 16 bytes from the network, it will start a DMA request to the PCI Bus Interface Unit. The Am79C973/Am79C975 controller will not wait for the first 64 bytes to pass to check for collisions in Low Latency Receive mode. The Am79C973/Am79C975 controller must be in STOP before switching to this mode. See Figure 47.

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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller is in STOP or SPND (FASTSPNDE is set to 0) mode.

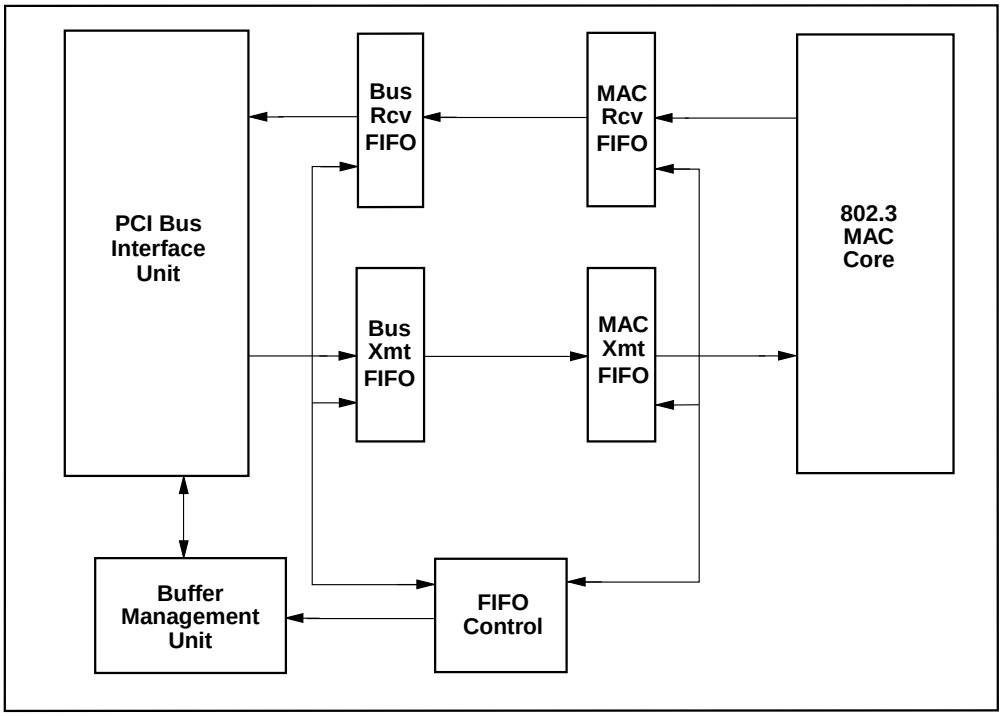


Figure 46. Block Diagram No SRAM Configuration

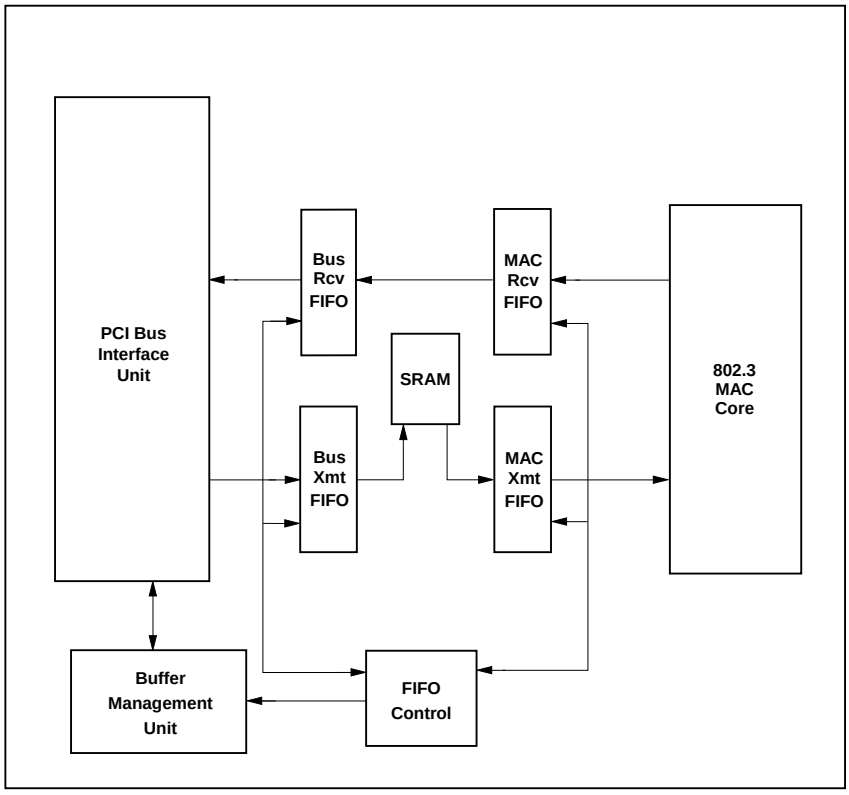


Figure 47. Block Diagram Low Latency Receive Configuration

EEPROM Interface

The Am79C973/Am79C975 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 Am79C973/Am79C975 controller will read the contents of the EEPROM that is attached to the interface. Because of this automatic-read capability of the Am79C973/Am79C975 controller, an EEPROM can be used to program many of the features of the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller. Access to the Am79C973/Am79C975 configuration space, the Expansion ROM or any I/O resource is not possible during the EEPROM read operation. The Am79C973/Am79C975 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 Am79C973/Am79C975 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 68 bytes of the EEPROM should be FFh.

If no EEPROM is present at the time of the automatic read operation, the Am79C973/Am79C975 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.

EEPROM Auto-Detection

The Am79C973/Am79C975 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 Am79C973/Am79C975 controller will sample the value of the EESK/$\overline{\text{LED1}}$/SFBD pin. If the sampled value is a 1, then the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 PHY Control and Status
- BCR33 PHY Address

- BCR35 PCI Vendor ID
- BCR36 PCI Power Management Capabilities (PMC) Alias Register
- BCR37 PCI DATA Register Zero (DATA0) Alias Register
- BCR38 PCI DATA Register One (DATA1) Alias Register
- BCR39 PCI DATA Register Two (DATA2) Alias Register
- BCR40 PCI DATA Register Three (DATA3) Alias Register
- BCR41 PCI DATA Register Four (DATA4) Alias Register
- BCR42 PCI DATA Register Five (DATA5) Alias Register
- BCR43 PCI DATA Register Six (DATA6) Alias Register
- BCR44 PCI DATA Register Seven (DATA7) Alias Register
- BCR45 OnNow Pattern Matching Register 1
- BCR46 OnNow Pattern Matching Register 2
- BCR47 OnNow Pattern Matching Register 3
- CSR12 Physical Address Register 0
- CSR13 Physical Address Register 1
- CSR14 Physical Address Register 2
- CSR116 OnNow Miscellaneous

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.

EEPROM MAP

The automatic EEPROM read operation will access 41 words (i.e., 82 bytes) of the EEPROM. The format of the EEPROM contents is shown in Table 13 (next page), beginning with the byte that resides at the lowest EEPROM address.

Note: *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.*

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 51h) 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 82 bytes of EEPROM data equals the value FFh. The checksum adjust byte is needed by the Am79C973/Am79C975 controller in order to verify that the EEPROM content has not been corrupted.

LED Support

The Am79C973/Am79C975 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 Am79C973/Am79C975 controller. Use of buffering can be avoided if a low power LED is used.

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, Receive Match, Receive Status, Magic Packet, Disable Transceiver, and Transmit Status.

Table 13. Am79C973 EEPROM Map

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 IS0 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	CSR116[15:8] (OnNow Misc. Config).	06h	CSR116[7:0] (OnNow Misc. Config.)
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	BCR36[15:8] (Conf. Space. byte 43h alias)	30h	BCR36[7:0] (Conf. Space byte 42h alias)
19h	33h	BCR37[15:8] (DATA_SCALE alias 0)	32h	BCR37[7:0] (Conf. Space byte 47h0alias)
1Ah	35h	BCR38[15:8] (DATA_SCALE alias 1)	34h	BCR38[7:0] (Conf. Space byte 47h1alias)
1Bh	37h	BCR39[15:8] (DATA_SCALE alias 2)	36h	BCR39[7:0] (Conf. Space byte 47h2alias)
1Ch	39h	BCR40[15:8] (DATA_SCALE alias 3)	38h	BCR40[7:0] (Conf. Space byte 47h3alias)
1Dh	3Bh	BCR41[15:8] (DATA_SCALE alias 4)	3Ah	BCR41[7:0] (Conf. Space byte 47h4alias)
1Eh	3Dh	BCR42[15:8] (DATA_SCALE alias 0)	3Ch	BCR42[7:0] (Conf. Space byte 47h5alias)
1Fh	3Fh	BCR43[15:8] (DATA_SCALE alias 0)	3Eh	BCR43[7:0] (Conf. Space byte 47h6alias)
20h	41h	BCR44[15:8] (DATA_SCALE alias 0)	40h	BCR44[7:0] (Conf. Space byte 47h7alias)
21h	43h	BCR48[15:8]Reserved location:must be 00h	42h	BCR48[7:0]Reserved location: must be 00h
22h	45h	BCR49[15:8]Reserved location:must be 00h	44h	BCR49[7:0]Reserved location: must be 00h
23h	47h	BCR50[15:8]Reserved location:must be 00h	46h	BCR50[7:0]Reserved location: must be 00h
24h	49h	BCR51[15:8]Reserved location:must be 00h	48h	BCR51[7:0]Reserved location: must be 00h
25h	4Bh	BCR52[15:8]Reserved location:must be 00h	4Ah	BCR52[7:0]Reserved location: must be 00h
26h	4Dh	BCR53[15:8]Reserved location:must be 00h	4Ch	BCR53[7:0]Reserved location: must be 00h
27h	4Fh	BCR54[15:8]Reserved location:must be 00h	4Eh	BCR54[7:0]Reserved location: must be 00h
28h	51h	Checksum adjust byte for the 82 bytes of the EEPROM contents, checksum of the 82 bytes of the EEPROM should total to FFh	50h	BCR55[7:0]Reserved location: must be 00h
Empty locations – Ignored by device				
3Eh	7Dh	Reserved for Boot ROM usage	7Ch	Reserved for Boot ROM usage
3Fh	7Fh	Reserved for Boot ROM usage	7Eh	Reserved for Boot ROM usage

Note: **Lowest EEPROM address.*

Table 14. Am79C975 EEPROM Map

Word Addr.	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	CSR116[15:8] (OnNow Misc. Config).	06h	CSR116[7:0] (OnNow Misc. Config.)
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	BCR22[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	BCR36[15:8] (Conf. Sp. byte 43h alias)	30h	BCR36[7:0] (Conf. Sp. byte 42h alias)
19h	33h	BCR37[15:8] (DATA_SCALE alias0)	32h	BCR37[7:0] (Conf. Sp. byte 47h0 alias)
1Ah	35h	BCR38[15:8] (DATA_SCALE alias 1)	34h	BCR38[7:0] (Conf. Sp. byte 47h1 alias)
1Bh	37h	BCR39[15:8] (DATA_SCALE alias 2)	36h	BCR39[7:0] (Conf. Sp. byte 47h2 alias)
1Ch	39h	BCR40[15:8] (DATA_SCALE alias 3)	38h	BCR40[7:0] (Conf. Sp. byte 47h3 alias)
1Dh	3Bh	BCR41[15:8] (DATA_SCALE alias 4)	3Ah	BCR41[7:0] (Conf. Sp. byte 47h4 alias)
1Eh	3Dh	BCR42[15:8] (DATA_SCALE alias 5)	3Ch	BCR42[7:0] (Conf. Sp. byte 47h5 alias)
1Fh	3Fh	BCR43[15:8] (DATA_SCALE alias 6)	3Eh	BCR43[7:0] (Conf. Sp. byte 47h6 alias)
20h	41h	BCR44[15:8] (DATA_SCALE alias 7)	40h	BCR44[7:0] (Conf. Sp. byte 47h7 alias)
21h	43h	BCR48[15:8] N_IP_ADR[15:8]	42h	BCR48[7:0] N_IP_ADR[7:0]
22h	45h	BCR49[15:8] N_IP_ADR[31:24]	44h	BCR49[7:0] N_IP_ADR[23:16]
23h	47h	BCR50[15:8] M_IEEE_ADR[15:8]	46h	BCR50[7:0] M_IEEE_ADR[7:0]
24h	49h	BCR51[15:8] M_IEEE_ADR[31:24]	48h	BCR51[7:0] M_IEEE_ADR[23:16]
25h	4Bh	BCR52[15:8] M_IEEE_ADR[47:40]	4Ah	BCR52[7:0] M_IEEE_ADR[39:32]
26h	4Dh	BCR53[15:8] M_IP_ADR[15:8]	4Ch	BCR53[7:0] M_IP_ADR[7:0]
27h	4Fh	BCR54[15:8] M_IP_ADR[31:24]	4Eh	BCR54[7:0] M_IP_ADR[23:16]
28h	51h	Checksum adjust byte for the 82 bytes of the EEPROM contents, checksum of the 82 bytes of the EEPROM should total to FFh	50h	BCR55[7:0] SMIU Slave Address
3Eh	7Dh	Reserved for Boot ROM usage	7Ch	Reserved for Boot ROM usage
3Fh	7Fh	Reserved for Boot ROM usage	7Eh	Reserved for Boot ROM usage

Note: ** Lowest EEPROM address.*

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 15.

Table 15. LED Default Configuration

LED Output	Indication	Driver Mode	Pulse Stretch
LED0	Link Status	Open Drain - Active Low	Enabled
LED1	Receive Status	Open Drain - Active Low	Enabled
LED2	--	Open Drain - Active Low	Enabled
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 48.

Power Savings Mode

Power Management Support

The Am79C973/Am79C975 controller supports power management as defined in the PCI Bus Power Management Interface Specification V1.1 and Network Device Class Power Management Reference Specification V1.0.These specifications define the network device power states, PCI power management interface including the Capabilities Data Structure and power management registers block definitions, power management events, and OnNow network Wake-up events. In addition, the Am79C973/Am79C975 controller supports legacy power management schemes, such as Remote Wake-Up (RWU) mode. When the system is in RWU mode, PCI bus power is on, the PCI clock may be slowed down or stopped, and the wake-up output pin may drive the CPU's System Management Interrupt (SMI) line.

Auxiliary Power

The Am79C973/Am79C975 uses the AUXDET pin to detect whether it is powered by an auxiliary power supply that is always on or by the PCI power supply that goes down during power saving modes.

If bit 15 of PMC is zero, indicating that $\overline{\text{PME}}$ assertion in D3cold is not supported, the PME_Status and PME_En bits of the PMCSR register will be reset by a PCI bus reset (assertion of $\overline{\text{RST}}$ pin). This reset will actually occur after the EEPROM read following the reset is complete to allow the controller to be configured.

To fully satisfy the requirements of the PCI power management specification in an adapter card configuration, the AUXDET pin should be connected directly to the auxiliary power supply and also to ground through a resistor. This will sense the presence of the auxiliary power and correctly report the capability of asserting $\overline{\text{PME}}$ in D3cold.

For hardwired configurations where auxiliary power is know to be always available or never available, the AUXDET input may be disabled by connecting it directly or through a resistor to VDD. This will allow BCR36 bit 15 to directly control PMC bit 15.

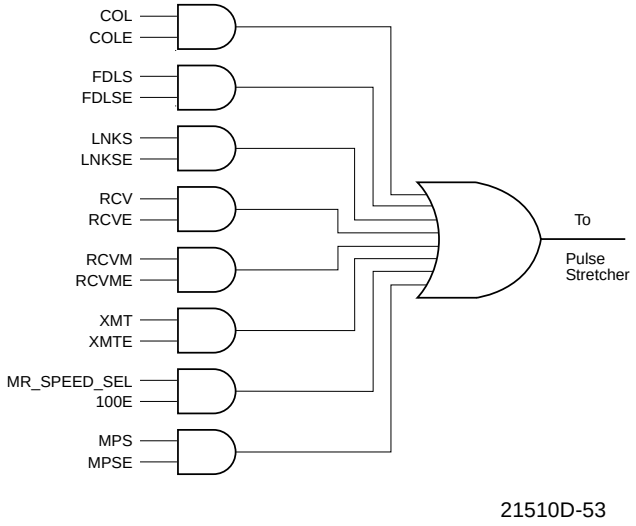


Figure 48. LED Control Logic

The general scheme for the Am79C973/Am79C975 power management is that when a PCI Wake-up event is detected, a signal is generated to cause hardware external to the Am79C973/Am79C975 device to put the computer into the working (S0) mode.

The Am79C973/Am79C975 device supports three types of wake-up events:

1. Magic Packet Detect
2. OnNow Pattern Match Detect
3. Link State Change

Figure 49 shows the relationship between these Wake-up events and the various outputs used to signal to the external hardware.

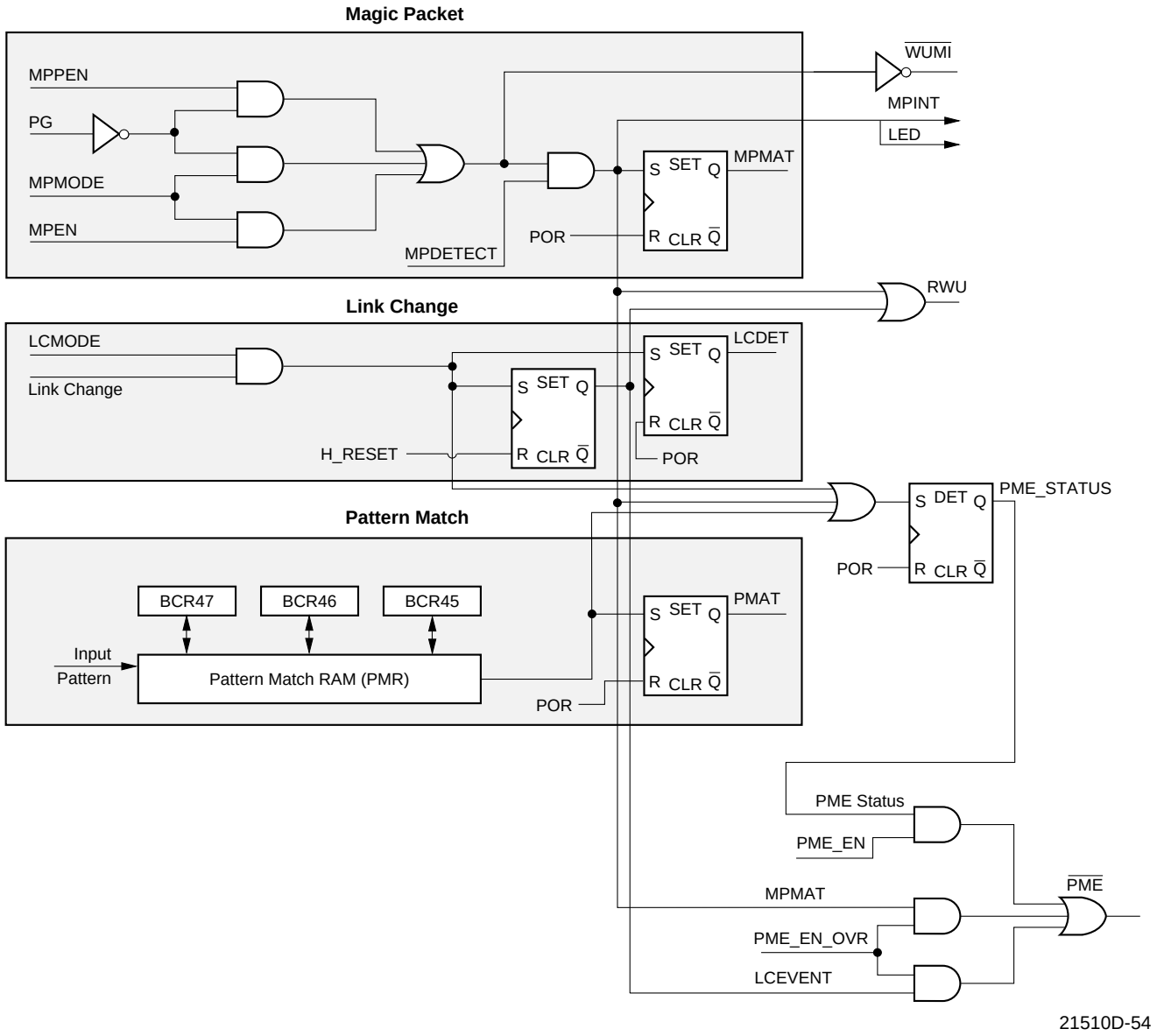


Figure 49. OnNow Functional Diagram

OnNow Wake-Up Sequence

The system software enables the $\overline{\text{PME}}$ pin by setting the PME_EN bit in the PMCSR register (PCI configuration registers, offset 44h, bit 8) to 1. When a Wake-up event is detected, the Am79C973/Am79C975 controller sets the PME_STATUS bit in the PMCSR register (PCI configuration registers, offset 44h, bit 15). Setting this bit causes the $\overline{\text{PME}}$ signal to be asserted. Assertion of the $\overline{\text{PME}}$ signal causes external hardware to wake up the CPU. The system software then reads the PMCSR register of every PCI device in the system to determine which device asserted the $\overline{\text{PME}}$ signal.

When the software determines that the signal came from the Am79C973/Am79C975 controller, it writes to the device's PMCSR to put the device into power state D0. The software then writes a 0 to the PME_STATUS bit to clear the bit and turn off the $\overline{\text{PME}}$ signal, and it calls the device's software driver to tell it that the device is now in state D0. The system software can clear the PME_STATUS bit either before, after, or at the same time that it puts the device back into the D0 state.

Link Change Detect

Link change detect is one of Wake-up events defined by the OnNow specification and is supported by the RWU mode. Link Change Detect mode is set when the LCMODE bit (CSR116, bit 8) is set either by software or loaded through the EEPROM.

When this bit is set, any change in the Link status will cause the LCDET bit (CSR116, bit 9) to be set. When

the LCDET bit is set, the RWU pin will be asserted and the PME_STATUS bit (PMCSR register, bit 15) will be set. If either the PME_EN bit (PMCSR, bit 8) or the PME_EN_OVR bit (CSR116, bit 10) are set, then the $\overline{PME}$ will also be asserted.

OnNow Pattern Match Mode

In the OnNow Pattern Match Mode, the Am79C973/ Am79C975 device compares the incoming packets with up to eight patterns stored in the Pattern Match RAM (PMR). The stored patterns can be compared with part or all of incoming packets, depending on the pattern length and the way the PMR is programmed. When a pattern match has been detected, then PMAT bit (CSR116, bit 7) is set. The setting of the PMAT bit causes the PME_STATUS bit (PMCSR, bit 15) to be set, which in turn will assert the $\overline{PME}$ pin if the PME_EN bit (PMCSR, bit 8) is set.

Pattern Match RAM (PMR)

PMR is organized as an array of 64 words by 40 bits as shown in Figure 50. The PMR is programmed indirectly through the BCRs 45, 46, and 47. When the BCR45 is written and the PMAT_MODE bit (BCR45, bit 7) is set to 1, Pattern Match logic is enabled. No bus accesses into the PMR are possible when the PMAT_MODE bit is set, and BCR46, BCR47, and all other bits in BCR45 are ignored. When PMAT_MODE is set, a read of BCR45 returns all bits undefined except for PMAT_MODE. In order to access the contents of the PMR, PMAT_MODE bit should be programmed to 0.

When BCR45 is written to set the PMAT_MODE bit to 0, the Pattern Match logic is disabled and accesses to the PMR are possible. Bits 6:0 of BCR45 specify the address of the PMR word to be accessed. Writing to BCR45 does not immediately affect the contents of the PMR. Following the write to BCR45, the PMR word addressed by the bits 6:0 of the BCR45 may be read by reading BCR45, BCR46, and BCR47 in any order. To write to the PMR word, the write to BCR45 must be followed by a write to BCR46 and a write to BCR47 in that order to complete the operation. The PMR will not actually be written until the write to BCR47 is complete.

The first two 40-bit words in this RAM serve as pointers and contain enable bits for the eight possible match patterns. The remainder of the RAM contains the match patterns and associated match pattern control bits. The byte 0 of the first word contains the Pattern Enable bits. Any bit position set in this byte enables the corresponding match pattern in the PMR, as an example if the bit 3 is set, then the Pattern 3 is enabled for matching. Bytes 1 to 4 in the first word are pointers to the beginning of the patterns 0 to 3, and bytes 1 to 4 in the second word are pointers to the beginning of the patterns 4 to 7, respectively. Byte 0 of the second word has no function associated with it. The byte 0 of the words 2 to 63 is the Control Field of the PMR. Bit 7 of this field is the End of Packet (EOP) bit. When this bit is set, it indicates the end of a pattern in the PMR. Bits 6-4 of the Control Field byte are the SKIP bits. The value of the SKIP field indicates the number of the Dwords to be skipped before the pattern in this PMR word is compared with data from the incoming frame. A maximum of seven Dwords may be skipped. Bits 3-0 of the Control Field byte are the MASK bits. These bits correspond to the pattern match bytes 3-0 of the same PMR word (PMR bytes 4-1). If bit *n* of this field is 0, then byte *n* of the corresponding pattern word is ignored. If this field is programmed to 3, then bytes 0 and 1 of the pattern match field (bytes 2 and 1 of the word) are used and bytes 3 and 2 are ignored in the pattern matching operation.

The contents of the PMR are not affected by H_RESET, S_RESET, or STOP. The contents are undefined after a power up reset (POR).

Magic Packet Mode

In Magic Packet mode, the Am79C973/Am79C975 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 Am79C973/Am79C975 controller are still possible. A Magic Packet is a frame that is addressed to the Am79C973/Am79C975 controller and contains a data sequence anywhere in its data field made up of 16 consecutive copies of the device's physical address (PADR[47:0]). The Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller supports two different modes of address detection for a Magic Packet frame. If MPPLBA (CSR5, bit 5) or EMPPLBA (CSR116, bit 6) are at their default value of 0, the Am79C973/Am79C975 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 or EMPPLBA are set to 1, the destination address of the Magic Packet frame can be unicast, multicast, or broadcast.

	BCR 47		BCR 46		BCR 45	
BCR Bit Number	15 8	7 0	15 8	7 0	15 8	
	PMR_B4	PMR_B3	PMR_B2	PMR_B1	PMR_B0	
Pattern Match RAM Address	Pattern Match RAM Bit Number 39 32	31 24	23 16	15 8	7 0	Comments
0	P3 pointer	P2 pointer	P1 pointer	P0 pointer	Pattern Enable bits	First Address
1	P7 pointer	P6 pointer	P5 pointer	P4 pointer	X	Second Address
2	Data Byte 3	Data Byte 2	Data Byte1	Data Byte 0	Pattern Control	Start Pattern P_1
2+n	Data Byte 4n+3	Date Byte 4n+2	Data Byte 4n+1	Data Byte 4n+0	Pattern Control	End Pattern P_1
J	Data Byte 3	Data Byte 2	Data Byte 1	Data Byte 0	Pattern Control	Start Pattern P_k
J+m	Data Byte 4m+3	Data Byte 4m+2	Data Byte 4m+1	Data Byte 4m+0	Pattern Control	End Pattern P_k
63						Last Address

7	6 5 4	3 2 1 0
EOP	SKIP	MASK

21510D-55

Figure 50. Pattern Match RAM

Note: *The setting of MPPLBA or EMPPLBA only effects the address detection of the Magic Packet frame. The Magic Packet's data sequence must be made up of 16 consecutive copies of the device's physical address (PADR[47:0]), regardless of what kind of destination address it has.*

There are two general methods to place the Am79C973/Am79C975 controller into the Magic Packet mode. The first is the software method. In this method, either the BIOS or other software, sets the MPMODE bit (CSR5, bit 1). Then Am79C973/ Am79C975 controller must be put into suspend mode (see description of CSR5, bit 0), allowing any current network activity to finish. Finally, either PG must be deasserted (hardware control) or MPEN (CSR5, bit 2) must be set to 1 (software control).

Note: *FASTSPNDE (CSR7, bit 15) has no meaning in Magic Packet mode.*

The second method is the hardware method. In this method, the MPPEN bit (CSR116, bit 4) is set at power up by the loading of the EEPROM. This bit can also be set by software. The Am79C973/Am79C975 controller will be placed in the Magic Packet Mode when either the PG input is deasserted or the MPEN bit is set. $\overline{\text{WUMI}}$ output will be asserted when the Am79C973/ Am79C975 controller is in the Magic Packet mode. Magic Packet mode can be disabled at any time by asserting PG or clearing MPEN bit.

When the Am79C973/Am79C975 controller detects a Magic Packet frame, it sets the MPMAT bit (CSR116, bit 5), the MPINT bit (CSR5, bit 4), and the PME_STATUS bit (PMCSR, bit 15). The setting of the MPMAT bit will also cause the RWU pin to be asserted and if the PME_EN or the PME_EN_OVR bits are set, then the $\overline{\text{PME}}$ will be asserted as well. If IENA (CSR0, bit 6) and MPINTE (CSR5, bit 3) are set to 1, $\overline{\text{INTA}}$ will be asserted. Any one of the four LED pins can be programmed to indicate that a Magic Packet frame has been received. MPSE (BCR4-7, bit 9) must be set to 1 to enable that function.

Note: *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 Am79C973/Am79C975 controller will discard the frame internally, but will not resume normal transmit and receive operations until PG is asserted or MPEN is cleared. Once both of these events has occurred, indicating that the system has detected the Magic Packet and is awake, the controller will continue polling receive and transmit descriptor rings where it left off. It is not necessary to re-initialize the device. If the part is re-initialized, then the descriptor locations will be reset and the Am79C973/Am79C975 controller will not start where it left off.

If magic packet mode is disabled by the assertion of PG, then in order to immediately re-enable Magic Packet mode, the PG pin must remain deasserted for at least 200 ns before it is reasserted. If Magic Packet mode is disabled by clearing MPEN bit, 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 while the device is operating in Magic Packet mode. Either of the $\overline{\text{INTA}}$, the LED pins, RWU or the $\overline{\text{PME}}$ signal 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 will do so after enabling the Magic Packet mode.

CAUTION: *To prevent unwanted interrupts from other active parts of the Am79C973/Am79C975 controller, care must be taken to mask all likely interruptible events during Magic Packet mode. An example would be the interrupts from the Media Independent Interface, which could occur while the device is 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. The following paragraphs summarize the IEEE 1149.1-compatible test functions implemented in the Am79C973/Am79C975 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.

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 $\overline{\text{TRST}}$ 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 16 for a summary of supported instructions.

Table 16. 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 17.

Table 17. 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 18).

Table 18. Device ID Register

Bits 31-28	Version
Bits 27-12	Part Number (0010 0110 0010 0101)
Bits 11-1	Manufacturer ID. The 11 bit manufacturer ID cod for AMD is 00000000001 in accordance with JEDEC publication 106-A.
Bit 0	Always a logic 1

Note: *The content of the Device ID register is the same as the content of CSR88.*

Reset

There are four different types of RESET operations that may be performed on the Am79C973/Am79C975 device, H_RESET, S_RESET, STOP, and POR. The following is a description of each type of RESET operation.

H_RESET

Hardware Reset (H_RESET) is an Am79C973/Am79C975 reset operation that has been created by the proper assertion of the $\overline{RST}$ pin of the Am79C973/Am79C975 device while the PG pin is HIGH. 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 most of the 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 Am79C973/Am79C975 controller will attempt to read the EEPROM device through the EEPROM interface.

H_RESET will clear DWIO (BCR18, bit 7) and the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will not attempt to read the EEPROM device. After S_RESET, the host must perform a full re-initialization of the Am79C973/Am79C975 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 Am79C973/Am79C975 controller will not attempt to read the EEPROM device.

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 Am79C973/Am79C975 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.

Power on Reset

Power on Reset (POR) is generated when the Am79C973/Am79C975 controller is powered up. POR generates a hardware reset (H_RESET). In addition, it clears some bits that H_RESET does not affect.

Software Access

PCI Configuration Registers

The Am79C973/Am79C975 controller implements the 256-byte configuration space as defined by the PCI specification revision 2.1. The 64-byte header includes all registers required to identify the Am79C973/Am79C975 controller and its function. Additionally, PCI Power Management Interface registers are implemented at location 40h - 47h. The layout of the Am79C973/Am79C975 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
Cardbus CIS Pointer				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address				30h
Reserved			CAP-PTR	34h
Reserved				38h
MAX_LAT	MIN_GNT	Interrupt Pin	Interrupt Line	3Ch
PMC		NXT_ITM_PTR	CAP_ID	40h
DATA_REG	PMCSR_BSE	PMCSR		44H
Reserved				. .
Reserved				FCh

I/O Resources

The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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. Additionally, the first six bytes of the EEPROM will be loaded into CSR12 to CSR14. 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 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 Am79C973/Am79C975 controller. The internal S_RESET pulse that is generated by this access is different from both the assertion of the hardware $\overline{\text{RST}}$ pin (H_RESET) and from the assertion of the software STOP bit. Specifically, S_RESET is the equivalent of the assertion of the $\overline{\text{RST}}$ pin (H_RESET) except that S_RESET has no effect on the BCR or PCI Configuration space locations.

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 Am79C973/Am79C975 controller does not have a similar requirement. The write access is not required and does not have any effect.

Note: *The Am79C973/Am79C975 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* ms *to finish. The Am79C973/Am79C975 controller will terminate all slave accesses with the assertion of* $\overline{\text{DEVSEL}}$ *and* $\overline{\text{STOP}}$ *while* $\overline{\text{TRDY}}$ *is not asserted, signaling to the initiator to disconnect and retry the access at a later time.*

Word I/O Mode

After H_RESET, the Am79C973/Am79C975 controller is programmed to operate in Word I/O mode. DWIO (BCR18, bit 7) will be cleared to 0. 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 Am79C973/Am79C975 control registers. Table 21 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

The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller operates in 32-bit I/O mode.

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 can clear it to 0. The DWIO mode setting is unaffected by S_RESET or setting of the STOP bit. Table 22 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 23 will yield undefined data, a write operation may cause unexpected reprogramming of the Am79C973/Am79C975 control registers.

Table 21. Legal I/O Accesses in Word I/O Mode (DWIO = 0)

AD[4:0]	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 22. 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 23. Legal I/O Accesses in Double Word I/O Mode (DWIO =1)

AD[4:0]	BE[3:0]	Type	Comment
0XX00	0000	RD	DWord read of APROM locations 3h (MSB) to 0h (LSB), 7h to 4h, Bh to 8h or Fh to Ch
10000	0000	RD	DWord read of RDP
10100	0000	RD	DWord read of RAP
11000	0000	RD	DWord read of Reset Register
0XX00	0000	WR	DWord write to APROM locations 3h (MSB) to 0h (LSB), 7h to 4h, Bh to 8h or Fh to Ch
10000	0000	WR	DWord write to RDP
10100	0000	WR	DWord write to RAP
11000	0000	WR	DWord write to Reset Register

USER ACCESSIBLE REGISTERS

The Am79C973/Am79C975 controller has four types of user registers: the PCI configuration registers, the Control and Status registers (CSR), the Bus Control registers (BCR), and the PHY Management registers (ANR).

The Am79C973/Am79C975 controller implements all PCnet-ISA (Am79C960) registers, all C-LANCE (Am79C90) registers, plus a number of additional registers. The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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
- OnNow register

■ Setup Registers

These registers are intended to be initialized by the device driver to program the operation of various Am79C973/Am79C975 controller features.

The following is a list of the registers that would typically need to be programmed once during the setup of the Am79C973/Am79C975 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
CSR116^	OnNow Miscellaneous
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^	Internal PHY Control and Status
BCR33^	Internal PHY Address
BCR35^	PCI Vendor ID
BCR36	PCI Power Management Capabilities (PMC) Alias Register
BCR37	PCI DATA Register Zero (DATA0) Alias Register
BCR38	PCI DATA Register One (DATA1) Alias Register
BCR39	PCI DATA Register Two (DATA2) Alias Register
BCR40	PCI DATA Register Three (DATA3) Alias Register
BCR41	PCI DATA Register Four (DATA4) Alias Register
BCR42	PCI DATA Register Five (DATA5) Alias Register
BCR43	PCI DATA Register Six (DATA6) Alias Register
BCR44	PCI DATA Register Seven (DATA7) Alias Register
BCR45	OnNow Pattern Matching Register 1
BCR46	OnNow Pattern Matching Register 2
BCR47	OnNow Pattern Matching Register 3

■ **Running Registers**

These registers are intended to be used by the device driver software after the Am79C973/Am79C975 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 Am79C973/Am79C975 controller within a system. Each of these registers contains control bits, or status bits, or both.

RAP	Register Address Port
CSR0	Am79C973/Am79C975 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	Receive Collision Count
BCR32	Internal PHY Control and Status
BCR33	Internal PHY Address
BCR34	Internal PHY 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 Am79C973/Am79C975 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.

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

PCI Device ID Register

Offset 02h

The PCI Device ID register is a 16-bit register that uniquely identifies the Am79C973/Am79C975 control-

ler within AMD's product line. The Am79C973/ Am79C975 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) and PCnet-*FAST* (Am79C971) devices.

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

PCI Command Register

Offset 04h

The PCI Command register is a 16-bit register used to control the gross functionality of the Am79C973/ Am79C975 controller. It controls the Am79C973/ Am79C975 controller's ability to generate and respond to PCI bus cycles. To logically disconnect the Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 controller detects a parity error, it only sets the Detected Parity Error bit in the PCI Status register. When PERREN is 1, the Am79C973/Am79C975 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 affected 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 Am79C973/Am79C975 controller only generates Memory Write cycles.
3	SCYCEN	Special Cycle Enable. Read as zero; write operations have no effect. The Am79C973/Am79C975 controller ignores all Special Cycle operations.
2	BMEN	Bus Master Enable. Setting BMEN enables the Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller detects a parity error. The Am79C973/Am79C975 controller samples the AD[31:0], C/$\overline{BE}$[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 Am79C973/Am79C975 controller when data is transferred ($\overline{TRDY}$ and $\overline{IRDY}$ 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 Am79C973/Am79C975 controller sets the PERR bit if the target reports a data parity error by asserting the $\overline{PERR}$ 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller terminates a master cycle with a master abort sequence.

RMABORT is set by the Am79C973/Am79C975 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 Am79C973/Am79C975 master cycle with a target abort sequence.

RTABORT is set by the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will assert $\overline{\text{DEVSEL}}$ two clock periods after $\overline{\text{FRAME}}$ is asserted.

DEVSEL is read only.

8 DATAPERR Data Parity Error Detected. DATAPERR is set when the Am79C973/Am79C975 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 Am79C973/Am79C975 controller checks for parity error by sampling the AD[31:0] and C/$\overline{\text{BE}}$[3:0] and the PAR lines. During the data phase of all memory write commands, the Am79C973/Am79C975 controller checks the $\overline{\text{PERR}}$ input to detect whether the target has reported a parity error.

DATAPERR is set by the Am79C973/Am79C975 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 Am79C973/Am79C975 controller is capable of accepting fast back-to-back transactions with the first transaction addressing a different target.

6-5 RES Reserved locations. Read as zero; write operations have no effect.

4 NEW_CAP New Capabilities. This bit indicates whether this function implements a list of extended capabilities such as PCI power management. When set, this bit indicates the presence of New Capabilities. A value of 0 means that this function does not implement New Capabilities.

Read as one; write operations have no effect. The Am79C973/Am79C975 controller supports the Linked Additional Capabilities List.

3-0 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 Am79C973/Am79C975 controller revision number. The value of this register is 4Xh with the lower four bits being silicon-revision dependent.

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

PCI Programming Interface Register

Offset 09h

The PCI Programming Interface register is an 8-bit register that identifies the programming interface of Am79C973/Am79C975 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 Am79C973/ Am79C975 controller. The value of this register is 00h which identifies the Am79C973/Am79C975 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 Am79C973/ Am79C975 controller. The value of this register is 02h which classifies the Am79C973/Am79C975 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 Am79C973/ Am79C975 controller will control the bus once it starts its bus mastership period. The time is measured in clock cycles. Every time the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 I/O resources in all of I/O space. IOBASE must be written with a valid address before the Am79C973/ Am79C975 controller slave I/O mode is turned on by setting the IOEN bit (PCI Command register, bit 0). When the Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 I/O resources in all of memory space. MEMBASE must be written with a valid address before the Am79C973/Am79C975 controller slave memory mapped I/O mode is turned on by setting the MEMEN bit (PCI Command register, bit 1). When the Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller is used in. Subsystem Vendor IDs can be obtained from the PCI SIG. A value of 0 (the default) indicates that the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Capabilities Pointer Register

Offset 34h

Bit	Name	Description
7-0	CAP_PTR	The PCI Capabilities pointer Register is an 8-bit register that points to a linked list of capabilities implemented on this device. This register has a default value of 40h. The PCI Capabilities register is located at offset 34h in the PCI Configuration Space. It is read only.

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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller. The PCI Interrupt Line register is not modified by the Am79C973/Am79C975 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 by 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 Am79C973/Am79C975 controller is using. The value for the Am79C973/Am79C975 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 Am79C973/Am79C975 device needs to keep up with the network activity. The length of the burst period is calculated assuming a clock rate of 33 MHz. The register value specifies the time in units of 1/4 ms. The PCI MIN_GNT register is an alias of BCR22, bits 7-0. It is recommended that the BCR22 be programmed to a value of 1818h.

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 Am79C973/Am79C975 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 a 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

PCI Capability Identifier Register

Offset 40h

Bit	Name	Description
7-0	CAP_ID	This register, when set to 1, identifies the linked list item as being the PCI Power Management registers. This register has a default value of 1h. The PCI Capabilities Identifier register is located at offset 40h in the PCI Configuration Space. It is read only.

PCI Next Item Pointer Register

Offset 41h

Bit	Name	Description
7-0	NXT_ITM_PTR	The Next Item Pointer Register points to the starting address of the next capability. The pointer at this offset is a null pointer, indicating that this is the last capability in the linked list of the capabilities. This register has a default value of 0h. The PCI Next Pointer Register is located at offset 41h in the PCI Configuration Space. It is read only.

PCI Power Management Capabilities Register (PMC)

Offset 42h

Note: *All bits of this register are loaded from EEPROM. The register is aliased to BCR36 for testing purposes.*

Bit	Name	Description
15-11	PME_SPT	PME Support. This 5-bit field indicates the power states in which the function may assert $\overline{\text{PME}}$. A value of 0b for any bit indicates that the function is not capable of asserting the $\overline{\text{PME}}$ signal while in that power state. Bit(11) XXXX1b - $\overline{\text{PME}}$ can be asserted from D0. Bit(12) XXX1Xb - $\overline{\text{PME}}$ can be asserted from D1. Bit(13) XX1XXb - $\overline{\text{PME}}$ can be asserted from D2. Bit(14) X1XXXb - $\overline{\text{PME}}$ can be asserted from D3hot. Bit(15) 1XXXXb - $\overline{\text{PME}}$ can be asserted from D3cold. Read only. Bit 15 of the PMC register indicates that the controller is capable of generating $\overline{\text{PME}}$ from the D3 cold state. This capability depends on the presence of auxiliary power, as indicated by the AUXDET input. The capability can be disabled by loading a zero into bit 15 of BCR36 from the EEPROM. (This register is aliased to the PMC register.)
10	D2_SPT	D2 Support. If this bit is a 1, this function supports the D2 Power Management State. Read only.
9	D1_SPT	D1 Support. If this bit is a 1, this function supports the D1 Power Management State. Read only.
8-6	RES	Reserved locations. Written as zeros and read as undefined.
5	DSI	Device Specific Initialization. When this bit is 1, it indicates that special initialization of the function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. Read only.
4	RES	Reserved locations. Written as zeros and read as undefined.
3	PME_CLK	PME Clock. When this bit is a 1, it indicates that the function relies on the presence of the PCI clock for $\overline{\text{PME}}$ operation. When this bit is a 0 it indicates that no PCI clock is required for the function to generate $\overline{\text{PME}}$. Functions that do not support $\overline{\text{PME}}$ generation in any state must return 0 for this field. Read only.
2-0	PMIS_VER	Power Management Interface Specification Version. A value of 001b indicates that this function complies with the revision 1.1 of the PCI Power Management Interface Specification.

PCI Power Management Control/Status Register (PMCSR)

Offset 44h

Bit	Name	Description
15	PME_STATUS	PME Status. This bit is set when the function would normally assert the $\overline{\text{PME}}$ signal independent of the state of the PME_EN bit. Writing a 1 to this bit will clear it and cause the function to stop asserting a $\overline{\text{PME}}$ (if enabled). Writing a 0 has no effect. If the function supports $\overline{\text{PME}}$ from D3cold then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded.

Read/write accessible always. Sticky bit. This bit is reset by POR. H_RESET, S_RESET, or setting the STOP bit has no effect.

14-13 DATA_SCALE

Data Scale. This two bit read-only field indicates the scaling factor to be used when interpreting the value of the Data register. The value and meaning of this field will vary depending on the DATA_SCALE field.

Read only.

12-9 DATA_SEL Data Select. This optional four-bit field is used to select which data is reported through the Data register and DATA_SCALE field.

Read/write accessible always. Sticky bit. This bit is reset by POR. H_RESET, S_RESET, or setting the STOP bit has no effect.

8 PME_EN PME Enable. When a 1, PME_EN enables the function to assert $\overline{PME}$. When a 0, $\overline{PME}$ assertion is disabled.

This bit defaults to “0” if the function does not support $\overline{PME}$ generation from D3cold.

If the function supports $\overline{PME}$ from D3cold, then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded.

Read/write accessible always. Sticky bit. This bit is reset by POR. H_RESET, S_RESET, or setting the STOP bit has no effect.

7-2 RES Reserved locations. Read only.

1-0 PWR_STATE Power State. This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. The definition of the field values is given below.

00b - D0.

01b - D1.

10b - D2.

11b - D3.

These bits can be written and read, but their contents have no effect on the operation of the device.

Read/write accessible always.

PCI PMCSR Bridge Support Extensions Register

Offset 46h

Bit	Name	Description
7-0	PMCSR_BSE	The PCI PMCSR Bridge Support Extensions Register is an 8-bit register. PMCSR Bridge Support Extensions are not supported. This register has a default value of 00h. The PCI PMCSR Bridge Support Extensions register is located at offset 46h in the PCI Configuration Space. It is read only.

PCI Data Register

Offset 47h

Note: *All bits of this register are loaded from EEPROM. The register is aliased to lower bytes of the BCR37-44 for testing purposes.*

Bit	Name	Description
7-0	DATA_REG	The PCI Data Register is an 8-bit register. Refer to the “PCI Bus Power Management Interface Specification” version 1.1 for a more detailed description of this register. The PCI DATA register is located at offset 47h 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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: Am79C973/Am79C975 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 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	RES	Reserved locations. Read/Write accessible always. Read returns zero.
13	CERR	Collision Error is set by the Am79C973/Am79C975 controller when the device operates in half-duplex mode and the collision inputs (10 Mbps) failed to activate within 20 network bit times after the chip terminated transmission (SQE Test). This feature is a 10BASE-T PHY test feature. CERR reporting is disabled when the Am79C973/Am79C975 controller operates in full-duplex mode. When the MII port is selected, CERR is only reported when the external PHY is operating as a half-duplex 10BASE-T PHY. 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 ms 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller after the initialization sequence has completed. When IDON is set, the Am79C973/Am79C975 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: EXDINT, IDON, MERR, MISS, MFCO, RCVCCO, RINT, SINT, TINT, TXSTRT, UINT, STINT, MREINT, MCCINT, 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, $\overline{\text{INTA}}$ will be active independent of the state of INEA.

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

6 IENA — Interrupt Enable allows $\overline{\text{INTA}}$ to be active if the Interrupt Flag is set. If IENA = 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.

This bit will reset if the DXSUFLO bit (CSR3, bit 6) is reset and there is an underflow condition encountered.

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 bus master accesses, while the 32-bit hardware for which the Am79C973/Am79C975 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 Am79C973/Am79C975 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.
15-13	RES	Reserved locations. 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller to read the STP bit of receive descriptors. The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will scan through the next descriptor entries to locate the next STP bit that is set to a 1. The Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller ownership of the descriptor but also indicate STP = 0, then the Am79C973/Am79C975 controller will reset the OWN bit to 0 in these entries. If a scanned entry indicates host ownership with STP = 0, then the Am79C973/ Am79C975 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 Am79C973/Am79C975 controller, then the Am79C973/ Am79C975 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 Am79C973/ Am79C975 controller, then the Am79C973/Am79C975 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 E 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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.

1-0 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	RES	Reserved location. It is OK for legacy software to write a 1 to this location. This bit must be set back to 0 before setting INIT or STRT bits. Read/Write accessible always. This bit 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	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 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 Am79C973/ Am79C975 controller when the Missed Frame Counter (CSR112 and CSR113) 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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.

1-0 RES Reserved locations. Written as zeros and read as undefined.

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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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.
9-8	RES	Reserved locations. Written as zeros and read as undefined.
7	EXDINT	Excessive Deferral Interrupt is set by the Am79C973/ Am79C975 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 Am79C973/Am79C975 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. This bit is OR'ed with EMPPLBA bit (CSR116, bit 6).

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 Am79C973/Am79C975 controller when the device is in the Magic Packet mode and the Am79C973/Am79C975 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 Am79C973/Am79C975 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 The Am79C973/Am79C975 controller will enter the Magic Packet mode when MPMODE is set to 1 and either PG 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 Am79C973/Am79C975 controller to start requesting entrance into suspend mode. The host must poll SPND until it reads back 1 to determine that the Am79C973/Am79C975 controller has entered the suspend mode. Setting SPND to 0 will get the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 controller initialization. This field is written during the Am79C973/Am79C975 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 Am79C973/Am79C975 controller initialization. This field is written during the Am79C973/ Am79C975 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 Am79C973/Am79C975 controller performs a fast suspend whenever the SPND bit is set. When a fast suspend is requested, the Am79C973/Am79C975 controller performs a quick entry into the suspend mode. At the time the SPND bit is set, the Am79C973/Am79C975 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 Am79C973/ Am79C975 controller may enter the suspend mode with transmit and/or receive packets still in the FIFOs or the SRAM. When FASTSPNDE is 0 and the SPND bit is set, the Am79C973/ Am79C975 controller may take longer before entering the suspend mode. At the time the SPND bit is set, the Am79C973/ Am79C975 controller will complete the DMA process of a transmit packet if it had already begun

and the Am79C973/Am79C975 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 SRAM (if one is enabled) will be transmitted and all receive packets stored in the receive FIFOs, and the receive buffer area in the SRAM (if one is enabled) will be transferred into system memory. Since the FIFO and SRAM contents are flushed, it may take much longer before the Am79C973/Am79C975 controller enters the suspend mode. The amount of time that it takes depends on many factors including the size of the 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 PHY Management Read Error Interrupt. The PHY Read Error interrupt is set by the Am79C973/ Am79C975 controller to indicate that the currently read register from the 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 internal PHY.

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 PHY 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 PHY Management Auto-Poll Interrupt. The PHY Auto-Poll interrupt is set by the Am79C973/ Am79C975 controller to indicate that the currently read status does not match the stored previous status indicating a change in state for the internal PHY. A change in the Auto-Poll Access Method (BCR32, Bit 11) 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 PHY 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 PHY Management Command Complete Interrupt. The PHY Management Command Complete Interrupt is set by the Am79C973/Am79C975 controller when a read or write operation to the internal PHY 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 PHY 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 internal PHY Data Port (BCR34) only. Internal PHY Management Commands will not generate an interrupt. For instance Auto-Poll state machine generated 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 PHY Management Command Complete Internal Interrupt. The PHY Management Command Complete Interrupt is set by the Am79C973/Am79C975 controller when a read or write operation on the internal PHY management port is complete from an internal operation. Examples of internal operations are Auto-Poll or PHY Management Port generated 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 PHY 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 management frames. For instance, when MCCIINTE is set to 1 and the Auto-Poll state machine generates a management frame, the MCCIINT will set the INTR bit upon completion of the 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 PHY Detect Transition Interrupt. The PHY Detect Transition Interrupt is set by the Am79C973/Am79C975 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 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

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	PADR[15:0]	Physical Address Register, PADR[15:0]. The contents of this register are loaded from EEPROM after H_RESET or by an EEPROM read command (PRGAD, BCR19, bit 14). If the EEPROM is not present, the contents of this register are undefined. This register can also be 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

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	PADR[31:16]	Physical Address Register, PADR[31:16]. The contents of this register are loaded from EEPROM after H_RESET or by an EEPROM read command (PRGAD, BCR19, bit 14). If the EEPROM is not present, the contents of this register are undefined. This register can also be 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

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	PADR[47:32]	Physical Address Register, PADR[47:32].The contents of this register are loaded from EEPROM after H_RESET or by an EEPROM read command (PRGAD, BCR19, bit 14). If the EEPROM is not present, the contents of this register are undefined. This register can also be 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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.
12-9	RES	Reserved locations. Written as zeros and read as undefined.
8-7	PORTSEL[1:0]	Port Select bits allow for software controlled selection of the network medium. The only legal values for this field is 11. 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.
5	DRTY	Disable Retry. When DRTY is set to 1, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 con-

troller 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 and ENP bits are set in TMD1.

When APAD_XMT bit (CSR4, bit11) 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. 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 Am79C973/Am79C975 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 MIIILP, various loopback modes are defined as follows in Table 24.

Table 24. Loopback Configuration

LOOP	INTL	MIIILP	Function
0	0	0	Normal Operation
0	0	1	Internal Loop
1	0	0	External Loop

Refer to *Loop Back Operation* section for more details.

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Uper

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 Am79C973/Am79C975 controller microcode and is used to trigger the transmit descriptor ring polling operation of the Am79C973/ Am79C975 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 Am79C973/Am79C975 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 0000h. This corresponds to a polling interval of 65,536 clock periods (1.966 ms when CLK = 33 MHz). The TXPOLLINT value of 0000h is created during the microcode initialization routine and, therefore, might not be seen when reading CSR47 after H_RESET or S_RESET. If the user desires to program a value for POLLINT other than the default, 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 Am79C973/Am79C975 controller microcode and is used to trigger the receive descriptor ring polling operation of the Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/

Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller utilizes 32-bit software structures for the initialization block and the transmit and receive descriptor entries. When cleared, this bit indicates that the Am79C973/ Am79C975 controller utilizes 16-bit software structures for the initialization block and the transmit and receive descriptor entries. In this mode, the Am79C973/ Am79C975 controller is backwards compatible with the Am7990 LANCE and Am79C960 PCnet-ISA controllers.

The value of SSIZE32 is determined by the Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 controller bus master accesses.

If SSIZE32 is set, then the software structures that are common to the Am79C973/Am79C975 controller and the host system will supply a full 32 bits for each address pointer that is needed by the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller CSR bits and BCR bits and all descriptor, buffer, and initialization block entries not cited in Table 25 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. The Am79C973/ Am79C975 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 25. 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 or burst access
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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975Am79C973/ Am79C975 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 Am79C973/ Am79C975 controller is in full-duplex mode, in order for 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 enabled), 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. When operating with the SRAM, the Bus Receive FIFO, and the MAC Receive FIFO operate independently on the bus side and MAC side of the 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 26.

Table 26. 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 Am79C973/ Am79C975 controller can overwrite the beginning of the frame as soon as the data is transmit-

ted, because no collision handling is required in these modes.

Note that when the SRAM is being used, if the NOUFLO bit (CSR80, bit 14) is set to 1, there is the additional restriction that the complete transmit frame must be DMA'd into the Am79C973/ Am79C975 controller and reside within a combination of the Bus Transmit FIFO, the SRAM, and the MAC Transmit FIFO.

When the 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 Am79C973/ Am79C975 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 27.

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.

Table 27. Transmit Start Point Programming

XMTSP[1:0]	SRAM_SIZE	Bytes Written
00	0	20
01	0	64
10	0	128
11	0	220 max
00	>0	36
01	>0	64
10	>0	128
11	>0	Full Packet when NOUFLO bit is set

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 enabled), 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.

When operating with the SRAM, the Bus Transmit FIFO and the MAC Transmit FIFO operate independently on the bus side and MAC side of the 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 28.

Table 28. 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973/ Am79C975 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 Am79C973 controller is 0010 0110 0010 0101 (2625h) and the code for the Am79C975 is 0010 0110 0010 0111 (2627h).

This register is exactly the same as the Device ID register in the JTAG description. However, this part number is different from 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. 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.

Device	CSR88
Am79C973	5003h
Am79C975	7003h

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 bus clock periods divided by two, (i.e., the value in this register is given in 0.1 ms increments.) For example, the value 0600h (1536 decimal) will cause a MERR to be indicated after 153.6 ms 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.

CSR116: OnNow Power Mode Register

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

Bit	Name	Description
31-14	RES	Reserved locations. Written as zeros and read as undefined.
13	LCMODE_D3C.	This bit is a read/write from the PCI bus and is reset only at power-on. This bit is not written from the EEPROM. Power management software can set this bit before going to D3cold and even if there is a reset and the EEPROM loads because of an incorrect PG signal, the control bit will not be changed. This bit is OR'ed with LCMODE (CSR116 bit 8) for OnNow link chnage, but not for hardware link change
12	MPPEN_D3C	This bit is read/write from the the PCI bus and is reset only at power-on. This bit is not written from the EEPROM. Power management software can set this bit before going to D3cold and even if there is a reset and the EEPROM loads because of an incorrect PG signal, the control bit will not be changed. This bit is OR'ed with MPPEN (CSR116 bit 4) for both hardware magic packet or OnNow magic packet.
11	PMAT_MODE_D3C	This bit is read/write from the the PCI bus and is reset only at power-on. This bit is not written from the EEPROM. Power management software can set this bit before going to D3cold and even if there is a reset and the EEPROM loads because of an incorrect PG signal, the control bit will not be changed. This bit is OR'ed with PMAT_MODE (BCR45 bit 7) for OnNow magic packet.

10 PME_EN_OVR — PME_EN Overwrite. When this bit is set and the MPMAT or LCDET bit is set, the $\overline{\text{PME}}$ pin will always be asserted regardless of the state of PME_EN bit.

Read/Write accessible only when either the STOP bit or the SPND bit is set. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

9 LCDET — Link Change Detected. This bit is set when the MII auto-polling logic detects a change in link status and the LCMODE bit is set.

LCDET is cleared when power is initially applied (POR).

Read/Write accessible always.

8 LCMODE — Link Change Wake-up Mode. When this bit is set to 1, the LCDET bit gets set when the MII auto polling logic detects a Link Change.

Read/Write accessible only when either the STOP bit or the SPND bit is set. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

7 PMAT — Pattern Matched. This bit is set when the PMMODE bit is set and an OnNow pattern match occurs.

PMAT is cleared when power is initially applied (POR).

Read accessible always.

6 EMPPLBA — Magic Packet Physical Logical Broadcast Accept. If both EMPPLBA and MPPLBA (CSR5, bit 5) are at their default value of 0, the Am79C973/Am79C975 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 either EMPPLBA or MPPLBA is set to 1, the destination address of the Magic Packet frame can be unicast, multicast, or broadcast. Note that the setting of EMPPLBA and 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. EMPPLBA is set to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.

5 MPMAT — Magic Packet Match. This bit is set when PCnet-FAST+ detects a Magic Packet while it is in the Magic Packet mode.

MPMAT is cleared when power is initially applied (POR).

Read/Write accessible always.

4 MPPEN — Magic Packet Pin Enable. When this bit is set, the device enters the Magic Packet mode when the PG input goes LOW or MPEN bit (CSR5, bit 2) gets set to 1. This bit is OR'ed with MPEN bit (CSR5, bit 2).

Read/Write is accessible only when either the STOP bit or the SPND bit is set. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

3 RWU_DRIVER — RWU Driver Type. If this bit is set to 1, RWU is a totem pole driver; otherwise RWU is an open drain output.

Read/Write is accessible only when either the STOP bit or the SPND bit is set. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

2 RWU_GATE — RWU Gate Control. If this bit is set, RWU is forced to the high Impedance State when PG is LOW, regardless of the state of the MPMAT and LCDET bits.

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

bit is set. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

1 RWU_POL — RWU Pin Polarity. If RWU_POL is set to 1, the RWU pin is normally HIGH and asserts LOW; otherwise RWU is normally LOW and asserts HIGH.

Read/Write accessible only when either the STOP bit or the SPND bit is set. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

0 RST_POL — PHY_RST Pin Polarity. If the PHY_POL is set to 1, the PHY_RST pin is active LOW; otherwise PHY_RST is active HIGH.

Read/Write accessible only when either the STOP bit or the SPND bit is set. Cleared by H_RESET and is not affected by S_RESET or 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller for aggressiveness on a network. By changing the default value of 96 bit times (60h) the user can adjust the fairness or aggressiveness of the Am79C973/Am79C975 MAC on the network. By programming a lower number of bit times other then the ISO/IEC 8802-3 standard requires, the Am79C973/Am79C975 MAC will become more aggressive on the network. This aggressive nature will give rise to the Am79C973/Am79C975 controller possibly “capturing the network” at times by forcing other less ag-

gressive nodes to defer. By programming a larger number of bit times, the Am79C973/ Am79C975 MAC will become less aggressive on the network and may defer more often than normal. The performance of the Am79C973/Am79C975 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 Am79C973/Am79C975 controller can interrupt normal network behavior.

Read/Write is accessible only when either the STOP bit or the SPND bit is set. 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 Am79C973/Am79C975 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 as follows:

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/Write is accessible only when either the STOP bit or the SPND bit is set. 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 Am79C973/Am79C975 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 29.

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 or S_RESET.

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 Am79C973/Am79C975 controller.

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

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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.

Table 29. BCR Registers (Am79C973)

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	PHY Control and Status	Yes	Yes
33	MIIADDR	0000h	PHY Address	Yes	Yes
34	MIIMDR	N/A	PHY Management Data	Yes	No
35	PCIVID	1022h	PCI Vendor ID	No	Yes
36	PMC_A	C811h	PCI Power Management Capabilities (PMC) Alias Register	No	Yes
37	DATA0	0000h	PCI DATA Register Zero Alias Register	No	Yes
38	DATA1	0000h	PCI DATA Register One Alias Register	No	Yes
39	DATA2	0000h	PCI DATA Register Two Alias Register	No	Yes
40	DATA3	0000h	PCI DATA Register Three Alias Register	No	Yes
41	DATA4	0000h	PCI DATA Register Four Alias Register	No	Yes
42	DATA5	0000h	PCI DATA Register Five Alias Register	No	Yes
43	DATA6	0000h	PCI DATA Register Six Alias Register	No	Yes
44	DATA7	0000h	PCI DATA Register Seven Alias Register	No	Yes
45	PMR1	N/A	Pattern Matching Register 1	Yes	No
46	PMR2	N/A	Pattern Matching Register 2	Yes	No
47	PMR3	N/A	Pattern Matching Register 3	Yes	No
48	Reserved	0000h	Reserved (for Am79C975)	Yes*	Yes*
49	Reserved	0000h	Reserved (for Am79C975)	Yes*	Yes*
50	Reserved	0000h	Reserved (for Am79C975)	Yes*	Yes*
51	Reserved	0000h	Reserved (for Am79C975)	Yes*	Yes*
52	Reserved	0000h	Reserved (for Am79C975)	Yes*	Yes*

Table 29. BCR Registers (Am79C973)

53	Reserved	0000h	Reserved (for Am79C975)	Yes*	Yes*
54	Reserved	0000h	Reserved (for Am79C975)	Yes*	Yes*

Note: **Program only as '0' value.*

Table 30. BCR Registers (Am79C975)

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	PHY Control and Status	Yes	Yes
33	MIIADDR	0000h	PHY Address	Yes	Yes
34	MIIMDR	N/A	PHY Management Data	Yes	No
35	PCIVID	1022h	PCI Vendor ID	No	Yes
36	PMC_A	C811h	PCI Power Management Capabilities (PMC) Alias Register	No	Yes
37	DATA0	0000h	PCI DATA Register Zero Alias Register	No	Yes
38	DATA1	0000h	PCI DATA Register One Alias Register	No	Yes
39	DATA2	0000h	PCI DATA Register Two Alias Register	No	Yes
40	DATA3	0000h	PCI DATA Register Three Alias Register	No	Yes
41	DATA4	0000h	PCI DATA Register Four Alias Register	No	Yes
42	DATA5	0000h	PCI DATA Register Five Alias Register	No	Yes
43	DATA6	0000h	PCI DATA Register Six Alias Register	No	Yes
44	DATA7	0000h	PCI DATA Register Seven Alias Register	No	Yes
45	PMR1	N/A	Pattern Matching Register 1	Yes	No
46	PMR2	N/A	Pattern Matching Register 2	Yes	No
47	PMR3	N/A	Pattern Matching Register 3	Yes	No
48	N_IP_ADR[15:0]	0000h	Node IP Address [15:0]	Yes	Yes
49	N_IP_ADR[31:16]	0000h	Node IP Address [31:16]	Yes	Yes
50	M_IEEE_ADR[15:0]	0000h	Management IEEE Address [15:0]	Yes	Yes
51	M_IEEE_ADR[31:16]	0000h	Management IEEE Address [31:16]	Yes	Yes
52	M_IEEE_ADR[47:32]	0000h	Management IEEE Address [47:32]	Yes	Yes

Table 30. BCR Registers (Am79C975)

53	M_IP_ADR[15:0]	0000h	Management IP Address [15:0]	Yes	Yes
54	M_IPADR[31:16]	0000h	Management IP Address [31:16]	Yes	Yes

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	SMIUEN	(For Am79C975 only) SMIUEN is used to enable/disable the Serial Management Interface Unit in the Am79C975 controller. If SMIUEN is set to 0 (default), the SMIU is disable. If SMIUEN is set to 1, the SMIU is enabled. Read/Write accessible always. SMIUEN is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
14	DISSCR_SFEX	This bit is used to disable the scrambler/descrambler when the device is used in PECL mode. This bit defaults to 0, which enables the scrambler/descrambler for MLT3 applications. When DISSCR_SFEX is set to 1, the scrambler will be disabled for fiber applications. Read/Write accessible always. This bit is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
13	PHYSELEN	This bit enables writes to BCR18[4:3] for software selection of various operation and test modes. When PHYSELEN is set to 0 (default), the two bits can only be written from the EEPROM. When PHYSELEN is set to 1, writes to BCR18[4:3] are enabled. Read/Write accessible always. PHYSELEN is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
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	RESET_SFEX	This bit is used to reset the internal PHY. When RESET_SFEX is set to 1, the internal PHY will stay reset until RESET_SFEX is cleared to 0. Read/Write accessible always. RESET_SFEX is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
10	I2C_M3	(Am79C975 only). This bit is used to set the operating frequency of the SMIU core. It represents the value in the D6 bit position (see Appendix B on SMIU Bus Frequency. Read/Write accessible always. I2C_M3 is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
9	I2C_M2	(Am79C975 only). This bit is used to set the operating frequency of the SMIU core. It represents the

value in the D5 bit position (see Appendix B on SMIU Bus Frequency.

Read/Write accessible always. I2C_M2 is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

8 APROMWE Address PROM Write Enable. The Am79C973/Am79C975 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 Am79C973/Am79C975 controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is tri-stated by the Am79C973/Am79C975 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 Am79C973/Am79C975 controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is driven to a low level by the Am79C973/Am79C975 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 Am79C973/Am79C975 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 I2C_M1 (Am79C975 only). This bit is used to set the operating frequency of the SMIU core. It represents the value in the D4 bit position (see Appendix B on SMIU Bus Frequency.

Read/Write accessible always. I2C_M1 is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

5 I2C_M0 (Am79C975 only). This bit is used to set the operating frequency of the SMIU core. It represents the value in the D3 bit position (see Appendix B on SMIU Bus Frequency.

Read/Write accessible always. I2C_M0 is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

4 I2C_N2 (Am79C975 only). This bit is used to set the operating frequency of the SMIU core. It represents the value in the D2 bit position (see Appendix B on SMIU Bus Frequency.

Read/Write accessible always. I2C_N2 is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

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 MIIRXFRTGD, and $\overline{\text{LED2}}$ becomes MIIRXFRTGE.

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 SLEEP_SFEX

Setting this bit will reduce the power consumption of the internal PHY substantially.

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

1 I2C_N1 (Am79C975 only). This bit is used to set the operating frequency of the SMIU core. It represents the value in the D1 bit position (see Appendix B on SMIU Bus Frequency.

Read/Write accessible always. I2C_N1 is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

0 I2C_N0 (Am79C975 only). This bit is used to set the operating frequency of the SMIU core. It represents the value in the D0 bit position (see Appendix B on SMIU Bus Frequency.

Read/Write accessible always. I2C_N0 is cleared by H_RESET and is unaffected by S_RESET or by setting the STOP bit.

BCR4: LED 0 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 Am79C973/Am79C975 controller is operating at 100 Mbps mode. Read/Write accessible always. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
11-10	RES	Reserved locations. Written and read as zeros.
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 Am79C973/Am79C975 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C973/Am79C975 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. 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 in Link Pass state. 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	RES	Reserved location. Written and read as zeros.
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	RES	Reserved location. Written and read as zeros.
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. 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 Am79C973/Am79C975 controller is operating at 100 Mbps mode. Read/Write accessible always. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

11-10	RES	Reserved locations. Written and read as zeros.
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 Am79C973/ Am79C975 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C973/ Am79C975 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. 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 in Link Pass state. 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	RES	Reserved location. Written and read as zeros.
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	RES	Reserved location. Written and read as zeros.
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. 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.

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 Am79C973/Am79C975 controller is operating at 100 Mbps mode. Read/Write accessible always. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
11-10	RES	Reserved locations. Written and read as zeros.
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 Am79C973/

		Am79C975 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C973/ Am79C975 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. 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 in Link Pass state. 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	RES	Reserved location. Written and read as zeros.
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	RES	Reserved location. Written and read as zeros.
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. 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 Am79C973/Am79C975 controller is operating at 100 Mbps mode.

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

11-10 RES — Reserved locations. Written and read as zeros.

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 Am79C973/Am79C975 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C973/Am79C975 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.

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 in Link Pass state. 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	RES	Reserved location. Written and read as zeros.
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	RES	Reserved location. Written and read as zeros.
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. 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 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. This bit needs to be set if in full-duplex mode and external address rejection ($\overline{\text{EAR}}$ (BCR9, bit 2)) functionality is desired. Read/Write accessible always. FDRPAD is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
1	RES	Reserved locations. Written as zeros and read as undefined.
0	FDEN	Full-Duplex Enable. FDEN controls whether full-duplex operation is enabled. When FDEN is cleared and the Auto-Negotiation is disabled, full-duplex operation is not enabled and the Am79C973/Am79C975 controller

will always operate in the half-duplex mode. When FDEN is set, the Am79C973/Am79C975 controller will operate in full-duplex mode. ***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 Am79C973/Am79C975 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.

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 Am79C973/Am79C975 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: *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 Flash/EPROM as well as all Expansion ROM accesses to Flash/EPROM. ROMTMG, during read operations, defines the time from when the Am79C973/Am79C975 controller drives the lower 8 or 16 bits of the Expansion Bus Address bus to when the Am79C973/Am79C975 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 Am79C973/Am79C975 controller drives the lower 8 or 16 bits of the Expansion Bus Data to when the $\overline{\text{EBWE}}$ and $\overline{\text{EROMCS}}$ deassert. The register value specifies the time in number of clock cycles +1 according to Table 31.

Table 31. 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 EBDATA (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 EBDATA (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 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller. The complete packet may reside in any combination of the Bus Transmit FIFO, the 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 Am79C973/Am79C975 controller will never suffer transmit underflows, because the arbiter that controls transfers to and from the 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 Am79C973/Am79C975 controller before packet transmission begins.

The NOUFLO bit has no effect when the Am79C973/Am79C975 controller is operating in the NO-SRAM mode.

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 Am79C973/Am79C975 controller never performs more than one burst transaction within a single bus mastership period.) In this mode, the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller is programmed for DWord I/O (DWIO) mode. When cleared, this bit indicates that the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller. Specifically, the Am79C973/Am79C975 controller will set DWIO if it detects a DWord write access to offset 10h from the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 PHYSEL[1:0]

PHYSEL[1:0] bits allow for software controlled selection of different operation and test modes. The normal mode of operation is when both bits 0 and 1 are set to 0 to select the Expansion ROM/ Flash. Setting bit 0 to 1 and bit 1 to 0 allows snooping of the internal MII-compatible bus to allow External Address Detection Interface (EADI). Setting PHYSEL[1:0] = 10 enables the external MII mode. Since the internal 10/100 PHY is disabled in this mode, the Am79C973/ Am79C975 controller behaves like an Am79C972 PCnet-*FAST*+ device. Tehese controllers are not designed to support two PHYs at a time as only one PHY can be used at a given time. An external or internal PHY can be used by reconfiguring the EEPROM or the BCI18 register. See the following table for details.

PHYSEL [1:0]	Mode
00	Expansion ROM/Flash - Normal Mode
01	EADI/Internal MII Snoop Mode
10	Full MII Mode - An External PHY Required
11	Reserved

Read accessible always, these bits can only be written from the EEPROM unless a write-enable bit BCR2[13], is set. PHYSEL [1:0] is cleared by H_RESET and is not affected by S_RESET or STOP.

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 82 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 Am79C973/ Am79C975 controller will perform a read operation of 82 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 Am79C973/Am79C975 controller. Upon completion of the EEPROM read operation, the Am79C973/Am79C975 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 Fh) 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 Am79C973/Am79C975 controllers internal copy of the EEPROM contents. Write accesses to these locations may change the Am79C973/Am79C975 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 Am79C973/ Am79C975 controller and PVALID will be set, provided that an EEPROM existed on the interface pins and that the checksum for the entire 82 bytes of EEPROM was correct.

Note that when PREAD is set to a 1, then the Am79C973/ Am79C975 controller will no longer respond to any accesses directed toward it, until the PREAD operation has completed successfully. The Am79C973/ Am79C975 controller will terminate these accesses with the assertion of $\overline{\text{DEVSEL}}$ and $\overline{\text{STOP}}$ while $\overline{\text{TRDY}}$ is not asserted, signaling to the initiator to disconnect and retry the access at a later time.

If a PREAD command is given to the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and the auto-detection pin (EESK/$\overline{\text{LED1}}$/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/$\overline{\text{LED1}}$/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/$\overline{\text{LED1}}$/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 32 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 32.

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.

Table 32. 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

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.

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller utilizes 32-bit software structures for the initialization block and the transmit and receive descriptor entries. When cleared, this bit indicates that the Am79C973/Am79C975 controller utilizes 16-bit software structures for the initialization block and the transmit and receive descriptor entries. In this mode, the Am79C973/Am79C975 controller is backwards compatible with the Am7990 LANCE and Am79C960 PCnet-ISA controllers. The value of SSIZE32 is determined by the Am79C973/Am79C975 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 Am79C973/ Am79C975 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 Am79C973/Am79C975 controller bus master accesses.

If SSIZE32 is set, then the software structures that are common to the Am79C973/Am79C975 controller and the host system will supply a full 32 bits for each address pointer that is needed by the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller CSR bits and all descriptor, buffer, and initialization block entries not cited in the Table 33 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.

Table 33. 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 or burst access
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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 recommended 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 Am79C973/Am79C975 controller needs to keep up with the network activity. The length of the burst period is calculated assuming a clock rate of 33 MHz. The register value specifies the time in units of 1/4 ms. 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 Am79C973/Am79C975 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 ms, which is the time it takes to Am79C973/Am79C975 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
31-0	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 Am79C973/Am79C975 controller is used in. Subsystem Vendor IDs can be obtained form the PCI SIG. A value of 0 (the default) indicates that the Am79C973/Am79C975 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 Am79C973/Am79C975 controller is used in. The value of SID is up to the system vendor. A value of 0 (the default) indicates that the Am79C973/Am79C975 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 17FFh. An SRAM_SIZE value of all zeros specifies that no SRAM will be used 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 Am79C973/Am79C975 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.* ***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 Am79C973/Am79C975 controller will not operate correctly with less than four pages of memory per queue. See Table 34 for SRAM_BND programming details.*

Table 34. SRAM_BND Programming

SRAM Addresses	SRAM_BND [7:0]
Minimum SRAM_BND Address	04h
Maximum SRAM_BND Address	13h

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

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 Am79C973/Am79C975 controller will switch to an architecture applicable to cut-through switches. The Am79C973/Am79C975 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 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller may give erratic behavior when it is enable again. The minimum allowed number of pages is four. The Am79C973/Am79C975 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 35 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 Time Base 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 35. EBCS Values

EBCS	Expansion Bus Clock Source
000	CLK pin (PCI Clock)
001	Time Base 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 Time Base 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 Time Base Clock will not support 100 Mbit operation and should only be selected in 10 Mbit only configurations.

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 36.

Table 36. 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
31-16	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 Accesses*. 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 access 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 256 Time Base Clock periods. For instance, a value of 122 ms would be programmed with a value of 9531 (253Bh) if the Time Base Clock is running at 20 MHz. 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: PHY 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 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 MIIPD bit is set to 1, the MII port is selected. Any transition on the MIIPD bit will set the MIIPDTI bit in CSR7, bit 3. Read accessible always. MIIPD is read only. Write operations are ignored and should not be performed.
13-12	FMDC	Fast Management Data Clock (is used for manufacturing tests). When FMDC is set to 2h the MII Management Data Clock will run at 10 MHz max. 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 max. 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 max and will be fully compliant to IEEE 802.3u standards. See Table 37.

Table 37. FMDC Values

FMDC	Fast Management Data Clock
00	2.5 MHz max
01	5 MHz max
10	10 MHz max
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	Auto-Poll PHY. APEP when set to 1 the Am79C973/Am79C975 controller will poll the status register in the PHY. This feature allows the software driver or upper layers to see any changes in the status of the PHY. An interrupt when enabled is generated when the contents of the new status is different from the previous status.

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 — Auto-Poll Dwell Time. APDW determines the dwell time between PHY Management Frame accesses when Auto-Poll is turned on. See Table 38.

Table 38. 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will wait for the software driver to setup the Auto-Negotiation portions of the device. The PHY Address and Data programming in BCR33 and BCR34 is still valid. The Am79C973/Am79C975 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 — PHY Reset. When XPHYRST is set, the Am79C973/Am79C975 controller after an H_RESET or S_RESET will issue management frames that will reset the 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 — PHY Auto-Negotiation Enable. This bit will force the PHY into enabling Auto-Negotiation. When set to 0 the Am79C973/Am79C975 controller will send a 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 — PHY Full Duplex. When set, this bit will force the 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.

3 XPHYSP — PHY Speed. When set, this bit will force the 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.

2 RES — Reserved location. Written as zeros and read as undefined.

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 Am79C973/Am79C975 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	RES	Reserved location. Written as zeros and read as undefined.

BCR33: PHY Address Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	RES	Reserved locations. Written as zeros and read as undefined.
9-5	PHYAD	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 PHY address 1Fh is not valid. The PHY address of the internal PHY unit is 1Eh (30 dec.) The Network Port Manager copies the PHYAD after the Am79C973/Am79C975 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 Am79C973/Am79C975 controller. Read/Write accessible always. PHYAD is undefined after H_RESET and is unaffected by S_RESET and the STOP bit.
4-0	REGAD	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 internal MII management interface 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: PHY 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 Am79C973/Am79C975 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. 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/ Am79C975 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.

BCR36: PCI Power Management Capabilities (PMC) Alias Register

Note: *This register is an alias of the PMC register located at offset 42h of the PCI Configuration Space. Since PMC register is read only, BCR36 provides a means of programming it through the EEPROM. The contents of this register are copied into the PMC register. For the definition of the bits in this register, refer to the PMC register definition. Bits 15-0 in this register are programmable through the EEPROM. Read accessible always. Read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.*

BCR37: PCI DATA Register Zero (DATA0) Alias Register

Note: *This register is an alias of the DATA register and also of the DATA_SCALE field of the PMCSR register. Since these two are read only, BCR37 provides a means of programming them indirectly. The contents of this register are copied into the corresponding fields pointed with the DATA_SEL field set to zero. Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D0_SCALE	These bits correspond to the DATA_SCALE field of the PMCSR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. Read accessible always. D0_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7-0	DATA0	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. Read accessible always. DATA0 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit

BCR38: PCI DATA Register One (DATA1) Alias Register

Note: *This register is an alias of the DATA register and also of the DATA_SCALE field of the PMCSR register. Since these two are read only, BCR38 provides a means of programming them through the EEPROM. The contents of this register are copied into the corresponding fields pointed with the DATA_SEL field set to one. Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D1_SCALE	These bits correspond to the DATA_SCALE field of the PMCSR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field.

		Read accessible always. D1_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7-0	DATA1	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. Read accessible always. DATA1 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR39: PCI DATA Register Two (DATA2) Alias Register

Note: *This register is an alias of the DATA register and also of the DATA_SCALE field of the PMCSR register. Since these two are read only, BCR39 provides a means of programming them through the EEPROM. The contents of this register are copied into the corresponding fields pointed with the DATA_SEL field set to two. Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D2_SCALE	These bits correspond to the DATA_SCALE field of the PMCSR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. Read accessible always. D2_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7-0	DATA2	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. Read accessible always. DATA2 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit

BCR40: PCI DATA Register Three (DATA3) Alias Register

Note: *This register is an alias of the DATA register and also of the DATA_SCALE field of the PCMCR register. Since these two are read only, BCR40 provides a means of programming them through the EEPROM. The contents of this register are copied into the corresponding fields pointed with the DATA_SEL field set to three. Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D3_SCALE	These bits correspond to the DATA_SCALE field of the PMCSR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. Read accessible always. D3_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7-0	DATA3	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. Read accessible always. DATA3 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR41: PCI DATA Register Four (DATA4) Alias Register

Note: *This register is an alias of the DATA register and also of the DATA_SCALE field of the PCMCR register. Since these two are read only, BCR41 provides a means of programming them through the EEPROM. The contents of this register are copied into the corresponding fields pointed with the DATA_SEL field set to four. Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description

15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D4_SCALE	These bits correspond to the DATA_SCALE field of the PMC-SR (offset register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. Read accessible always. D4_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit
7-0	DATA4	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. Read accessible always. DATA4 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR42: PCI DATA Register Five (DATA5) Alias Register

Note: *This register is an alias of the DATA register and also of the DATA_SCALE field of the PCMCR register. Since these two are read only, BCR42 provides a means of programming them through the EEPROM. The contents of this register are copied into the corresponding fields pointed with the DATA_SEL field set to five. Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D5_SCALE	These bits correspond to the DATA_SCALE field of the PMC-SR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. Read accessible always. D5_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit
7-0	DATA5	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. Read accessible always. DATA5 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR43: PCI DATA Register Six (DATA6) Alias Register

Note: *This register is an alias of the DATA register and also of the DATA_SCALE field of the PCMCR register. Since these two are read only, BCR43 provides a means of programming them through the EEPROM. The contents of this register are copied into the corresponding fields pointed with the DATA_SEL field set to six. Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D6_SCALE	These bits correspond to the DATA_SCALE field of the PMC-SR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. Read accessible always. D6_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit
7-0	DATA6	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. Read accessible always. DATA6 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR44: PCI DATA Register Seven (DATA7) Alias Register

Note: *This register is an alias of the DATA register and also of the DATA_SCALE field of the PCMCR register. Since these two are read only, BCR44 provides a means of programming them through the EEPROM. The contents of this register are copied into the corresponding fields pointed with the DATA_SEL field set to seven. Bits 15-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D7_SCALE	These bits correspond to the DATA_SCALE field of the PMCSR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. Read accessible always. D7_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7-0	DATA7	These bits correspond to the PCI DATA register (offset register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. Read accessible always. DATA7 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR45: OnNow Pattern Matching Register 1

Note: *This register is used to control and indirectly access the Pattern Match RAM (PMR). When BCR45 is written and the PMAT_MODE bit (bit 7) is 1, Pattern Match logic is enabled. No bus accesses into PMR are possible, and BCR46, BCR47, and all other bits in BCR45 are ignored. When PMAT_MODE is set, a read of BCR45, BCR46, or BCR47 returns all undefined bits except for PMAT_MODE.*

When BCR45 is written and the PMAT_MODE bit is 0, the Pattern Match logic is disabled and accesses to the PMR are possible. Bits 6-0 of BCR45 specify the address of the PMR word to be accessed. Following the write to BCR45, the PMR word may be read by reading BCR45, BCR46 and BCR47 in any order. To write to PMR word, the write to BCR45 must be followed by a write to BCR46 and a write to BCR47 in that order to complete the operation. The RAM will not actually be written until the write to BCR47 is complete. The write to BCR47 causes all 5 bytes (four bytes of BCR46-47 and the upper byte of the BCR45) to be written to whatever PMR word is addressed by bits 6:0 of BCR45.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	PMR_B0	Pattern Match RAM Byte 0. This byte is written into or read from Byte 0 of the Pattern Match RAM Read and write accessible always. PMR_B0 is undefined after H_RESET, and is unaffected by S_RESET and the STOP bit.
7	PMAT_MODE	Pattern Match Mode. Writing a 1 to this bit will enable Pattern Match Mode and should only be done after the Pattern Match RAM has been programmed. Read and write accessible always. PMAT_MODE is reset to 0 after H_RESET, and is unaffected by S_RESET and the STOP bit.
6-0	PMR_ADDR	Pattern Match Ram Address. These bits are the Pattern Match Ram address to be written to or read from. Read and write accessible always. PMR_ADDR is reset to 0 after H_RESET, and is unaffected by S_RESET and the STOP bit.

BCR46: OnNow Pattern Matching Register 2

Note: *This register is used to control and indirectly access the Pattern Match RAM (PMR). When BCR45 is written and the PMAT_MODE bit (bit 7) is 1, Pattern Match logic is enabled. No bus accesses into PMR are possible, and BCR46, BCR47, and all other bits in BCR45 are ignored. When PMAT_MODE is set, a read of BCR45, BCR46, or BCR47 returns all undefined bits except for PMAT_MODE.*

When BCR45 is written and the PMAT_MODE bit is 0, the Pattern Match logic is disabled and accesses to the PMR are possible. Bits 6-0 of BCR45 specify the ad-

dress of the PMR word to be accessed. Following the write to BCR45, the PMR word may be read by reading BCR45, BCR46 and BCR47 in any order. To write to PMR word, the write to BCR45 must be followed by a write to BCR46 and a write to BCR47 in that order to complete the operation. The RAM will not actually be written until the write to BCR47 is complete. The write to BCR47 causes all 5 bytes (four bytes of BCR46-47 and the upper byte of the BCR45) to be written to whatever PMR word is addressed by bits 6:0 of BCR45.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	PMR_B2	Pattern Match RAM Byte 2. This byte is written into or read from Byte 2 of the Pattern Match RAM. Read and write accessible always. PMR_B2 is undefined after H_RESET, and is unaffected by S_RESET and the STOP bit.
7-0	PMR_B1	Pattern Match RAM Byte 1. This byte is written into or read from Byte 1 of Pattern Match RAM. Read and write accessible always. PMR_B1 is undefined after H_RESET, and is unaffected by S_RESET and the STOP bit.

BCR47: OnNow Pattern Matching Register 3

Note: *This register is used to control and indirectly access the Pattern Match RAM (PMR). When BCR45 is written and the PMAT_MODE bit (bit 7) is 1, Pattern Match logic is enabled. No bus accesses into PMR are possible, and BCR46, BCR47, and all other bits in BCR45 are ignored. When PMAT_MODE is set, a read of BCR45, BCR46, or BCR47 returns all undefined bits except for PMAT_MODE.*

When BCR45 is written and the PMAT_MODE bit is 0, the Pattern Match logic is disabled and accesses to the PMR are possible. Bits 6-0 of BCR45 specify the address of the PMR word to be accessed. Following the write to BCR45, the PMR word may be read by reading BCR45, BCR46 and BCR47 in any order. To write to PMR word, the write to BCR45 must be followed by a write to BCR46 and a write to BCR47 in that order to complete the operation. The RAM will not actually be written until the write to BCR47 is complete. The write to BCR47 causes all 5 bytes (four bytes of BCR46-47 and the upper byte of the BCR45) to be written to whatever PMR word is addressed by bits 6:0 of BCR45.

When PMAT_MODE is 0, the contents of the word addressed by bits 6:0 of BCR45 can be read by reading BCR45-47 in any order.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	PMR_B4	Pattern Match RAM Byte 4. This byte is written into or read from Byte 4 of Pattern Match RAM. Read and write accessible always. PMR_B4 is undefined after H_RESET, and is unaffected by S_RESET and the STOP bit.
7-0	PMR_B3	Pattern Match RAM Byte 3. This byte is written into or read from Byte 3 of Pattern Match RAM. Read and write accessible always. PMR_B3 is undefined after H_RESET, and is unaffected by S_RESET and the STOP bit.

BCR48-BCR55: Reserved Locations for Am79C975

These registers must be 00h for the Am79C973 controller.

PHY Management Registers (ANRs)

The Am79C973/Am79C975 device supports the MII basic register set and extended register set. Both sets of registers are accessible through the PHY Management Interface. As specified in the IEEE standard, the basic register set consists of the Control Register (Register 0) and the Status Register (Register 1). The extended register set consists of Registers 2 to 31 (decimal).

Table 39 lists all the registers implemented in the device. All the reserved registers should not be written to, and reading them will return a zero value.

20-23	Reserved	E
24	Summary Status	E
25-31	Reserved	E

Table 39. Am79C973/Am79C975 Internal PHY Management Register Set

Register Address (in Decimal)	Register Name	Basic/ Extended
0	PHY Control	B
1	PHY Status	B
2-3	PHY Identifier	E
4	Auto-Negotiation Advertisement	E
5	Auto-Negotiation Link Partner Ability	E
6	Auto-Negotiation Expansion	E
7	Auto-Negotiation Next Page	E
8-15	Reserved	E
16	Interrupt Enable and Status	E
17	PHY Control/Status	E
18	Descrambler Resynch. Timer	E
19	PHY Management Extension	E

Table 40. ANR0: PHY Control Register (Register 0)

Reg	Bits	Name	Description	Read/Write (Note 1)	Default Value	Soft Reset
0	15	Soft Reset (Note 2)	When write: 1 = PHY software reset, 0 = normal operation. When read: 1 = reset in process, 0 = reset done.	R/W, SC	0	0
0	14	Loopback	1 =asserts the internal LPBCK, 0 = deasserts the internal LPBCK	R/W	0	0
0	13	Speed Selection (Note 3)	1 = 100 Mbps, 0 = 10 Mbps	R/W	1	1
0	12	Auto-Negotiation Enable	1 = enable Auto-Negotiation, 0 = disable Auto-Negotiation	R/W	1	1
0	11	Power Down	1 = power down, 0 = normal operation	R/W	0	0
0	10	Isolate	1 = electrically isolate PHY 0 = normal operation	R/W	1	1
0	9	Restart Auto-Negotiation	1 = restart Auto-Negotiation, 0 = normal operation	R/W, SC	0	0
0	8	Duplex Mode (Note 3)	1 = full duplex, 0 = half duplex	R/W	1	Retains previous value
0	7	Collision Test	1 = enable COL signal test, 0 = disable COL signal test	R/W	0	0
0	6-0	Reserved	Write as 0, ignore on read	RO	0	0

Notes:

1. *R/W = Read/Write, SC = Self Clearing, RO = Read only.*
2. *Soft Reset does not reset the PDX block. Refer to the Soft Reset Section for details.*
3. *Bits 8 and 13 have no effect if Auto-Negotiation is enabled (Bit 12 = 1).*

ANR1: Status Register (Register 1)

The Status Register identifies the physical and Auto-negotiation capabilities of the local PHY. This register is read only; a write will have no effect.

Table 41. ANR1: PHY Status Register (Register 1)

Reg	Bits	Name	Description	Read/Write (Note 1)	Default Value
1	15	100BASE-T4	1 = 100BASE-T4 able, 0 = not 100BASE-T4 able	RO	0
1	14	100BASE-X Full Duplex	1 = 100BASE-X full duplex able, 0 = not 100BASE-X full duplex able	RO	1
1	13	100BASE-X Half Duplex	1 = 100BASE-X half duplex able, 0 = not 100BASE-X half duplex able	RO	1
1	12	10 Mbps Full Duplex	1 = 10 Mbps full duplex able, 0 = not 10 Mbps full duplex able	RO	1
1	11	10 Mbps Half Duplex	1 = 10 Mbps full duplex able, 0 = not 10 Mbps full duplex able	RO	1
1	10-7	Reserved	Ignore when read	RO	0
1	6	MF Preamble Suppression	1 = PHY can accept management (mgmt) frames with or without preamble, 0 = PHY can only accept mgmt frames with preamble	RO	1
1	5	Auto-Negotiation Complete	1 = Auto-Negotiation completed, 0 = Auto-Negotiation not completed	RO	0
1 (Note 1)	4	Remote Fault	1 = remote fault detected, 0 = no remote fault detected	RO, LH	0
1	3	Auto-Negotiation Ability	1 = PHY able to auto-negotiate, 0 = PHY not able to auto-negotiate	RO	1
1 (Note 1)	2	Link Status	1 = link is up, 0 = link is down	RO, LL	0
1	1	Jabber Detect	1 = jabber condition detected, 0 = no jabber condition detected	RO	0
1	0	Extended Capability	1 = extended register capabilities, 0 = basic register set capabilities only	RO	1

Note:

1. *LH = Latching High, LL = Latching Low.*
2. *The link status bit 2 is fed from register 24 bit 3. Register 24 bit 3 tracks the real time state of the PHY, but link status in ANR1 is latched. After a change of the state of the link, ANR1 must be read twice. The first read will provide old information but it causes the link status bit to be updated from register 24 bit 3. The second read will then provide the correct state of the link.*

ANR2 and ANR3: PHY Identifier (Registers 2 and 3)

Registers 2 and 3 contain a unique PHY identifier, consisting of 22 bits of the organizationally unique IEEE Identifier, a 6-bit manufacturer's model number, and a 4-bit manufacturer's revision number. The most significant bit of the PHY identifier is bit 15 of register 2; the least significant bit of the PHY identifier is bit 0 of register 3. Register 2, bit 15 corresponds to bit 3 of the IEEE Identifier and register 2, bit 0 corresponds to bit 18 of the IEEE Identifier. Register 3, bit 15 corresponds to bit 19 of the IEEE Identifier and register 3, bit 10 corresponds to bit 24 of the IEEE Identifier. Register 3, bits 9-4 contain the manufacturer's model number and bits 3-0 contain the manufacturer's revision number. These registers are shown in Table 42 and Table 43.

Table 42. ANR2: PHY Identifier (Register 2)

Reg	Bits	Name	Description	Read/ Write	Default Value	Soft Reset
2	15-0	PHY_ID[31-16]	IEEE Address (bits 3-18); Register 2, bit 15 is MS bit of PHY Identifier	RO	0000000000000000 (0000 Hex)	Retains original Value

Table 43. ANR3: PHY Identifier (Register 3)

Reg	Bits	Name	Description	Read/Write	Default Value	Soft Reset
3	15-10	PHY_ID[15-10]	IEEE Address (bits 19-24)	RO	011010 (1A Hex)	Retains original value
3	9-4	PHY_ID[9-4]	Manufacturer's Model Number (bits 5-0)	RO	110110 (36 Hex)	Retains original value
3	3-0	PHY_ID[3-0]	Revision Number (bits 3-0); Register 3, bit 0 is LS bit of PHY Identifier	RO	0000	Retains original value

ANR4: Auto-Negotiation Advertisement Register (Register 4)

This register contains the advertised ability of the Am79C973/Am79C975 device. The purpose of this register is to advertise the technology ability to the link partner device. See Table 44.

When this register is modified, Restart Auto-Negotiation (Register 0, bit 9) must be enabled to guarantee the change is implemented.

Table 44. ANR4: Auto-Negotiation Advertisement Register (Register 4)

Bit(s)	Name	Description	Read/ Write	H/W or Soft Reset
15	Next Page	When set, the device wishes to engage in next page exchange. If clear, the device does not wish to engage in next page exchange.	R/W	0
14	Reserved		RO	0
13	Remote Fault	When set, a remote fault bit is inserted into the base link code word during the Auto Negotiation process. When cleared, the base link code work will have the bit position for remote fault as cleared.	R/W	0
12:11	Reserved		RO	0
10	PAUSE	This bit should be set if the PAUSE capability is to be advertised.	R/W	0
9	Reserved		RO	0
8	Full Duplex - 100BASE-TX	This bit advertises Full Duplex capability. When set, Full Duplex capability is advertised. When cleared, Full Duplex capability is not advertised.	R/W	1
7	Half duplex - 100BASE-TX	This bit advertises Half Duplex capability for the Auto-negotiation process. Setting this bit advertises Half Duplex capability. Clearing this bit does not advertise Half Duplex capability.	R/W	1
6	Full Duplex - 10BASE-T	This bit advertises Full Duplex capability. When set, Full Duplex capability is advertised. When cleared, Full Duplex capability is not advertised.	R/W	1
5	Half duplex - 10BASE-T	This bit advertises Half Duplex capability for the Auto-negotiation process. Setting this bit advertises Half Duplex capability. Clearing this bit does not advertise Half Duplex capability.	R/W	1
4:0	Selector Field	The Am79C973/Am79C975 device is an 802.3 compliant device	RO	0x01

ANR5: Auto-Negotiation Link Partner Ability Register (Register 5)

The Auto-Negotiation Link Partner Ability Register is Read Only. The register contains the advertised ability of the link partner. The bit definitions represent the received link code word. This register contains either the base page or the link partner's next pages. See Table 45 and Table 46.

Table 45. ANR5: Auto-Negotiation Link Partner Ability Register (Register 5) - Base Page Format

Bit(s)	Name	Description	Read/ Write	H/W or Soft Reset
15	Next Page	Link partner next page request.	RO	0
14	Acknowledge	Link partner acknowledgment	RO	0
13	Remote Fault	Link partner remote fault request	RO	0
12:5	Technology Ability	Link partner technology ability field	RO	0
4:0	Selector Field	Link partner selector field.	RO	0

Table 46. ANR5: Auto-Negotiation Link Partner Ability Register (Register 5) - Next Page Format

Bit(s)	Name	Description	Read/ Write	H/W or Soft Reset
15	Next Page	Link partner next page request.	RO	0
14	Acknowledge	Link partner acknowledgment	RO	0
13	Message Page	Link partner message page request	RO	0
12	Acknowledge 2	1 = Link partner can comply with the request 0 = Link partner cannot comply with the request	RO	0
11	Toggle	Link partner toggle bit.	RO	0
10:0	Message Field	Link partner's message code.	RO	0

ANR6: Auto-Negotiation Expansion Register (Register 6)

The Auto-Negotiation Expansion Register provides additional information which aids the Auto-Negotiation process. The Auto-Negotiation Expansion Register bits are Read Only. See Table 47.

Table 47. ANR6: Auto-Negotiation Expansion Register (Register 6)

Bit(s)	Name	Description	Read/ Write	H/W or Soft Reset
15:5	Reserved		RO	0
4	Parallel Detection Fault	1=Parallel detection fault 0=No parallel detection fault	RO, LH	0
3	Link Partner Next Page Able	1 = Link partner is next page able. 0 = Link partner is not next page able.	RO	0
2	Next Page Able	1 = Am79C973/Am79C975 device channel is next page able 0 = Am79C973/Am79C975 device channel is not next page able	RO	1
1	Page Received	1 = A new page has been received. 0 = A new page has not been received.	RO, LH	0
0	Link Partner ANEG Able	1 = Link partner is Auto-Negotiation able. 0 = Link partner is not Auto-Negotiation able.	RO	0

ANR7: Auto-Negotiation Next Page Register (Register 7)

The Auto-Negotiation Next Page Register contains the next page link code word to be transmitted. On power-up the default value of 0x2001 represents a message page with the message code set to null. See Table 48.

Table 48. ANR7: Auto-Negotiation Next Page Register (Register 7)

Bit(s)	Name	Description	Read/ Write	H/W or Soft Reset
15	Next Page	Am79C973/Am79C975 device channel next page request.	R/W	0
14	Reserved		RO	0
13	Message Page	Am79C973/Am79C975 device channel message page request	R/W	1
12	Acknowledge 2	1 = Am79C973/Am79C975 device channel can comply with the request 0 = Am79C973/Am79C975 device channel cannot comply with the request	R/W	0
11	Toggle	Am79C973/Am79C975 device channel toggle bit.	RO	0
10:0	Message Field	Message code field.	R/W	0x001

Reserved Registers (Registers 8-15, 20-23, and 25-31)

The Am79C973/Am79C975 device contains reserved registers at addresses 8-15, 20-23, and 25-31. These registers should be ignored when read and should not be written at any time.

ANR16: INTERRUPT Status and Enable Register (Register 16)

The Interrupt bits indicate when there is a change in the Link Status, Duplex Mode, Auto-Negotiation status, or Speed status. Register 16 contains the interrupt status and interrupt enable bits. The status is always updated whether or not the interrupt enable bits are set. When an interrupt occurs, the system will need to read the interrupt register to clear the status bits and determine the course of action needed. See Table 49.

Table 49. ANR16: INTERRUPT Status and Enable Register (Register 16)

Bit(s)	Name	Description	Read/ Write	H/W or Soft Reset
15:14	Reserved		RO	0
13	Interrupt Test Enable (Note 1)	1 = When this bit is set, setting bits 12:9 of this register will cause an INTR condition and will set bits 4:1 accordingly. The effect is to test the register bits with a forced interrupt condition. 0 = Bits 4:1 are only set if the interrupt condition (if any bits in 12:9 are set) occurs.	R/W	0
12	Link Status Change Interrupt Enable	1 = Link Status Change 0 = This interrupt is masked.	R/W	0
11	Duplex Mode Change Interrupt Enable	1 = Duplex Mode Change 0 = This interrupt is masked.	R/W	0
10	Auto-Neg Change Interrupt Enable	1 = Auto-Neg Change 0 = This interrupt is masked.	R/W	0
9	Speed Change Interrupt Enable	1 = Speed Change 0 = This interrupt is masked.	R/W	0
8	Global Interrupt Enable	1= Global Interrupt 0 = This interrupt is masked.	R/W	0
7:5	Reserved		RO	0
4	Link Status Change Interrupt	1 = Link Status has changed on a port. 0 = No change in Link Status	RO, LH	0
3	Duplex Mode Change Interrupt	1 = Duplex Mode has changed on a port 0 = No change in Duplex mode	RO, LH	0
2	Auto-Neg Change Interrupt	1 = Auto-Neg status has changed on a port 0 = No change in Auto-Neg status	RO, LH	0
1	Speed Change Interrupt	1 = Speed status has changed on a port 0 = No change	RO, LH	0
0	Global Interrupt	1 = Indicates a change in status of any of the above interrupts 0 = Indicates no change in Interrupt Status	RO, LH	0

Note:
1. All bits, except bit 13, are cleared on read (COR). The register must be read twice to see if it has been cleared.

ANR17: PHY Control/Status Register (Register 17)

This register is used to control the configuration of the 10/100 PHY unit of the Am79C973/Am79C975 device. See Table 50.

When configuring the device to enable/disable the scrambler/descrambler (SDISSCR), and/or to enable/disable the alignment (SDISALIGN), a software reset after a write operation to the appropriate bits in this register is mandatory for proper configuration. If a register bit is only appropriate to use at one speed, then the speed will be indicated in parenthesis in the Name column, for example, (10M).

Table 50. ANR17: PHY Control/Status Register (Register 17)

Reg	Bits	Name	Description	Read/Write	H/W Reset	Soft Reset
17	15	SDISALIGN (100M)	1 = pass unaligned data to internal PHY 0 = enable alignment block	R/W	0	Retains Previous Value
17	14	SDISSCR (100M)	1 = disable scrambler/descrambler 0 = enable scrambler/descrambler	R/W	0	Retains Previous Value
17	13	Force Link Good Enable	1 = link status forced to link up state. 0 = link status is determined by the device.	R/W	0	0
17	12	Disable Link Pulse (10M)	1 = Link pulses sent from the 10BASE-T transmitter are suppressed.	R/W	0	0
17	11	SQE_TEST Disable (10M)	1 = Disables the SQE heartbeat which occurs after each 10BASE-T transmission. 0 = The heart beat assertion occurs on the COL pin approximately 1 µs after transmission and for a duration of 1 µs.	R/W	0	0
17	10	EN_FEFI (100M)	1 = enable FEFI, 0 = disable FEFI This bit is ignored if auto-neg is enabled.	R/W	0	0
17	9	Jabber Detect Disable (10M)	1 = disable jabber detect 0 = enable jabber detect	R/W	0	0
17	8:7	LBK[1-0] (100M)	00 = normal operation 01 = unused 10 = unused 11 = serial loopback	R/W	00	00
17	6	Receive Polarity Reversed (10M)	1 = Receive polarity of the 10BASE-T receiver is reversed. 0 = Receive polarity is correct.	RO	0	0
17	5	Auto Receive Polarity Correction Disable (10M)	1 = polarity correction circuit is disabled for 10BASE-T. 0 = Self correcting polarity circuit is enabled	R/W	0	0
17	4	Extended Distance Enable (10M)	1 = 10BASE-T receive squelch thresholds are reduced to allow reception of frames which are greater than 100 meters. 0 = Squelch thresholds are set for standard distance of 100 meters.	R/W	0	0
17	3	TX_DISABLE	1 = TX± outputs not active for MLT-3 and 10BASE-T. TX± outputs to logical “0” for PECL. 0 = Transmit valid data.	R/W	0	0
17	2	TX_CRS	1 = CRS is asserted when transmit or receive medium is active. 0 = CRS is asserted when receive medium is active.	RO	1	1
17	1	Reserved	Reserved.	RO	0	0
17	0	PHY Isolated	1 = Internal PHY is isolated 0 = Internal PHY is enabled	RO	0/1	0/1

ANR18: Descrambler Resynchronization Timer Register (Register 18)

Descrambler Resynchronization Timer Register (shown in Table 51) allows the user to program the time it takes for the descrambler to start the resynchronization process. This is to ensure that the Descrambler resynchronizes itself to the next IDLE symbol stream after it receives a packet of excessive length. This register should be programmed as described in the Table 51. The programmed timer value should always be greater than the length of the maximum size packet in normal operation.

Table 51. ANR18: Descrambler Resynchronization Timer (Register 18)

Reg	Bits	Name	Description	Read/ Write	Default Value	Soft Reset
18	15-0	Descrambler Resynch Timer	Each bit indicates 4 clocks, or 160 ns. The count decrements from a default value of 1 ms or an initial value loaded by the user. This counter provides a maximum timer value of 10.5 ms.	R/W	0001100001101010 (Note 1)	0001100011010 10 (Note 1)

Note:

1. The corresponding time to this setting is 1ms.

ANR19: PHY Management Extension Register (Register 19)

Table 52 contains the PHY Management Extension Register (Register 19) bits.

Table 52. ANR19: PHY Management Extension Register (Register 19)

Reg	Bits	Name	Description	Read/Write	Default Value	Soft Reset
19	15:6	Reserved	Write as 0, ignore on read.	RO	0	0
19	5	Mgmt Frame Format	1 = last management frame was invalid (opcode error, etc.); 0 = last management frame was valid.	RO	0	0
19	4-0	PHY Address	PHY Address defaults to 11110.	RO	11110	Retains Previous Value

ANR24: Summary Status Register (Register 24)

The Summary Status register is a global register containing status information. This register is Read/Only and represents the most important data which a single register access can convey. The Summary Status register indicates the following: Link Status, Full Duplex Status, Auto-Negotiation Alert, and Speed. See Table 53.

Table 53. ANR24: Summary Status Register (Register 24)

Bit(s)	Name	Description	Read/ Write	H/W or Soft Reset
15-4	Reserved	Write as 0; Ignore on Read	0	0
3	Link Status	1 = Link Status is up. 0 = Link Status is down.	R/O	0
2	Full Duplex	1 = Operating in full duplex mode 0 = Operating in half duplex mode	R/O	0
1	AutoNEG Alert	1 = AutoNEG status has changed 0 = AutoNEG status unchanged	R/O	0
0	Speed	1 = Operating at 100 Mbps 0 = Operating at 10 Mbps	R/O	1

Initialization Block

Note: *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 54.*

Note: *The Am79C973/Am79C975 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 55.

Table 54. 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	TDRA 23-16	
IADR+14h	TDRA 15-00				
IADR+16h	TLEN	0	RES	TDRA 23-16	

Table 55. 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 56. If a value other than those listed in Table 56 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 57.

If a value other than those listed in Table 57 is desired, CSR76 and CSR78 can be written after initialization is complete.

Table 56. 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

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 57. 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. If the DRCVBC bit (CSR15, bit 14) is set as well, the broadcast packets will be rejected. See Figure 51.

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 Am79C973/ Am79C975 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.

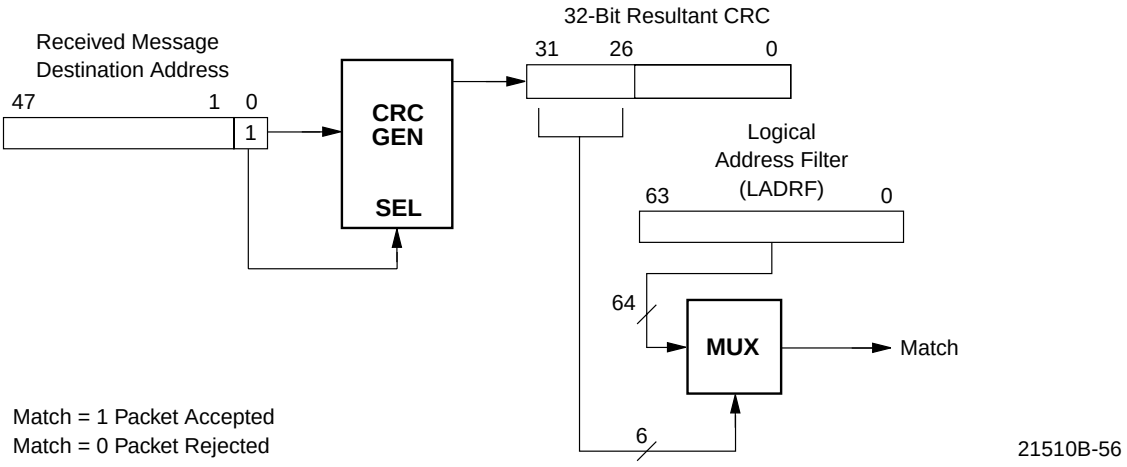


Figure 51. Address Match Logic

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 58 (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 59 (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 60 (CRDA = Current Receive Descriptor Address).

Table 58. 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 59. 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 60. Receive Descriptor (SWSTYLE = 3)

Address	31	30	29	28	27	26	25	24	23	22-16	15-12	11-0
CRDA+00h	RES								RES	RES	0000	MCNT
CRDA+04h	OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	BPE	RES	1111	BCNT
CRDA+08h	RBADR[31:0]											
CRDA+0Ch	USER SPACE											

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 Am79C973/Am79C975 controller (OWN = 1). The Am79C973/Am79C975 controller clears the OWN bit after filling the buffer that the descriptor points to. The host sets the OWN bit after emptying the buffer. Once the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host. CRC will also be set when Am79C973/Am79C975 receives an RX_ER indication from the external PHY through the MII.
26	BUFF	Buffer error is set any time the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host.

25 STP — Start of Packet indicates that this is the first buffer used by the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host.

23 BPE — Bus Parity Error is set by the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host.

This bit does not exist when the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host.

This bit does not exist when the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host. This bit does not exist when the Am79C973/Am79C975 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 Am79C973/Am79C975 controller.
11-0	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 Am79C973/Am79C975 controller.

RMD2

Bit	Name	Description
31	ZERO	This field is reserved. The Am79C973/Am79C975 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. Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host.

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 61 (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 62 (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 63 (CXDA = Current Transmit Descriptor Address).

Table 61. 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 62. 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 63. 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												

TMD0

Bit	Name	Description
31-0	TBADR	Transmit Buffer address. This field contains the address of the transmit buffer that is associated with this descriptor.

TMD1

Bit	Name	Description
31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C973/Am79C975 controller (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The Am79C973/Am79C975 controller clears the OWN bit after transmitting the contents of the buffer. Both the Am79C973/ Am79C975 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 Am79C973/Am79C975 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. This bit should be set with the ENP bit. However, for backward compatibility, it is recommended that this bit be set for every descriptor of the intended frame. 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and is read by the host. When LTINTEN is cleared to 0 (CSR5, bit 14), the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller. This bit has meaning only if the ENP bit is set.

26 DEF — Deferred indicates that the Am79C973/Am79C975 controller had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the Am79C973/Am79C975 controller is ready to transmit. DEF is set by the Am79C973/Am79C975 controller and cleared by the host.

25 STP — Start of Packet indicates that this is the first buffer to be used by the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller.

24 ENP — End of Packet indicates that this is the last buffer to be used by the Am79C973/Am79C975 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 Am79C973/Am79C975 controller.

23 BPE — Bus Parity Error is set by the Am79C973/Am79C975 controller

when a parity error occurred on the bus interface during a data transfers from the transmit buffer associated with this descriptor. The Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host.

This bit does not exist, when the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller. This field is written by the host and is not changed by the Am79C973/Am79C975 controller. There are no minimum buffer size restrictions.

TMD2

Bit	Name	Description
31	BUFF	Buffer error is set by the Am79C973/Am79C975 controller during transmission when the Am79C973/Am79C975 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 Am79C973/Am79C975 controller obtained the STATUS byte (TMD1[31:24]) of the next descriptor. BUFF is set by the Am79C973/Am79C975 controller and cleared by the host. If a Buffer Error occurs, an Underflow Error will also occur. BUFF is set by the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller and cleared by the host.
29	EXDEF	Excessive Deferral. Indicates that the transmitter has experienced Excessive Deferral on this transmit frame, where Excessive Deferral is defined in 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 Am79C973/Am79C975 controller does not retry on late collisions. LCOL is set by the Am79C973/Am79C975 controller and cleared by the host.
27	LCAR	Loss of Carrier is set when the carrier is lost during an Am79C973/Am79C975 controller

		initiated transmission when operating in half-duplex mode. The Am79C973/Am79C975 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. LCAR is not valid in Internal Loopback Mode. LCAR is set by the Am79C973/Am79C975 controller and cleared by the host. LCAR will be set when the PHY is in Link Fail state during 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 Am79C973/ Am79C975 controller and cleared by the host.
25-4	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 Am79C973/ Am79C975 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	0290h
08h	PCI Revision ID	8	RO	40h
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
31h - 33h	Reserved	8	RO	00h
34h	Capabilities Pointer	8	RO	40h
35h - 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	PCI Capability Identifier	8	RO	01h
41h	PCI Next Item Pointer	8	RO	00h
42h	PCI Power Management Capabilities	16	RO	00h
44h	PCI Power Management Control/Status	8	RO	00h
46h	PCI PMCSR Bridge Support Extensions	8	RO	00h
47h	PCI Data	8	RO	00h
48h - 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	Am79C973/Am79C975 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	CSR19	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	POLL: Poll Time Counter	T
47	CSR47	uuuu uuuu	PI: Polling Interval	S
48	CSR48	uuuu uuuu	Reserved	
49	CSR49	uuuu uuuu	Reserved	
50	CSR50	uuuu uuuu	Reserved	
51	CSR51	uuuu uuuu	Reserved	
52	CSR52	uuuu uuuu	Reserved	
53	CSR53	uuuu uuuu	Reserved	
54	CSR54	uuuu uuuu	Reserved	
55	CSR55	uuuu uuuu	Reserved	
56	CSR56	uuuu uuuu	Reserved	
57	CSR57	uuuu uuuu	Reserved	
58	CSR58	see 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	262 5003 (Am79C973) 262 7003 (Am79C975)	Chip ID Register Lower	T
89	CSR89	uuuu 262	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	0000 0000	OnNow Miscellaneous	**S**
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	CSR122	uuuu 0000	Receive Frame Alignment Control	**S**
123	CSR123	uuuu uuuu	Reserved	
124	CSR124	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	Reserved	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	PHY Control and Status	Yes	Yes
33	MIIADDR	N/A	PHY Address	Yes	Yes
34	MIIMDR	N/A	PHY Management Data	Yes	No
35	PCIVID	1022h	PCI Vendor ID	No	Yes
36	PMC_A	C811h	PCI Power Management Capabilities (PMC) Alias Register	No	Yes
37	DATA0	0000h	PCI DATA Register Zero Alias Register	No	Yes
38	DATA1	0000h	PCI DATA Register One Alias Register	No	Yes
39	DATA2	0000h	PCI DATA Register Two Alias Register	No	Yes
40	DATA3	0000h	PCI DATA Register Three Alias Register	No	Yes
41	DATA4	0000h	PCI DATA Register Four Alias Register	No	Yes
42	DATA5	0000h	PCI DATA Register Five Alias Register	No	Yes
43	DATA6	0000h	PCI DATA Register Six Alias Register	No	Yes
44	DATA7	0000h	PCI DATA Register Seven Alias Register	No	Yes
45	PMR1	N/A	Pattern Matching Register 1	Yes	No
46	PMR2	N/A	Pattern Matching Register 2	Yes	No
47	PMR3	N/A	Pattern Matching Register 3	Yes	No

PHY Management Registers

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

Register Address	Symbol	Name	Default Value After H_RESET
0	ANR0	PHY Control Register	2500h
1	ANR1	PHY Status Register	7849h
2	ANR2	PHY_ID[31:16]	0000h
3	ANR3	PHY_ID[15:0]	6BA0h
4	ANR4	Auto-Negotiation Advertisement Register	03C1h
5	ANR5	Auto-Negotiation Link Partner Ability Register	0000h
6	ANR6	Auto-Negotiation Expansion Register	0004h
7	ANR7	Auto-Negotiation Next Page Register	2001h
8-15	ANR8-ANR15	Reserved	--
16	ANR16	Interrupt Status and Enable Register	0000h
17	ANR17	PHY Control/Status Register	0001h
18	ANR18	Descrambler Resynchronization Timer Register	186Ah
19	ANR19	PHY Management Extension Register	--
20-23	ANR20-ANR23	Reserved	--
24	ANR24	Summary Status Register	0001h
25-31	ANR25-ANR31	Reserved	--

PROGRAMMABLE REGISTER SUMMARY

Am79C973/Am79C975 Control and Status Registers

Register	Contents			
CSR0	Status and control bits: (DEFAULT = 0004)			
	8000 ERR 4000 -- 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 -- 2000 -- 1000 MISSM	0800 MERRM 0400 RINTM 0200 TINTM 0100 IDONM	0080 -- 0040 DXSUFLO 0020 LAPPEN 0010 DXMT2PD	0008 EMBA 0004 BSWP 0002 -- 0001 --
CSR4	Interrupt masks, configuration and status bits: (DEFAULT = 0115)			
	8000 -- 4000 DMAPLUS 2000 -- 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 -- 0001 --
CSR5	Extended Interrupt masks, configuration and status bits: (DEFAULT = 0XXX)			
	8000 TOKINTD 4000 LTINTEN 2000 -- 1000 --	0800 SINT 0400 SINTE 0200 -- 0100 --	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 11 PHY Selected 10 Reserved			
	8000 PROM 4000 DRCVBC 2000 DRCVPA 1000 --	0800 -- 0400 -- 0200 -- 0100 PORTSEL1	0080 PORTSEL0 0040 INTL 0020 DRTY 0010 FCOLL	0008 DXMTFCS 0004 LOOP 0002 DTX 0001 DRX
CSR47	TXPOLLINT: Transmit Polling Interval			
CSR49	RXPOLLINT: Receive Polling Interval			
CSR58	Software Style (mapped to BCR20) bits [7:0] = SWSTYLE, Software Style Register. 0000 LANCE/PCnet-ISA 0002 PCnet-32			
	8000 -- 4000 -- 2000 -- 1000 --	0800 -- 0400 APERREN 0200 -- 0100 SSIZE32	0080 -- 0040 -- 0020 -- 0010 --	0008 SWSTYLE3 0004 SWSTYLE2 0002 -- 0001 SWSTYLE0

Am79C973/Am79C975 Control and Status Registers (Concluded)

Register	Contents
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 20/36 (No SRAM/SRAM) 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 220 max/Full Packet (No SRAM/SRAM with UFLO bit set) bytes have been written bits [9:8] = XMTFW, Transmit FIFO Watermark 0000 Start DMA when 16 write cycles can be made 0100 Start DMA when 32 write cycles can be made 0200 Start DMA when 64 write cycles can be made 0300 Start DMA when 128 write cycles can be made bits [7:0] = DMA Burst Register
CSR88~89	Chip ID (Contents = v2625003 (for Am79C973); v = Version Number) Chip ID (Contents = v2627003 (for Am79C975); v = Version Number)
CSR112	Missed Frame Count
CSR114	Receive Collision Count
CSR116	OnNow Miscellaneous
	8000 -- 4000 -- 2000 -- 1000 -- 0800 -- 0400 -- 0200 PME_EN_OVR 0100 LCDET 0080 PMAT 0040 EMPPLBA 0020 MPMAT 0010 MPPEN 0008 RWU_DRIVER 0004 RWU_GATE 0002 RWU_POL 0001 RST_POL
CSR122	Receive Frame Alignment Control
	8000 -- 4000 -- 2000 -- 1000 -- 0800 -- 0400 -- 0200 -- 0100 -- 0080 -- 0040 -- 0020 -- 0010 -- 0008 -- 0004 -- 0002 -- 0001 RCVALGN
CSR124	BMU Test Register (DEFAULT = 0000)
	8000 -- 4000 -- 2000 -- 1000 -- 0800 -- 0400 -- 0200 -- 0100 -- 0080 -- 0040 -- 0020 -- 0010 -- 0008 -- 0004 RPA 0002 -- 0001 --
CSR125	MAC Enhanced Configuration Control (DEFAUT = 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)

Am79C973/Am79C975 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	--	0400	--	0040	--	0004	--
		2000	--	0200	--	0020	--	0002	ASEL 0001
		1000	--	0100	APROMWE	0010	--	--	
4	LED0	Programs the function and width of the LED0 signal. (DEFAULT = 00C0)							
		8000	LEDOUT	0800	--	0080	PSE	0008	--
		4000	LEDPOL	0400	--	0040	LNKSE	0004	RCVE
		2000	LEDDIS	0200	MPSE	0020	RCVME	0002	--
		1000	100E	0100	FDLSE	0010	XMTE	0001	COLE
5	LED1	Programs the function and width of the LED1 signal. (DEFAULT = 0084)							
		8000	LEDOUT	0800	--	0080	PSE	0008	--
		4000	LEDPOL	0400	--	0040	LNKSE	0004	RCVE
		2000	LEDDIS	0200	MPSE	0020	RCVME	0002	--
		1000	100E	0100	FDLSE	0010	XMTE	0001	COLE
6	LED2	Programs the function and width of the LED2 signal. (DEFAULT = 0088)							
		8000	LEDOUT	0800	--	0080	PSE	0008	--
		4000	LEDPOL	0400	--	0040	LNKSE	0004	RCVE
		2000	LEDDIS	0200	MPSE	0020	RCVME	0002	--
		1000	100E	0100	FDLSE	0010	XMTE	0001	COLE
7	LED3	Programs the function and width of the LED3 signal. (DEFAULT = 0090)							
		8000	LEDOUT	0800	--	0080	PSE	0008	--
		4000	LEDPOL	0400	--	0040	LNKSE	0004	RCVE
		2000	LEDDIS	0200	MPSE	0020	RCVME	0002	--
		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	--
		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	--
		4000	ROMTMG2	0400	--	0040	BREADE	0004	--
		2000	ROMTMG1	0200	MEMCMD	0020	BWRITE	0002	--
		1000	ROMTMG0	0100	EXTREQ	0010	--	0001	--
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
20	SWSTYLE	Software Style (DEFAULT = 0000, maps to CSR 58)							

Am79C973/Am79C975 Bus Configuration Registers (Concluded)

RAP Addr	Register	Contents
22	PCILAT	PCI Latency (DEFAULT = FF06)
		bits [15:8] = MAX_LAT bits [7:0] = MIN_GNT
25	SRAMSIZE	SRAM Size (DEFAULT = 0000)
		bits [7:0] = SRAM_SIZE
26	SRAMBND	SRAM Boundary (DEFAULT = 0000)
		bits [7:0] = SRAM_BND
27	SRAMIC	SRAM Interface Control (Default = 0000)
		8000PTR TST 4000LOLATRX bits [5:3] = EBCS, Expansion Bus Clock Source 0000 CLK pin, PCI clock 0008 Time Base Clock 0010 EBCLK pin, Expansion Bus Clock bits [2:0] = CLK_FAC, Expansion Bus Clock Factor 0000 1/1 clock factor 0001 1/2 clock factor 0002 -- 0003 --
28	EPADDRL	Expansion Port Address Lower (Default = 0000)
29	EPADDRU	Expansion Port Address Upper (Default = 0000)
		8000 FLASH 4000 LAINC 2000 -- 1000 -- 0800 -- 0400 -- 0200 -- 0100 -- 0080 -- 0040 -- 0020 -- 0010 -- 0008 EPADDRU3 0004 EPADDRU2 0002 EPADDRU1 0001 EPADDRU0
30	EBDATA	Expansion Bus Data Port
31	STVAL	Software Timer Interrupt Value (DEFAULT = FFFF)
32	MIICAS	PHY Status and Control (DEFAULT = 0000)
		8000 *ANTST* 4000 MIIPD 2000 FMDC1 1000 FMDC0 0800 APEP 0400 APDW2 0200 APDW1 0100 APDW0 0080 DANAS 0040 XPHYRST 0020 XPHYANE 0010 XPHYFD 0008 XPHYSP 0004 -- 0002 MIILP 0001 --
33	MIIADDR	PHY Address (DEFAULT = 0000)
		bits [9:5] = PHYAD, Physical Layer Device Address bits [4:0] = REGAD, Auto-Negotiation Register Address
34	MIIMDR	PHY Data Port
35	PCI Vendor ID	PCI Vendor ID Register (DEFAULT = 1022h)
36	PMC Alias	PCI Power Management Capabilities (DEFAULT = 0000)
37	DATA 0	PCI Data Register Zero Alias Register (DEFAULT = 0000)
38	DATA 1	PCI Data Register One Alias Register (DEFAULT = 0000)
39	DATA 2	PCI Data Register Two Alias Register (DEFAULT = 0000)
40	DATA 3	PCI Data Register Three Alias Register (DEFAULT = 0000)
41	DATA 4	PCI Data Register Four Alias Register (DEFAULT = 0000)
42	DATA 5	PCI Data Register Five Alias Register (DEFAULT = 0000)
43	DATA 6	PCI Data Register Six Alias Register (DEFAULT = 0000)
44	DATA 7	PCI Data Register Seven Alias Register (DEFAULT = 0000)
45	PMR 1	OnNow Pattern Matching Register 1
46	PMR 2	OnNow Pattern Matching Register 2
47	PMR 3	OnNow Pattern Matching Register 3

ABSOLUTE MAXIMUM RATINGS

Storage Temperature . . . . . –65°C to +150°C

Ambient Temperature. . . . . -65°C to +70°C

Supply voltage with respect to V_{SSB}, V_{SS}, D_{VSSD}, D_{VSSP}, and D_{VSSX} . . . . . –0.3 V to 3.63 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:

V_{DD}, V_{DDB}, V_{DD_PCI}, ... +3.0 V to 3.6 V

D_{VDDD}, D_{VDDA}, D_{VDDP}, D_{VDDTX}, D_{VDDRX}, and D_{VDDCO} . . . . . .+3.3 V ±5%

All inputs within the range: . . . . . V_{SS} - 0.5 V to 5.5 V

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 I/O (Non-PCI Pins)					
V_{IH}	Input HIGH Voltage		2.0		V
V_{IL}	Input LOW Voltage			0.8	V
V_{OL}	Output LOW Voltage	I_{OL1} = 4 mA I_{OL2} = 6 mA I_{OL3} = 12 mA (Note 1)		0.4	V
V_{OH}	Output HIGH Voltage (Notes 2, 3)	I_{OH1}= -4 mA I_{OH2}= -2 mA (Note 3)	2.4		V
I_{OZ}	Output Leakage Current (Note 4)	0 V <V_{OUT} <V_{DD}	-10	10	µA
I_{IX}	Input Leakage Current (Note 5)	0 V <V_{IN} <V_{DD}	-10	10	µA
I_{IL}	Input LOW Current (Note 6)	V_{IN} = 0 V; V_{DD} = 3.6 V	-200	-10	µA
I_{IH}	Input HIGH Current (Note 6)	V_{IN} = 2.7 V; V_{DD} = 3.6 V	-50	10	µA
PCI Bus Interface - 5 V Signaling					
V_{IH}	Input HIGH Voltage		2.0	5.5	V
V_{IL}	Input LOW Voltage		-0.5	0.8	V
I_{OZ}	Output Leakage Current (Note 4)	0 V <V_{IN} < V_{DD_PCI}	-10	10	µA
I_{IL}	Input LOW Current	V_{IN} = 0.5 V	--	-70	µA
I_{IH}	Input HIGH Current	V_{IN} = 2.7 V	--	70	µA
I_{IX_PME}	Input Leakage Current (Note 7)	0 V = < V_{IN} < 5.5 V	-1	1	µA
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH} = -2 mA	2.4		V
V_{OL}	Output LOW Voltage	I_{OL}4 = 3 mA I_{OL}2 = 6 mA (Note 1)		0.55	V
PCI Bus Interface - 3.3 V Signaling					
V_{IH}	Input HIGH Voltage		0.5 V_{DD_PCI}	V_{DD_PCI} + 0.5	V
V_{IL}	Input LOW Voltage		-0.5	0.3 V_{DD_PCI}	V
I_{OZ}	Output Leakage Current (Note 4)	0 V < V_{OUT} < V_{DD_PCI}	-10	10	µA
I_{IL}	Input HIGH Current	0 V < V_{IN} < V_{DD_PCI}	-10	10	µA
I_{IX_PME}	Input Leakage Current (Note 7)	0 V = < V_{IN} < 5.5 V	-1	1	µA
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH} = -500 µA	2.4		V
V_{OL}	Output LOW Voltage	I_{OL} = 1500 µA		0.1 V_{DD_PCI}	V

DC CHARACTERISTICS OVER COMMERCIAL OPERATING RANGES unless otherwise specified (Concluded)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
Pin Capacitance					
C_{IN}	Pin Capacitance	F_C = 1 MHz (Note 8)		10	pF
C_{CLK}	CLK Pin Capacitance	F_C = 1 MHz (Notes 8,9)	5	12	pF
C_{IDSEL}	IDSEL Pin Capacitance	Fc = 1 MHz (Notes 8, 10		8	pF
LPIN	Pin Inductance	Fc = 1 MHz (Note 8)		20	nH
Power Supply Current (Note 11)					
I_{DD}	Dynamic Current	PCI CLK at 33 MHz		320.0	mA
I_{DD_WU1}	Wake-up current when the device is in the D1, D2, or D3 state and the PCI bus is in the B0 or B1 state.	PCI CLK at 33 MHz, Device in Magic Packet or OnNow mode, receiving non-matching packets		300.0	mA
I_{DD_WU2}	Wake-up current when the device is in the D2 or D3 state and the PCI bus is in the B2 or B3 state.	PCI CLK LOW, PG LOW, Device at Magic Packet or OnNow mode, receiving non-matching packets		290.0	mA
I_{DD_S}	Static I_{DD}	PCI CLK, $\overline{RST}$, and RST HIGH.		135.0	mA

Notes:

1. *I_{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], $\overline{EROMCS}$, AS_$\overline{EBOE}$, $\overline{EBWE}$, and PHY_RST.*
   I_{OL3} applies to LED0, LED1, LED2, LED3, and WUMI.
   I_{OL4} applies to AD[31:0], C/$\overline{BE}$[3:0], PAR, and $\overline{REQ}$ pins in 5 V signalling environment.
2. *V_{OH} does not apply to open-drain output pins.*
3. *I_{OH2} applies to all other outputs.*
4. *I_{OZ} applies to all output and bidirectional pins, except the $\overline{PME}$ pin. Tests are performed at V_{IN} = 0 V and at V_{DD} only.*
5. *I_{IX} applies to all input pins except $\overline{PME}$, TDI, TCLK, and TMS pins.*
6. *I_{IL} and I_{IH} apply to the TDI, TCLK, and TMS pins.*
7. *I_{IX_PME} applies to the $\overline{PME}$ pin only. Tests are performed at V_{IN} = 0 V and 5.5 V only.*
8. *Parameter not tested. Value determined by characterization.*
9. *C_{CLK} applies only to the CLK pin.*
10. *C_{IDSEL} applies only to the IDSEL pin.*
11. *Power supply current values listed here are preliminary estimates and are not guaranteed.*

SWITCHING CHARACTERISTICS: BUS INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
Clock Timing					
F_{CLK}	CLK Frequency		0	33	MHz
t_{CYC}	CLK Period	@ 1.5 V for 5 V signaling @ 0.4 V_{DD} for 3.3 V signaling	30	–	ns
t_{HIGH}	CLK High Time	@ 2.0 V for 5 V signaling @ 0.4 V_{DD} for 3.3 signaling	12		ns
t_{LOW}	CLK Low Time	@ 0.8 V for 5 V signaling @ 0.3 V_{DD} for 3.3 V signaling	12		ns
t_{FALL}	CLK Fall Time	over 2 V p-p for 5 V signaling over 0.4 V_{DD} for 3.3 V signaling (Note 1)	1	4	V/ns
t_{RISE}	CLK Rise Time	over 2 V p-p for 5 V signaling over 0.4 V_{DD} for 3.3 V signaling (Note 1)	1	4	V/ns
Output and Float Delay Timing					
t_{VAL}	$\overline{\text{AD}}[31:00]$, C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$, $\overline{\text{PERR}}$, $\overline{\text{SERR}}$ Valid Delay		2	11	ns
t_{VAL} ($\overline{\text{REQ}}$)	$\overline{\text{REQ}}$ Valid Delay		2	12	ns
t_{ON}	$\overline{\text{AD}}[31:00]$, C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$ Active Delay		2		ns
t_{OFF}	$\overline{\text{AD}}[31:00]$, C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$ Float Delay			28	ns
Setup and Hold Timing					
t_{SU}	$\overline{\text{AD}}[31:00]$, C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$, IDSEL Setup Time		7		ns
t_H	$\overline{\text{AD}}[31:00]$, C/$\overline{\text{BE}}$[3:0], PAR, $\overline{\text{FRAME}}$, $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$, IDSEL Hold Time		0		ns
t_{SU} ($\overline{\text{GNT}}$)	$\overline{\text{GNT}}$ Setup Time		10		ns
t_H (GNT)	GNT Hold Time		0		ns

SWITCHING CHARACTERISTICS: BUS INTERFACE (CONCLUDED)

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
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.*

Analog I/O - PECL Mode

Symbol	Parameter Description	Test Conditions	Minimum	Maximum	Unit
V_I	Input Voltage Range (Note 1)		2.0 V	D_{VDD}-0.30	V
V_{OH}	Output High Voltage	PECL Load (Notes 2, 3)	D_{VDD}-0.45	D_{VDD}-0.30	V
V_{OL}	Output Low Voltage	PECL Load (Notes 2, 3)	D_{VDD}-1.4	D_{VDD}-1.15	V
V_{DIFF}	Input Differential Voltage (Note 1)	D_{VDD}=Maximum	0.40	1.1	V

Notes:

1. *Applies to RX+, RX-, SDI+, SDI- inputs only. Any voltage applied to these pins must not be below V_I min or above V_I max.*
2. *Tested for* D_{VDD} *= Minimum, shown limits are specified over entire* D_{VDD} *operating range.*
3. *Applies to TX+,TX- outputs only. Measured with the load of 82* W *to D_{VDD} and 150* W *to D_{VSS} on each SDI+ and SDI-.*

SWITCHING CHARACTERISTICS: EXTERNAL ADDRESS DETECTION INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
External Address Detection Interface: Internal PHY @ 25 MHz					
t_{EAD7}	SFBD change to ↓ RX_CLK		0	20 (Note 1)	ns
tEAD8	EAR deassertion to ↑ RX_CLK (first rising edge)		40		ns
t_{EAD9}	EAR assertion after SFD event (frame rejection)		0	5,080	ns
t_{EAD10}	$\overline{EAR}$ assertion width		50		ns
External Address Detection Interface: Internal PHY @ 2.5 MHz					
t_{EAD11}	$\overline{EAR}$ deassertion to ↑ RX_CLK (first rising edge)		400		ns
t_{EAD12}	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 to ↑SF/BD (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. Analog I/O - MLT-3 Mode

Analog I/O - MLT-3 Mode

Symbol	Parameter Description	Test Conditions	Minimum	Maximum	Unit
V_{TXD}	Output Peak Voltage		950	1050	mV
V_{SDA}	Input Differential Assert Threshold (peak to peak)		--	1,000	mV
V_{SDD}	Input Differential Deassert Threshold (peak to peak)		200	--	mV
I_{IX}	Input Leakage Current		-10	10	µA

10BASE-T Mode

Symbol	Parameter Description	Test Conditions	Minimum	Maximum	Unit
V_{OUT}	Output Voltage on TX± (peak)		1.55	1.98	V
V_{DIFF}	Input Differential Squelch Assert on RX± (peak)		300	520	mV

V_{DIFF}	Input Differential De-Assert Voltage on RX± (peak)		150	300	mV
I_{IX}	Input Leakage Current		-10	10	µa

Note: *V_{OUT} reflects output levels prior to 1:÷2 transformer stage.*

EXTERNAL CLOCK

Figure 52 External Clock Timing

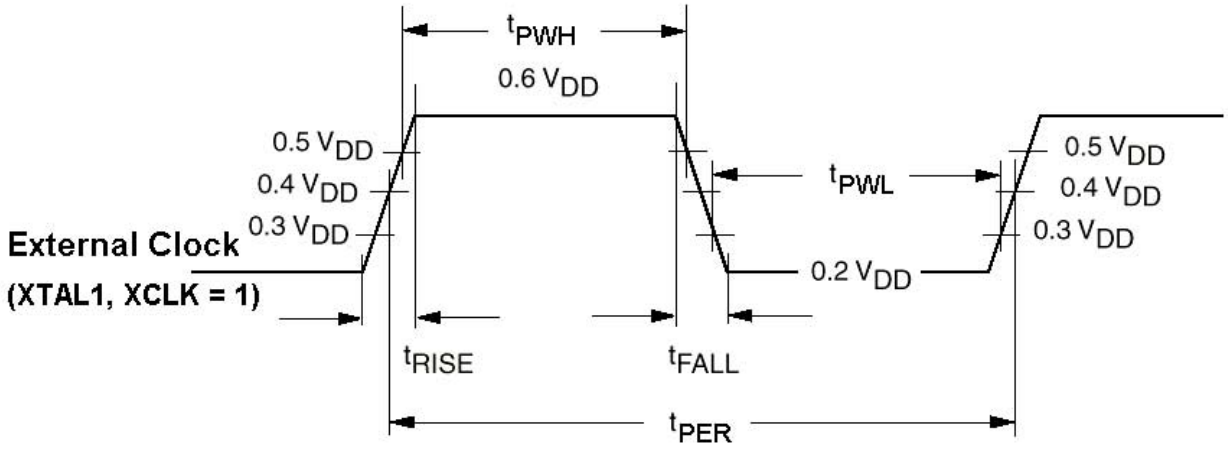


Table 64. Clock (XTAL1, XCLK = 1) Switching Characteristics

Parameter Symbol	Parameter Name	Test Condition	Min	Nom	Max	Unit
F_{XTAL1}	XTAL1 Frequency		-	25	-	MHz
t_{PER}	XTAL1 Period	@ 0.4 V_{DD} (3.3V)	-	40	-	ns
t_{PWH}	XTAL1 High Time	@ 0.4 V_{DD} (3.3V)	18	22	-	ns
t_{PWL}	XTAL1 Low Time	@ 0.3 V_{DD} (3.3V)	18	22	-	ns
t_{RISE}	XTAL1 Rise Time	over 0.4 V_{DD} (3.3V)	1	4		V/ns
t_{FALL}	XTAL1 Fall Time	over 0.4 V_{DD} (3.3V)	1	4		V/ns

Table 65. Crystal (XTAL1, XTAL2, XCLK = 0) Requirements

Item	Test Condition	Min	Nom	Max	Unit
Frequency		-	25	-	MHz
Drive Level		-	-	-	mW
Load Capacitance		-	18	-	pF
Frequency Stability		-	-	50 to 100	ppm
ESR		-	-	50	ohms

Table 66. Crystal (XTAL1, XTAL2, XCLK = 0) Requirements

Component	Manufacturer	Part Number
Crystal	Epson America	MA-505-25.000M
Crystal	Ecliptek	EC-AT-25.000M
Crystal	Ecliptek	ECSM-AT-25.000M

PMD Interface

PECL

No.	Symbol	Parameter Description	Test Conditions	Min	Max	Unit
160	t_R (Note 1)	TX+, TX- Rise Time	PECL Load	0.5	3	ns
161	t_F (Note 1)	TX+, TX- Fall Time	PECL Load	0.5	3	ns
162	t_{SK} (Note 1)	TX+ to TX- skew	PECL Load	--	±200	ps

Note:
1. Not included in the production test.

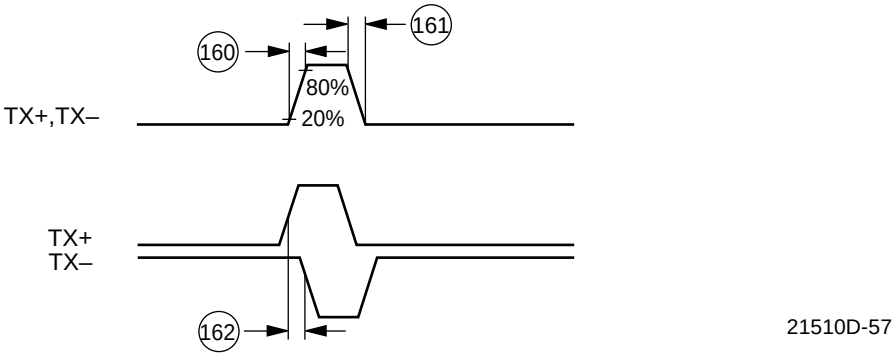


Figure 53 PMD Interface Timing (PECL)

MLT-3

No.	Symbol	Parameter Description	Test Conditions	Min	Max	Unit
170	t_R (Note 1)	TX+, TX- Rise Time		3	5	ns
171	t_F (Note 1)	TX+, TX- Fall Time		3	5	ns
172	t_{SK} Note 1)	TX+ to TX- skew		--	±250	ps

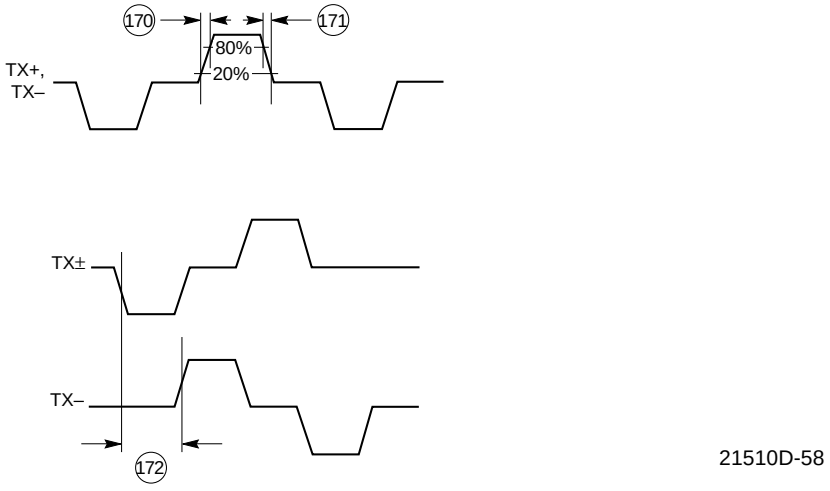
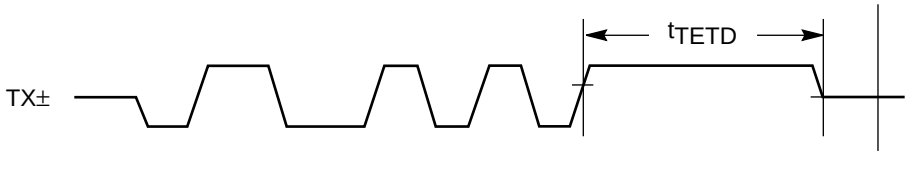


Figure 54 PMD Interface Timing (MLT-3)

10BASE-T

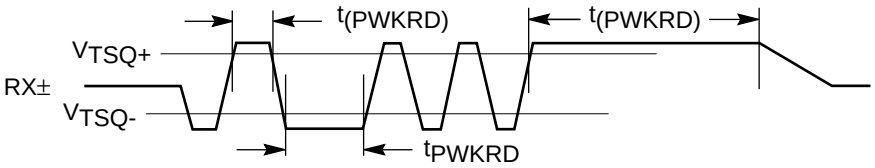
No.	Symbol	Parameter Description	Test Conditions	Min	Max	Unit
	t_{TETD}	Transmit End of Transmission		250	375	ns
	t_{PWKRD}	RX± Pulse Width Maintain/Turn Off Threshold	$\|V_{IN}\| > \|V_{THS}\|$ (Note 1)	136	200	ns

Note: *RX± pulses narrower than t_{PWDRD} (min) will maintain internal Carrier Sense on. RX± pulses wider than t_{PWKRD} (max) will turn internal Carrier Sense off.*



21510D-59

Figure 55 10 Mbps Transmit (TX±) Timing Diagram

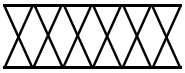
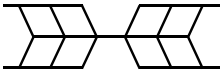


21510D-60

Figure 56 10 Mbps Receive (RX±) Timing Diagram

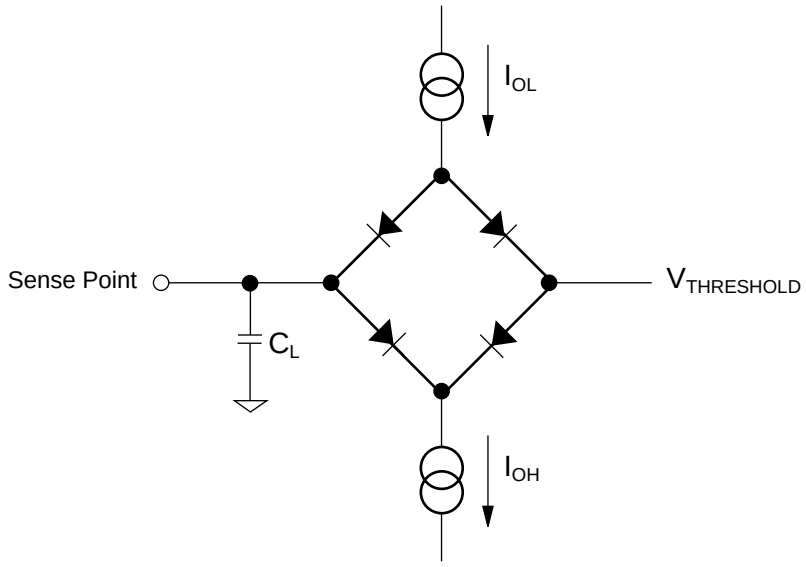
SWITCHING WAVEFORMS

Key to Switching Waveforms

WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don’t Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance “Off” State

KS000010-PAL

SWITCHING TEST CIRCUITS



21510D-61

Figure 57 Normal and Tri-State Outputs

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE

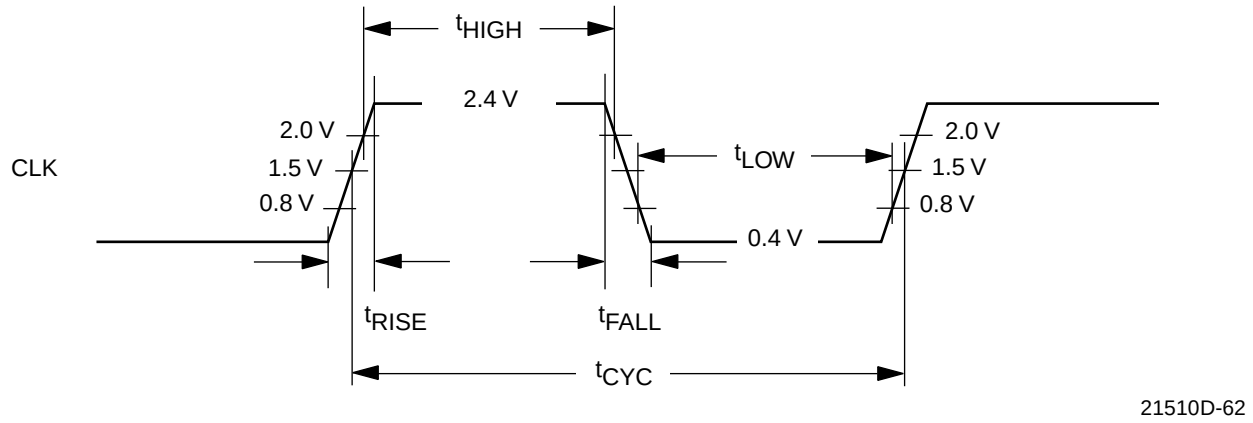


Figure 58 CLK Waveform for 5 V Signaling

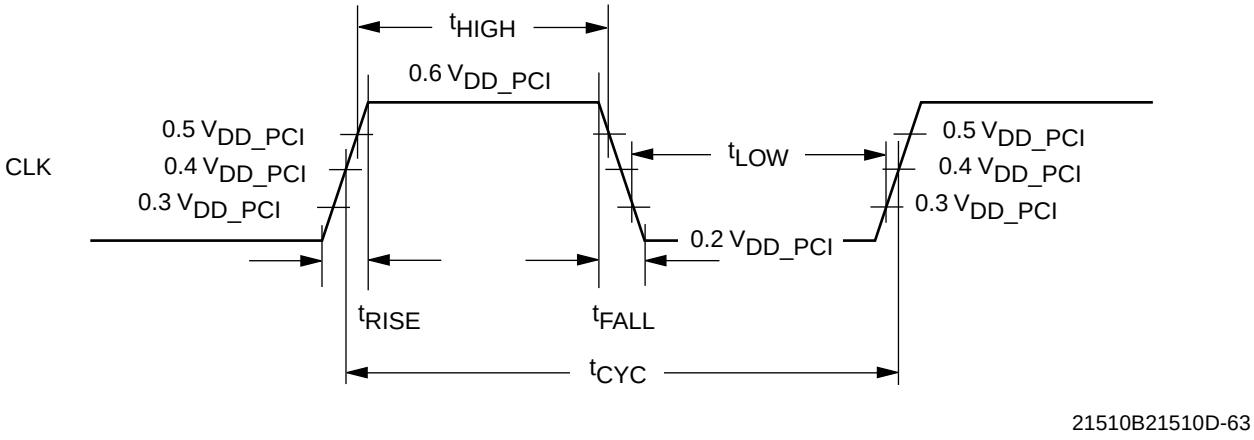


Figure 59 CLK Waveform for 3.3 V Signaling

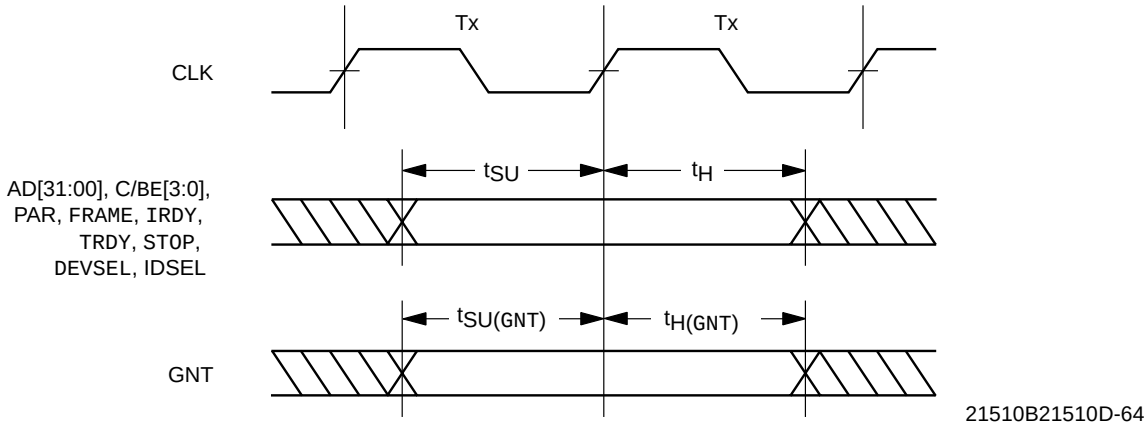


Figure 60 Input Setup and Hold Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE (Continued)

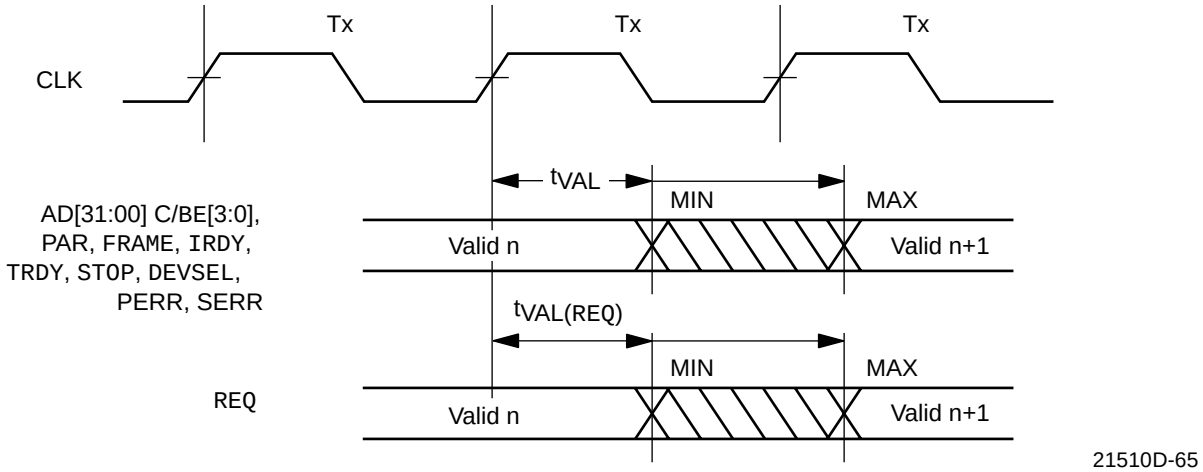


Figure 61 Output Valid Delay Timing

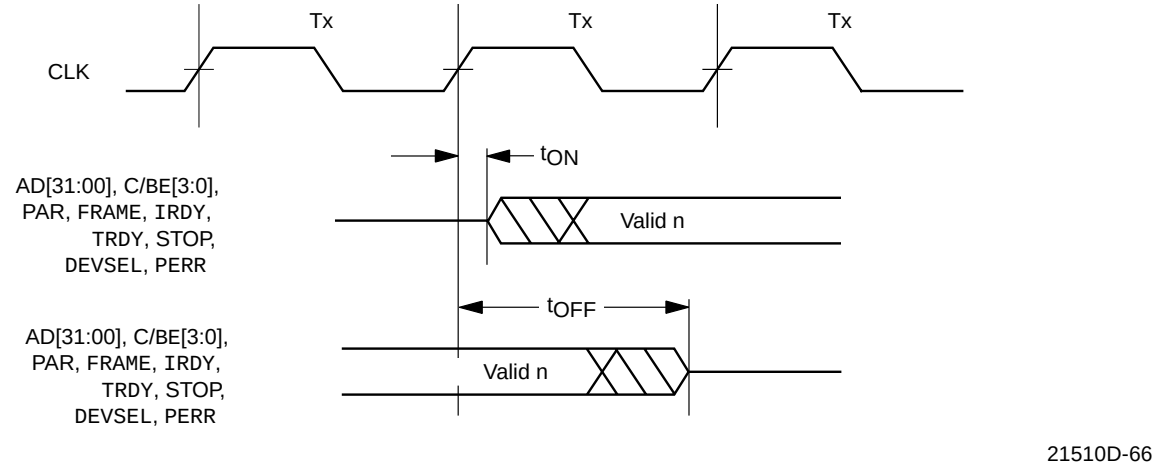


Figure 62 Output Tri-state Delay Timing

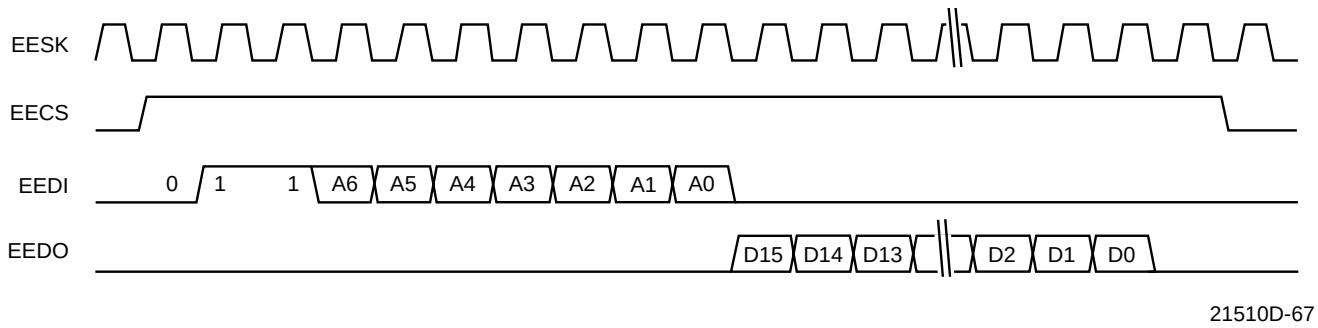


Figure 63 EEPROM Read Functional Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE (Continued)

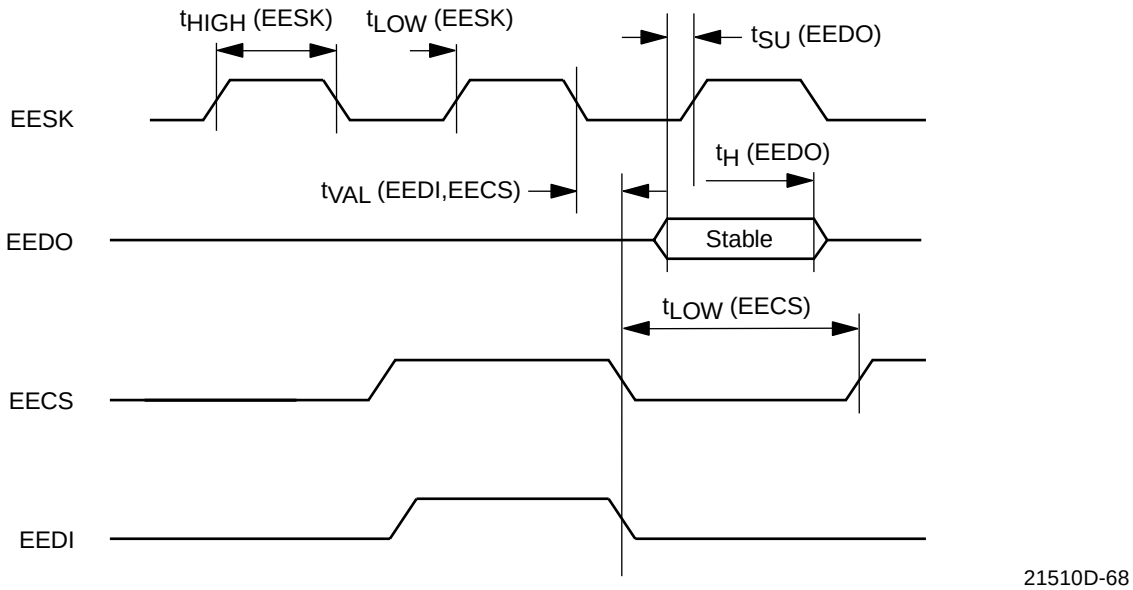


Figure 64 Automatic PREAD EEPROM Timing

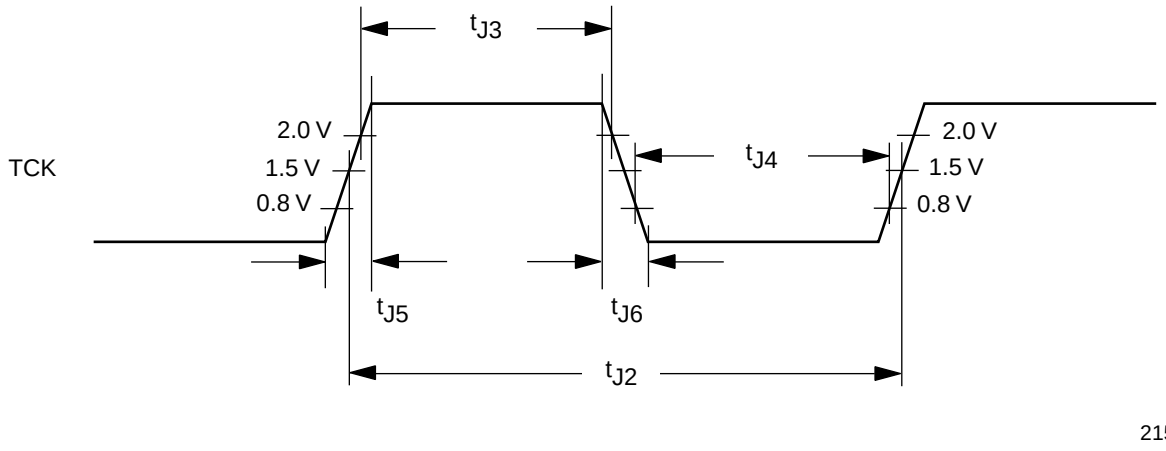


Figure 65 JTAG (IEEE 1149.1) TCK Waveform for 5 V Signaling

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE (Concluded)

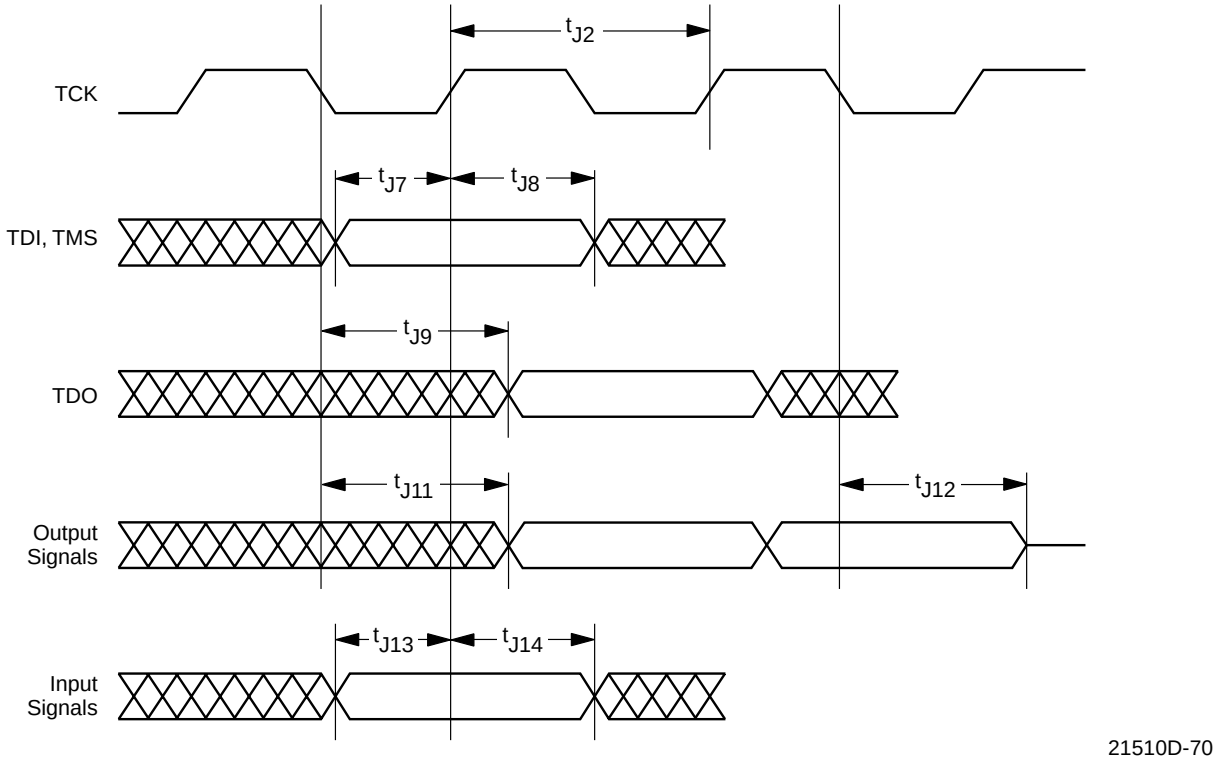


Figure 66 JTAG (IEEE 1149.1) Test Signal Timing

SWITCHING WAVEFORMS: EXPANSION BUS INTERFACE

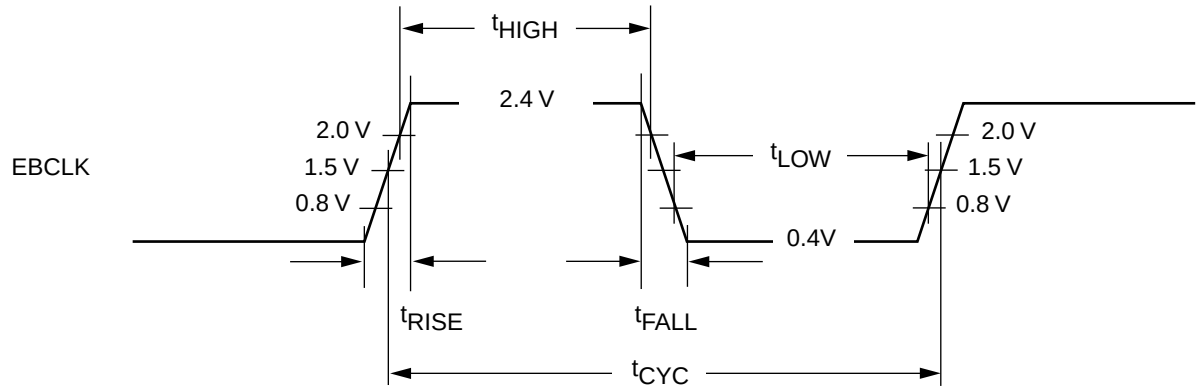


Figure 67 EBCLK Waveform

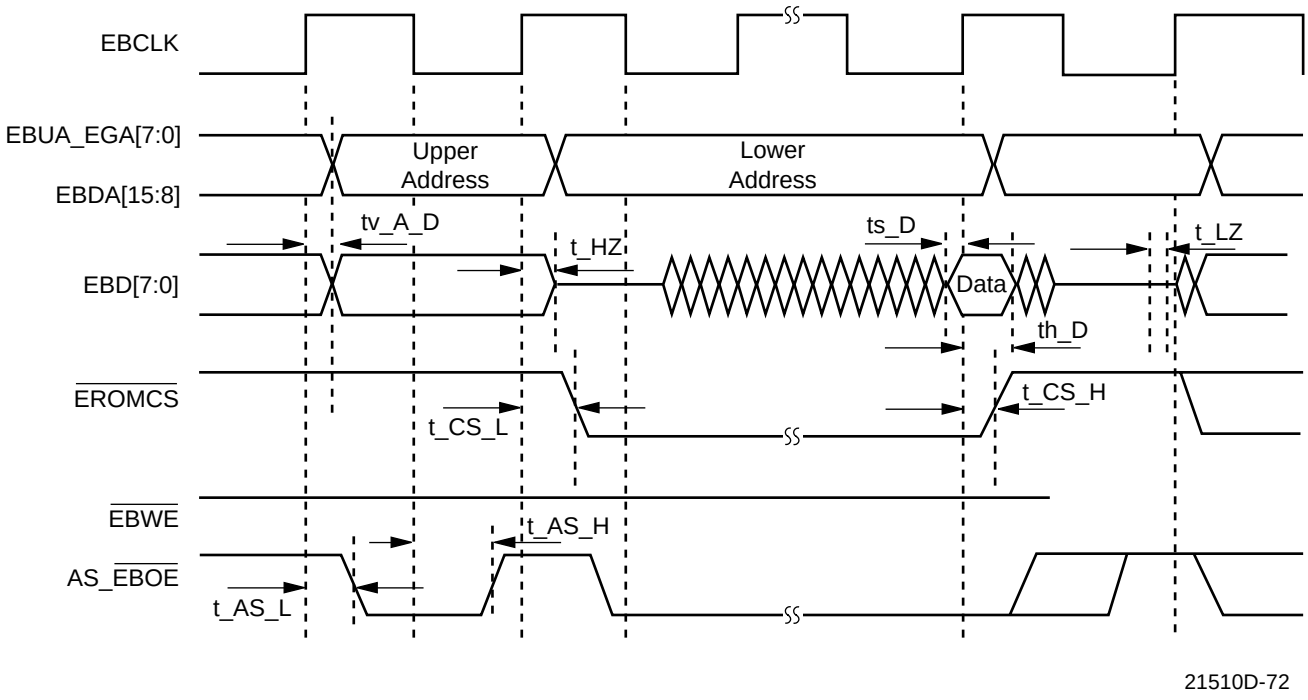


Figure 68 Expansion Bus Read Timing

SWITCHING WAVEFORMS: EXPANSION BUS INTERFACE (Concluded)

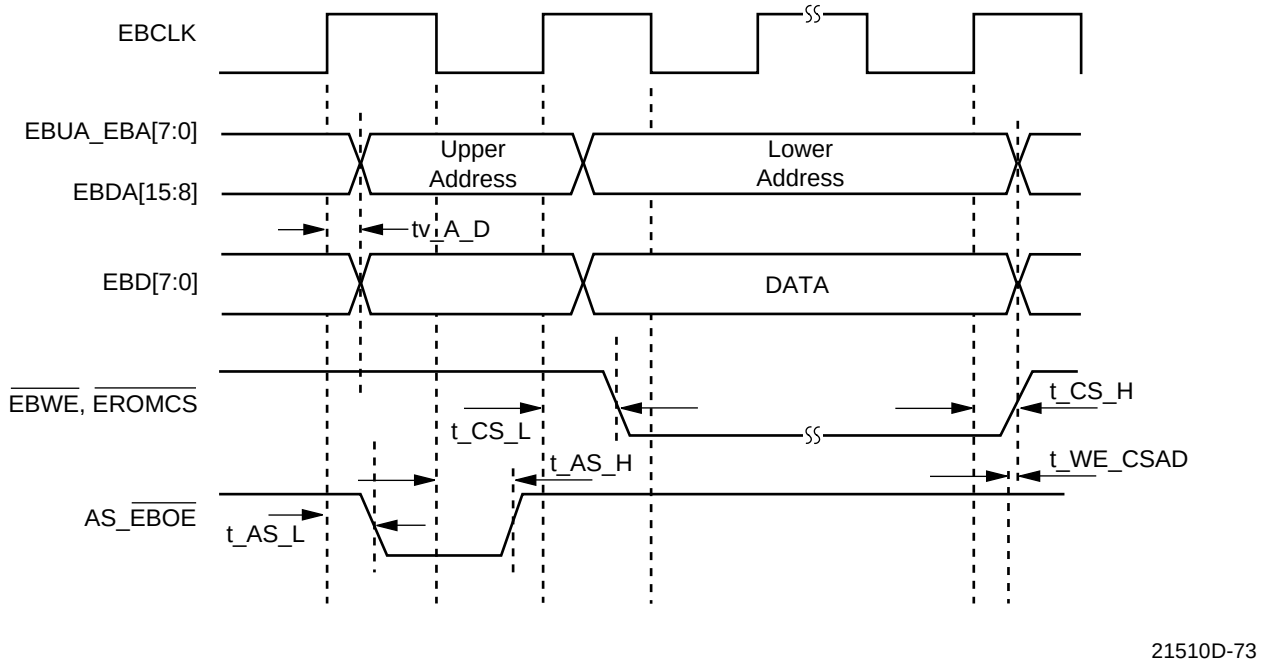
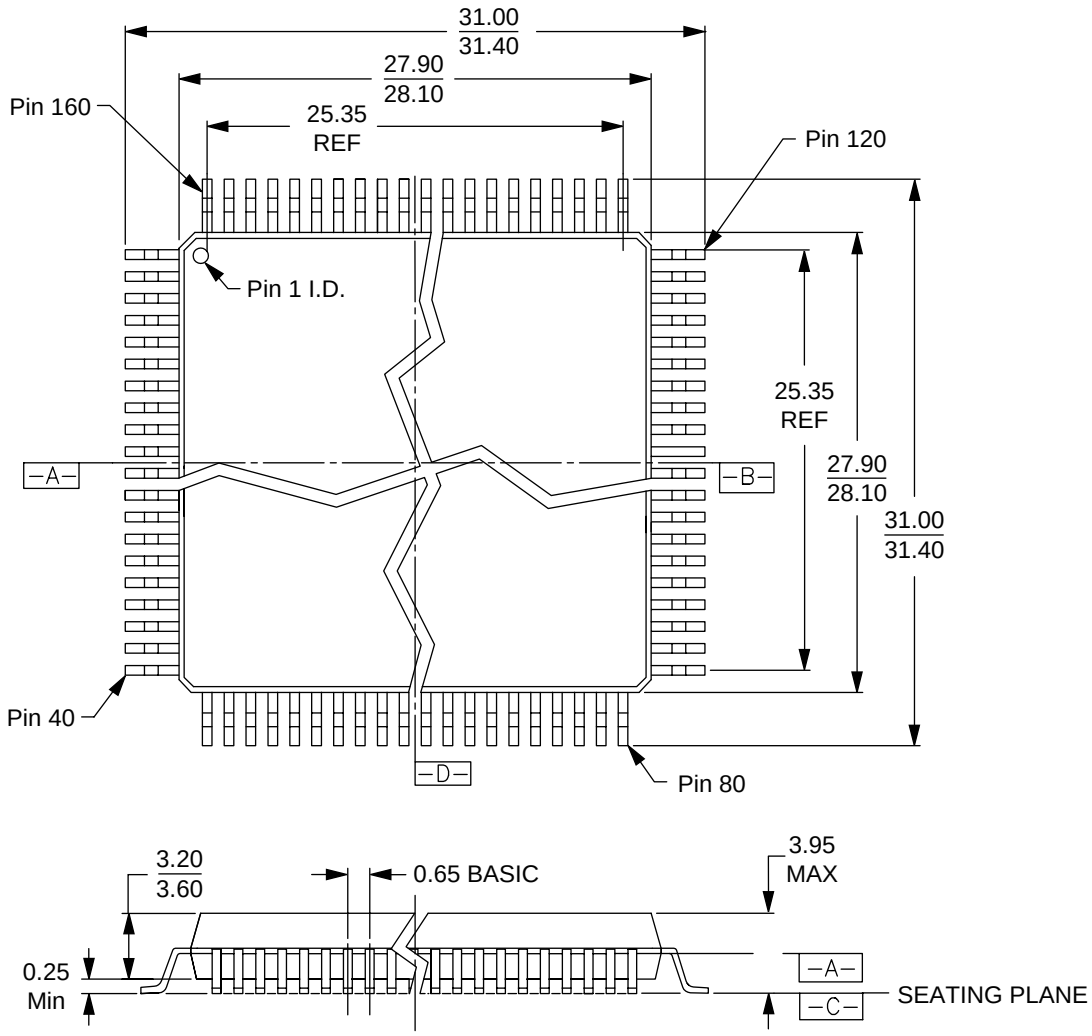


Figure 69 Expansion Bus Write Timing

PHYSICAL DIMENSIONS*

PQR160

Plastic Quad Flat Pack (measured in millimeters)

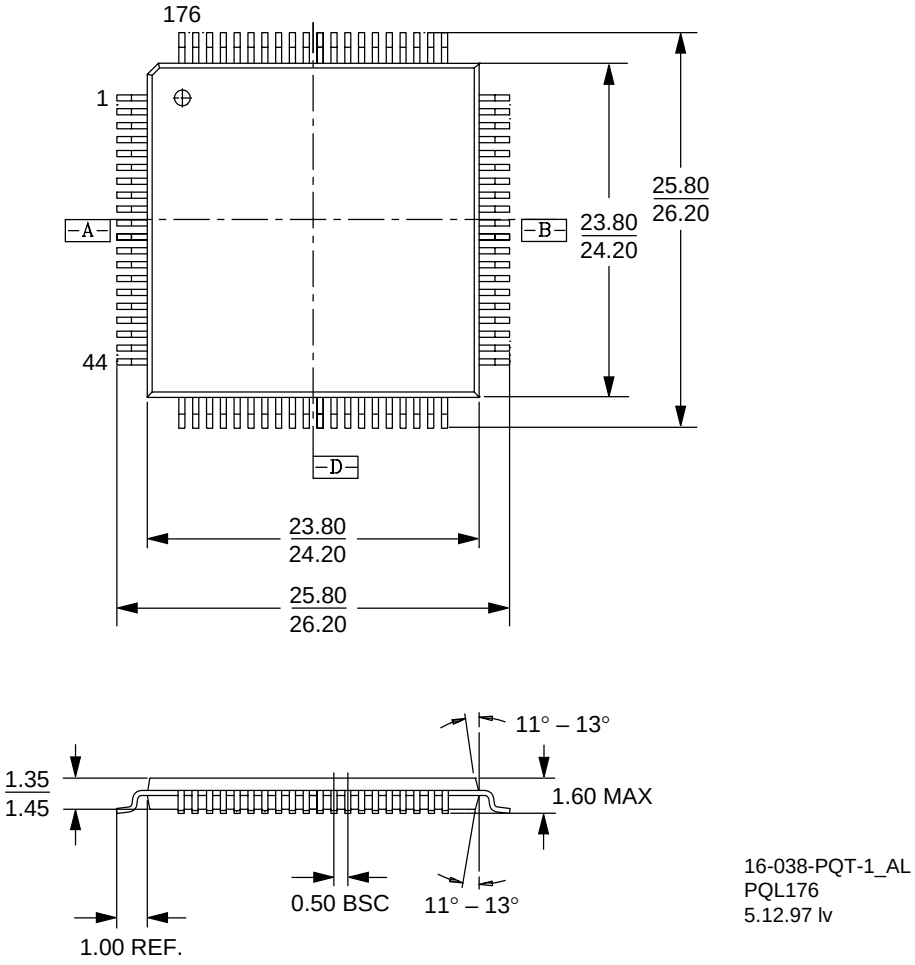


16-038-PQR-1
PQR160
12-22-95 lv

**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

21510E

PCnet™-*FAST III* Recommended Magnetics

APPENDIX A: PCnet™-*FAST III* Recommended Magnetics

The PCnet-*FAST III* controller uses magnetics that have a Transmit (TX) turns ratio of 1:1.414 and a Receive (RX) turns ratio of 1:1. The following table shows the current approved vendor list of 3.3 V magnetics recommended for use with the PCnet-FAST III device. These magnetics modules are pin compatible with the AMD Am79C873 NetPHY-1 10/100 PHY magnetics and LevelOne LXT970 10/100 PHY magnetics that are used today with PCnet-*FAST* and PCnet-*FAST*+, but are *not* the same part numbers since the turns ratios are different.

Table 67. Recommended Magnetics Vendors

Vendor	Part Number	Package
Halo	TG22-SI43ND	16-pin module
Halo	TG22-SI41N2	16-pin module
Halo	TG110-SI41N2	16-pin module
Bel Fuse	S558-5999-G9	16-pin module
Bel Fuse	S558-5999-G8	16-pin module
Pulse Engineering	H1081	16-pin module
Pulse Engineering	H1119	16-pin module
Bothhand	16ST61A8	16-pin module
PCA Electronics	EPF8095G	16-pin module
PCA Electronics	EPF8096G	16-pin module
Transpower Technologies	RJ622-CL1	RJ45/Magnetics Combo
XFMRS	XF973-COMBO1-4	RJ45/Magnetics Combo

AMD

Serial Management Interface Unit (Am79C975 only)

APPENDIX B: Serial Management Interface Unit (Am79C975 only)

Related Documents:

- System Management Bus Specification Revision 1.0, February 15, 1995
- Phillips Semiconductors: The I^2C-bus and how to use it (including specifications), April 1995

Overview

The Am79C973 and Am79C975 devices are fully-integrated 32-bit PCI bus 10/100 Mbps Ethernet controllers with advanced power and network management features. With the Serial Management Interface, the Am79C975 device offers a very powerful system management feature in addition to the management features offered with the Am79C973 device.

The Serial Management Interface Unit (SMIU) is based on the industry standard Inter-IC (I^2C) bus and System Management Bus (SMBus) specifications. It enables a system to exchange short message with another network station (e.g., management console) for remote monitoring and alerting of system management parameters and events. The SMIU is capable of communicating within the system and over the network during normal operation or in low power mode, even if the Am79C975 controller is not initialized or setup for transmit or receive operation by a network driver.

One application for the Serial Management Interface is a system where a dedicated microcontroller monitors various hardware components (e.g., fan, memory, chip set) and parameters (e.g., temperature, voltage) in order to obtain information about the status of the system. If there is any problem or issue with the system, the microcontroller can send a message via the serial management interface of the Am79C975 Am79C975 controller to another station on the network, for example, to alert the system administrator of the trouble event.

A less robust (but more cost effective) implementation in a typical PC system simply uses the host processor to control operation of the I^2C/SMBus interfaces of the PCI chipset (southbridge) and hardware monitoring device, and the Serial Management Interface of the Am79C975 controller.

The independent interface allows access to the network even at times when there is no operating system and network driver running on the system. The interface, however, also allows access to the network in parallel to the normal traffic.

The electrical interface of the SMIU is comprised of 3 pins: clock, data, and interrupt. The host can access a set of registers via the interface to identify the Am79C975 controller, to obtain information about the status of the device, to get the network address of the local node and the management station of the network, and to control the transmission and reception of management frames. The Am79C975 controller provides internal transmit and receive data memories of 128 bytes each to store the management frames. The receive path includes a pattern match filter to qualify incoming frames.

The SMIU is designed such that the amount of software running on the microcontroller in order to transmit or receive management frames is minimized. The SMIU does not use the PCI clock (CLK). It will operate with CLK stopped or running.

The electrical interface of the SMIU follows the I^2C specification. The signaling and register access protocol of the SMIU is a subset of the System Management Bus (SMBus) protocol.

Am79C975 PIN DESIGNATIONS

Listed by Pin Number

160-pin PQFP package		176-pin TQFP package	
Pin No	**Pin Name**	**Pin No**	**Pin Name**
96	$\overline{\text{MIRQ}}$	106	$\overline{\text{MIRQ}}$
98	MDATA	108	MDATA
100	MCLOCK	110	MCLOCK

Listed By Group

Serial Management Interface Unit (SMIU)				
Pin Name	**Pin Function**	**Type**	**Driver**	**No. of Pins**
MCLOCK	SMIU Clock	I/O	OD6	1
MDATA	SMIU Data	I/O	OD6	1
$\overline{\text{MIRQ}}$	SMIU Interrupt	O	OD6	1

Note: *OD6 = Open Drain Output, I_{OL} = 6 mA, 50 pF load.*

Listed By Function

MCLOCK

SMIU Clock **Input/Output**

MCLOCK is the clock pin of the serial management interface. MCLOCK is typically driven by an external master (e.g., the southbridge or a dedicated microcontroller). The Am79C975 controller will drive the clock line low in order to insert wait states before it starts sending out data in response to a read. The frequency of the clock signal can vary between 10 KHz and 100 KHz and it can change from cycle to cycle.

Note: *MCLOCK is capable of running at a frequency as high as 1.25 MHz to allow for shorter production test time.*

MDATA

SMIU Data **Input/Output**

MDATA is the data pin of the serial management interface. MDATA can be driven by an external master (e.g., the microcontroller or southbridge) or by the Am79C975 controller. The interface protocol defines exactly at what time Am79C975 has to listen to the MDATA pin and at what time the controller must drive the pin (see section Basic Operations for more details).

MIRQ

SMIU Interrupt **Output**

$\overline{\text{MIRQ}}$ is an asynchronous attention signal that the Am79C975 controller provides to indicate that a management frame has been transmitted or received. The assertion of the $\overline{\text{MIRQ}}$ signal can be controlled by a global mask bit (MIRQEN) or individual mask bits (MRX_DONEM, MTX_DONEM), located in the Command register. Note that the SMIU interrupt acknowledge does not follow the SMB alert protocol, but simply requires clearing the interrupt bit.

Basic Operation

Transferring Data

The Serial Management Interface Unit (SMIU) of the Am79C975 controller uses a two pin interface to communicate with other devices. MCLOCK is the clock pin. MDATA is the data pin. Both signals are bussed and shared with other devices in the system. There is at least one master device in the system (e.g., a microcontroller). A master is a device that initiates a transfer and also provides the clock signal. The Am79C975 controller is always a slave device.

The master starts a data transfer on the serial management bus by asserting the START condition. The START condition is defined as a HIGH to LOW transition on MDATA while MCLOCK is HIGH. Data will follow with the most significant bit (MSB) of a byte transferred first. Data can only change during the LOW period of MCLOCK and must be stable on the MDATA pin during the HIGH period of MCLOCK. Every byte of data must be acknowledged by the receiving device with the Acknowledge bit (ACK). ACK is defined as a LOW pulse on MDATA that follows the same timing as a regular data bit. In a write operation, the Am79C975 controller is the receiving device and it must generate ACK. In a read operation, the master is the receiving device and it must generate ACK. An inverted Acknowledge bit (NACK) is used to signal the transmitting device that the data transfer should terminate. A data transfer is ended when the master asserts the STOP condition. The STOP condition is defined as a LOW to HIGH transition on MDATA while MCLOCK is HIGH.

Implementation note: The assertion of START forces the state machine in the decoder logic to look for the slave address of the Am79C975 device. The assertion of STOP forces the state machine to reset and to wait for the assertion of a START condition.

The first byte in every data transfer is the 7-bit address of the Am79C975 device followed by the Read/Write bit. The MSB of the 7-bit address is the first bit on the MDATA line. A 0 in the Read/Write bit indicates a write operation from the master to the Am79C975 controller, a 1 indicates a read operation. The Am79C975 controller does not support the General Call address (00h). It will ignore the address by not asserting ACK.

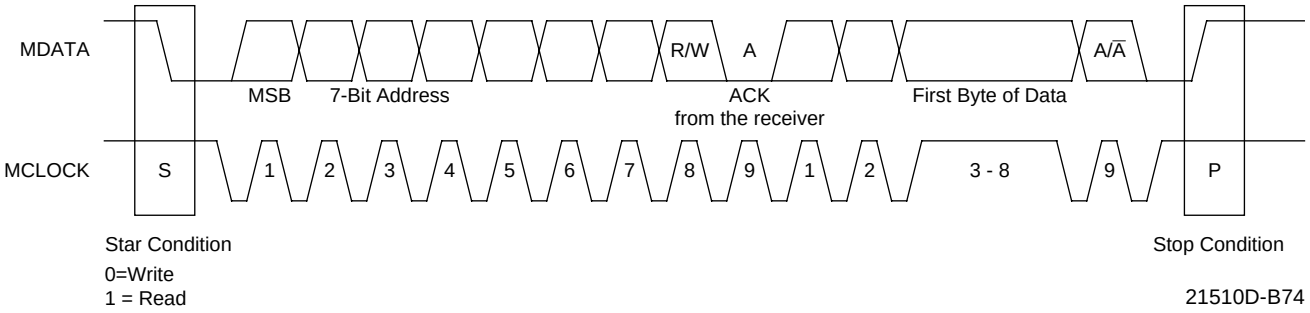


Figure 70. Standard Data Transfer on the Serial Management Interface

A data transfer that involves a change in the direction data is transferred is a more complex operation. An example is a data write transfer followed by a data read transfer. After finishing the data write transfer, the master must initiate the turn-around of the MDATA line by asserting a repeated START condition followed by a repeated 7-bit slave address and the READ bit. The Am79C975 controller will drive the MCLOCK line low in order to insert wait states before it starts driving the read data onto the MDATA pin., The master acts as the receiver and must generate ACK. (See section *Byte Read Command or Block Read Command* for more details).

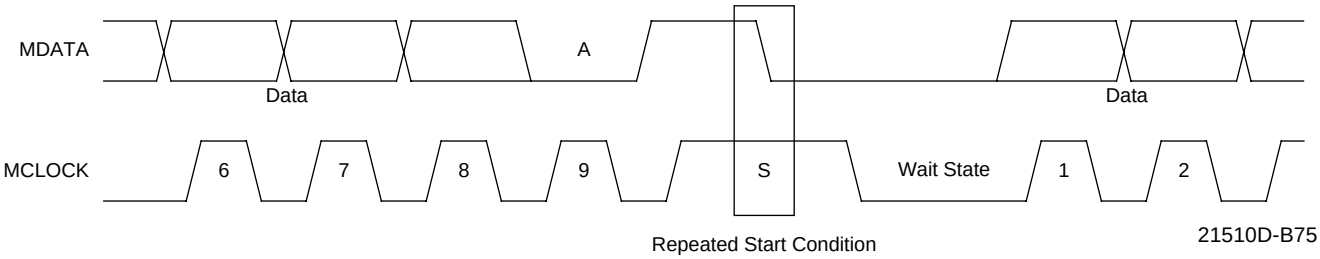


Figure 71. Data Transfer with Change in Direction (with wait state)

Am79C975 Slave Address

The default value for the 7-bit slave address of the Am79C975 SMIU is 5Bh. This is the address assigned for a device from the class Networking Controller by Phillips. The address value can be changed via the EEPROM. If bit 7 of EEPROM location 50h is set to 0, the default slave address will be used. Otherwise, bits 6-0 of EEPROM location 50h define the slave address for the SMIU. If a system uses multiple Am79C975 controllers, recommended values for the slave address of the other devices are 58h, 59h and 5Ah.

7	6-0	
D	Slave Address	D=0: Default address will be used as slave address D=1: Bits 6-0 define the slave address

Register Access

The I^2C specification allows for an unlimited number of bytes being transferred per data transfer. The SMB specification defines a set of commands that structure the data transfers and limit their length. The SMIU of the Am79C975 controller follows the SMB specification and implements a subset of the commands to allow access to the SMIU registers and internal memories.

Write Byte Command

The Write Byte command is used to write 1 byte of data to an SMIU register. The command starts with the START condition (S). The next 7 bits are the slave address of the Am79C975 controller, followed by a 0 bit to indicate a write operation (W). The Am79C975 controller will drive SDATA LOW for 1 bit time to acknowledge the first byte (A). The next 8 bits specify the address of the SMIU register (MReg) that is accessed, followed by another ACK from the Am79C975 controller. The third byte is the write data to the SMIU register, followed by ACK. The master terminates the transfer with the assertion of the STOP condition (P).

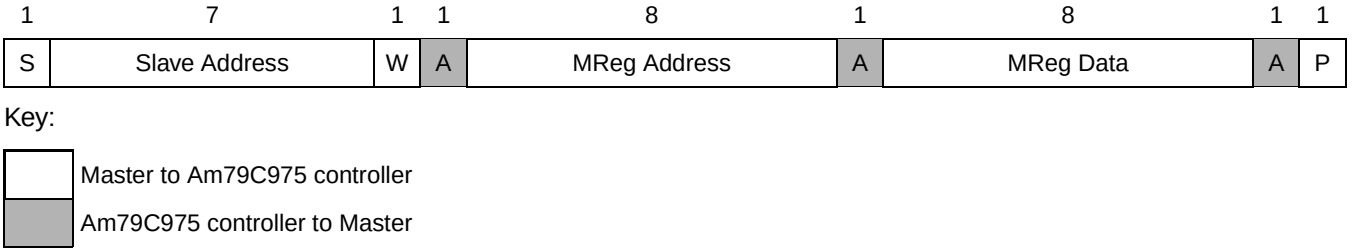


Figure 72. Write Byte Command

Note that the Am79C975 controller does not validate the register address specified in the Write Byte command. A Write Byte command to a non-existing register, or to register that is read-only, or to one of the registers that require a Block Read or Write command may cause unexpected reprogramming of an SMIU register.

Read Byte Command

The Read Byte command is used to read 1 byte of data from an SMIU register. This command is more complex compared to the Write Byte command, since it involves a change in the direction of the data transfer. The command starts with the START condition (S). The next 7 bits are the slave address of the Am79C975 controller, followed by a 0 bit to indicate that the data transfer starts with a write operation from the master to the Am79C975 controller (W). The Am79C975 controller must acknowledge the first byte by driving MDATA LOW for 1 bit time (A). The next byte specifies the address of the SMIU register (MReg) that is accessed, followed by another ACK from the Am79C975 controller. The master initiates the turn-around of the transfer direction by asserting a repeated START condition, followed by the repeated 7-bit slave address. This time, the Read/Write bit is set to 1 to indicate that the next byte of data is driven by the Am79C975 controller (R). The Am79C975 controller acknowledges the transfer and then drives the one byte of register data onto the MDATA line. The acknowledge for the register data is generated by the master, since he is the receiver of the data. The master generates a NACK (N) to force the Am79C975 controller to stop driving the MDATA line since the data transfer consists of only one byte. The Read Byte command terminates with the assertion of the STOP condition (P) by the master.

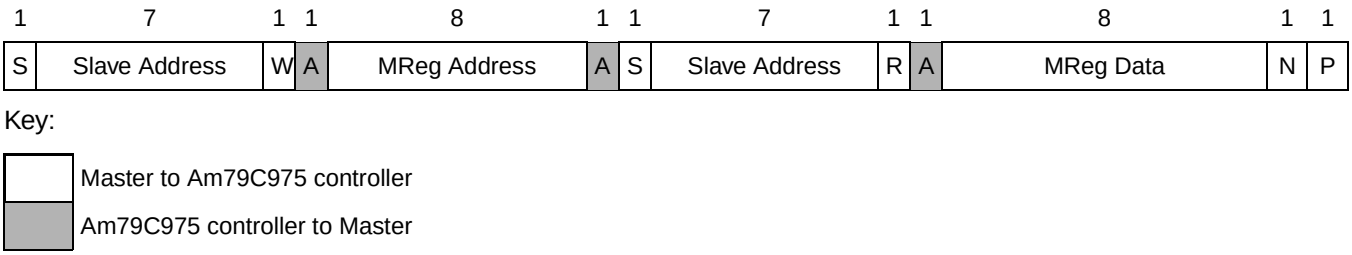


Figure 73. Read Byte Command

Note: *The Am79C975 controller does not validate the register address specified in the Read Byte command. A Read Byte command to a non-existing register, or to register that is write-only, or to one of the registers that require a Block Read or Write command will yield undefined data.*

If the second slave address in the Read Byte Command is not the one of the Am79C975 controller, the device will release the MDATA line to generate a NACK. The master, receiving the NACK, must abort the cycle by generating a STOP condition.

Block Write Command

The Block Write command is used to write data to the SMIU Transmit Data memory or to the Receive Pattern RAM. The command starts with the START condition (S). The next 7 bits are the slave address of the Am79C975 controller, followed by a 0 bit to indicate a write operation (W). The next byte specifies the address of the SMIU register (MReg) that is accessed. The address of the Receive Pattern RAM Data port (34) or the address of the Transmit Data port (36) is the only valid values for the MReg address. The third byte

of the Block Write command specifies the byte count. The byte count value is ignored by the Am79C975 controller. The device is capable of receiving any amount of data even passed the limit of 32 bytes as defined by the SMB specification. Since the Am79C975 controller is the receiver of all data transfers, it must acknowledge each byte by driving the MDATA line LOW for 1 bit time (A). The master indicates the termination of the Block Write command with the assertion of the STOP condition (P).

Note that the Am79C975 controller does not validate the register address specified in the Block Write command. A Block Write command to a register other than the Transmit Data port or the Receive Pattern RAM Data port may cause unexpected reprogramming of an SMIU register. The Am79C975 controller also does not check the length of the Block Write command. The master must make sure that the Transmit Data memory or the Receive Pattern RAM are not written beyond their respective length.

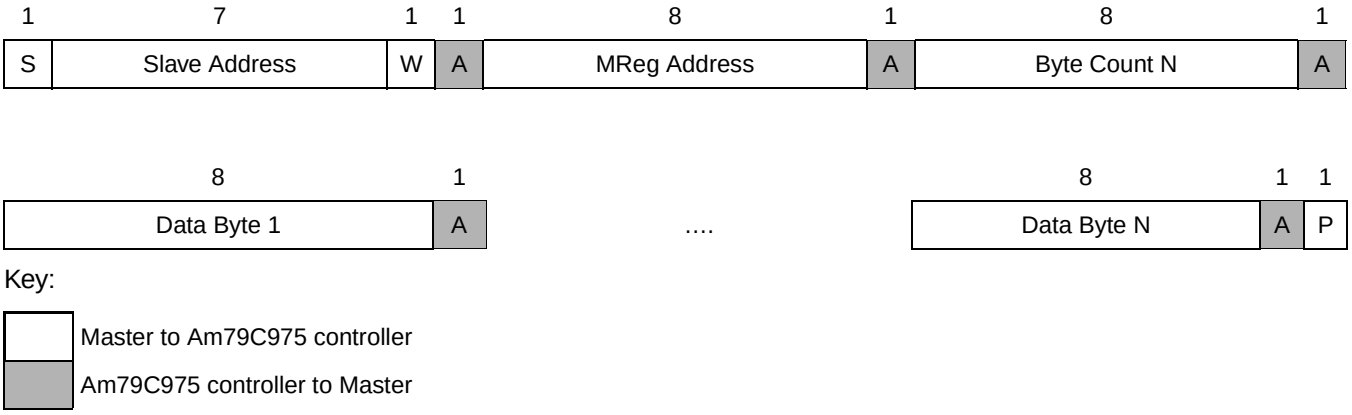


Figure 74. Block Write Command

Block Read Command

The Block Read command is used to read data from the SMIU Receive Data memory. This command is more complex compared to the Block Write command, since it involves a change in the direction of the data transfer. The command starts with the START condition (S). The next 7 bits are the slave address of the Am79C975 controller, followed by a 0 bit to indicate that the data transfer starts with a write operation from the master to the Am79C975 controller (W). The Am79C975 controller must acknowledge the first byte by driving MDATA LOW for 1 bit time (A). The next byte specifies the address of the SMIU register (MReg) that is accessed, followed by another ACK from the Am79C975 controller. The address of the Receive Data port (40), is the only valid value for the MReg address. The master initiates the turn-around of the transfer direction by asserting a repeated START condition, followed by the repeated 7-bit slave address. This time, the Read/Write bit is set to 1 to indicate that the next bytes of data are driven by the Am79C975 controller (R). The Am79C975 controller acknowledges the transfer and then continues driving the MDATA line. The first byte is the byte count indicating how many bytes of data will follow. The byte count field will indicate 32 in all but the last transaction, in which the byte count field will indicate the remaining bytes of the frame. This time, the ACK is generated by the master, since he is the receiver of the data. Receive data will follow. Being the receiver, the master must ACK each byte by driving the MDATA line LOW for 1 bit time. The last byte, however, must be followed by a NACK (N) to force the Am79C975 controller to stop driving the MDATA line. The Am79C975 device is capable of a block read past the 32 byte limit that is indicated in the byte count field. If the host does not assert NACK after the 32nd byte, the Am79C975 controller will continue driving receive data onto the MDATA line until the host asserts NACK. If the master runs out of storage space for the incoming data, he can abort the data transfer after any byte by asserting NACK. The Block Read command terminates with the assertion of the STOP condition (P) by the master.

Note: *The Am79C975 controller does not validate the register address specified in the Block Read command. A Block Read command to a register other than the Receive Data port will yield unexpected data. The master must also make sure to only issue a Block Read command when there is data left in the Receive Data memory.*

If the second slave address in the Block Read Command is not the one of the Am79C975 controller, the device will release the MDATA line to generate a NACK. The master, receiving the NACK, must abort the cycle by generating a STOP condition.

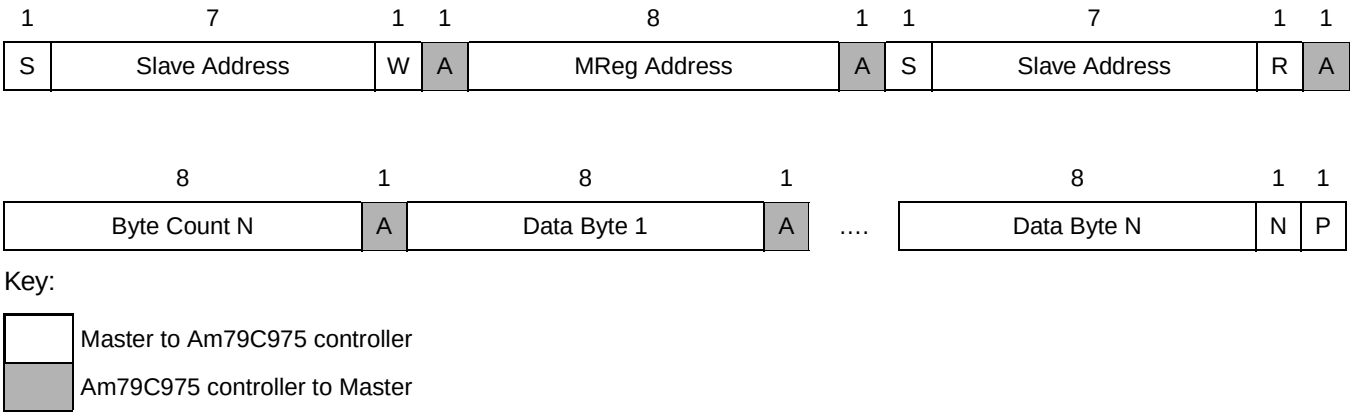


Figure 75. Block Read Command

Detailed Functions

Global Enable/Disable

Bit 15 (SMIUEN) in BCR2 is used to enable/disable the Serial Management Interface Unit in the Am79C975 controller. If SMIUEN is set to 0 (default), the SMIU is disabled. If SMIUEN is set to 1, the SMIU is enabled. BCR2 is programmable via the EEPROM.

Identification

The SMIU of the Am79C975 controller provides a comprehensive set of registers that allow the identification of the device. ID information includes Vendor ID, Device ID and Revision ID. In addition, the Subsystem Vendor ID and Subsystem ID allow a system manufacturer to differentiate his product from a product by another vendor who is also using the Am79C975 controller. All SMIU ID registers are shadow registers of the respective PCI registers and can be initialized via the EEPROM (with the exception of the Revision ID). The host can verify that the registers are correctly initialized from the EEPROM by checking the PVALID bit in the SMIU Am79C975 Status register.

Initialization

The SMIU of the Am79C975 controller does not require any complex initialization in order to transmit or receive management frames. Only the acknowledgment frame filter needs to be setup in order to receive incoming frames (see section Receive Operation later).

The host should check the content of the Transceiver Status register to make sure the Am79C975 controller is connected to a network (LINK set to 1), before any transmit or receive operation is started.

Note that the SMIU is not accessible while the Am79C975 controller is reading the content of the EEPROM after H_RESET (for ~ 1.7 ms). The device will not drive the Acknowledge bit during that time and the access by the master will time-out.

SMIU Bus Frequency

The SMIU operating frequency is set by programming BCR2 register bits. The equation for SMIU bus frequency is $F_{I2C} = 2500/(M+1)*2^{N+1}$ kHz, assuming that a 25 MHz crystal has been used. The maximum value of M is F (Hex) and N is 7 (Hex). With different combinations of M and N, the SMIU bus frequency can vary from 0.5 kHz to 1.25 MHz. It should be noted that the frequencies that can be programmed through BCR2 are not continuous. For example, say F_{I2C} = 10 kHz, then $(M+1)*2^{N+1} = 250$. It is impossible to have such M and N combinations that make an exact 250. With M = E (Hex) and N = 3 (Hex), we get F_{I2C} = 10.41, which will work with a 10 kHz bus frequency. The following is a list of BCR2 contents for various SMIU bus frequencies.

BCR2 (Hex)	F_{I2C} (kHz)
A643	10
A463	13
A642	20
A422	30
A641	40
A461	50
A202	60
A640	80
A460	100

Notes:

1. *Bit 15 and 13 are default 1.*
2. *M = bit (10, 9, 6, 5) and N = bit (4, 1, 0).*
3. *Similarly, other F_{I2C} can be programmed with proper combinations of M and N.*

BCR2 Register Bits for setting SMIU Frequency

BCR2 (Bit No.)	Name	Position
10	I2C_M3	D6
9	I2C_M2	D5
6	I2C_M1	D4
5	I2C_M0	D3
4	I2C_N2	D2
1	I2C_N1	D1
0	I2C_N0	D0

The frequency at which the SMIU will operate is calculated using the following expression: F_{I2C} = 2500/(M+1)*2^{N+1} kHz.

Where: F_{I2C} is the desired frequency (10-100 kHz)

M is the value stored in bits D3-D6

N is the value stored in bits D0-D2.

Status

The SMIU of the Am79C975 controller provides two registers to determine the status of the device.

The Am79C975 Status register indicates the status of the current power state of the device (D0-D3). The power state itself has no affect on the operation of the SMIU. The management interface will operate in any power state, as long as power is provided, i.e. operation in D_{3cold} is not possible.

The Am79C975 Status register also indicates the status of the normal operation of the device. The register contains bits to indicated if the controller is stopped or started and if the transceiver and receiver are enabled.

The Transceiver Status register provides the status of the physical layer interface that is integrated into the Am79C975 controller. The host can read this register to determine if the controller is connected to an active network (LINK set to 1), the speed of the connection (100 Mbps: SPEED=1, 10 Mbps: SPEED=0) and if the connection is half duplex (DUPLEX=0) or full duplex (DUPLEX=1). In most cases, the transceiver will be configured using the autonegotiation process. AUTONEG_DONE will be set to ONE, when autonegotiation with the other end of the network cable has completed. LINK, SPEED and DUPLEX reflect the result of the autonegotiation process. If AUTONEG_DONE is ZERO, the LINK, SPEED and DULPEX bits reflect the status of a manual transceiver configuration.

Transmit Operation

The System Management Interface Unit (SMIU) of the Am79C975 controller provides a separate 128-byte Transmit Data memory to setup a management or alert frame for transmission. The host must load the frame byte by byte using one or multiple Block Write commands to the Transmit Data port. The command code of the Block Write command must be set to 36, the address of the Transmit Data port. The byte count field can have any value since it is ignored by the Am79C975 controller. The device is capable of receiving any amount of transmit data even passed the limit of 32 bytes as defined by the SMB specification. The location within the Transmit Data memory where the next byte is written to is controlled by the SMIU Transmit Address register (MTX_ADR). This register will come up cleared to 0 after H_RESET. With every byte write the address register will auto-increment. This allows a FIFO-type access to the Transmit Data memory and the host does not need to keep track of the location he is writing to. In addition, MTX_ADR can be set to any address within the Transmit Data memory in order to modify a specific location. The host must load all frame information starting from the destination address and ending with the last byte of data. The FCS is automatically appended by the Am79C975 controller. The host must load at least 60 bytes to the Transmit Data memory in order to create a legal length frame. Padding is not supported by the SMIU. The setting of the APAD_XMT bit in CSR4 only effects the transmission of normal frames and not of management frames.

The SMIU provides a set of registers that contain network address information. The address information can be used to setup the alert frame. The six Node IEEE Address registers contain the unique 48-bit address of the station, the Am79C975 controller is used in. The four Node IP Address registers contain the 32-bit IP address of that station. The six Management Station IEEE Address registers contain the unique 48-bit address of the station that is used as the management console in the network. The four Management Station IP Address registers contain the 32-bit IP address of the management console. All 3 sets of registers are loaded from the EEPROM. The Node IP Address, Management IEEE Address and Management IP Address registers can also be used as general purpose registers to load information stored in the EEPROM and make it accessible via the serial management interface to the external host.

The host must setup the Transmit Message Length register (MTX_LEN) with the number of bytes loaded to the Transmit Data memory so that the MAC knows how many bytes to transmit. The host is responsible to load a valid value into the Transmit Message Length register. Any value below 60 will create a runt frame on the

network. Any value above 128 will create unpredictable results. Padding is not supported by the SMIU.

Once the Transmit Data memory is filled with the data of the alert frame and the Transmit Message Length register is setup with the length of the alert frame, the host must set the MTX_START bit in the Transmit Status register to start the transmission. The Am79C975 controller will transmit the alert frame after any pending frame transmission (including retries) has completed. The host can poll the MTX_DONE bit in the Interrupt register to determine if the transmission of the alert frame has already ended. The MTX_DONE bit will be set to a 1 after the end of transmission, independent of the success of the operation. The MTX_DONE bit will auto-clear after reading the Interrupt register.

The MTX_ADR register is cleared by setting the MTX_START bit. The host must not load data for a new alert frame into the Transmit Data memory until the transmission of the current frame is ended as indicated by the MTX_DONE bit. It is possible to issue a new MTX_START command without loading new data to the Transmit Data memory or updating the MTX_LEN register.

The Am79C975 controller provides an asynchronous interrupt pin to signal the host that the transmission of the alert frame is complete. After the end of the transmission, the $\overline{\text{MIRQ}}$ pin will be asserted, if the global interrupt enable bit MIRQEN in the Command register is set to a 1 and the transmit interrupt mask bit (MTX_DONEM) is cleared to 0 (default state). Once $\overline{\text{MIRQ}}$ is asserted, the host can read the MTX_DONE bit in the Interrupt register to determine that the interrupt was caused by the end of the transmission. The read of the Interrupt register will clear the MTX_DONE bit and cause the deassertion of the $\overline{\text{MIRQ}}$ pin. The Interrupt register also provides a global interrupt bit $\overline{\text{MIRQ}}$ that is the OR of the MTX_DONE and MRX_DONE bits.

Once the MTX_DONE bit indicates that the transmission of the alert frame has ended, MTX_START is cleared and the SMIU Transmit Status register provides the error status for the transmission. Three error conditions are reported: Late Collision, Loss of Carrier and Retry Error. There is also an error summary bit (MTX_ERR). The transmit status bits remain valid as long as the MTX_START bit is set to 0.

The Transmit Retry Error condition requires special attention. It can happen, that the START or STOP bit is set in the Am79C975 controller to change normal operation. START and STOP cause a reset of the transmit logic. Normal transmission of an alert frame will succeed. If, however, the SMIU is in the middle of a backoff interval or the transmission of the alert frame is suffering a collision, the transmission will abort and the MTX_RTRY bit will be set. If MTX_RTRY is set in the SMIU Transmit Status register, the host should retransmit the frame.

Receive Operation

The System Management Interface Unit (SMIU) of the Am79C975 controller provides a separate 128-byte Receive Data memory to store an incoming management or acknowledgment frame. The normal receive address matching mechanism (physical address, logical address, broadcast address) of the MAC cannot be used by the SMIU. The Am79C975 controller provides an Acknowledgment Frame Filter instead. The filter is stored in a 40-byte Receive Pattern RAM.

The Receive Pattern RAM is organized as 8 pattern words of 5 bytes each. Each pattern word holds four bytes of data to be compared with the incoming frame plus mask information that indicates which bytes should be included in the comparison. The pattern word also contains a field that indicates how many Dwords of the incoming frame should be skipped before this Dword of pattern is compared with frame data. This field makes it unnecessary to store data for long series of bytes that will be excluded from the comparison anyway. A maximum of 7 Dwords (28 bytes) can be skipped. If the filter pattern contains a string of more than 7 Dwords that must be excluded from the comparison, one or more pattern words will be loaded with the value 7h in the Skip field and 0h in the Mask field. Finally the most significant bit of the pattern word indicates whether or not this word is the end of the stored pattern.

The format of the pattern words is shown below.

Bits	Name	Description
31:0	Pattern	Bytes of data to be compared with the incoming frame. The least significant byte corresponds to the first byte of the Dword received from the network.
35:32	Mask	Bits 3:0 of this field correspond to bytes 3:0 of the pattern field. If bit n of this field is 0, byte n of the pattern is ignored in the comparison.
38:36	Skip	This field indicates how many Dwords of the incoming frame must be skipped before the pattern in this word is compared with data from the incoming frame. A maximum of 7 Dwords may be skipped per pattern word.
39	EOP	End of Pattern. If this bit is set, this pattern word contains the last Dword of the frame filter.

The host must load the Receive Pattern RAM using one or multiple Block Write commands to the Receive Pattern RAM Data port. The command code of the Block Write command must be set to 34, the address of the Receive Pattern RAM Data port. The byte count field can have any value since it is ignored by the Am79C975 controller. The device is capable of receiving any amount of frame filter bytes even passed the limit of 32 bytes as defined by the SMB specification. The location within the Receive Pattern RAM where the next byte is written to is controlled by the Receive Pattern RAM Address register (MRX_PADR). This register will come up cleared to 0 after H_RESET. With every byte write the address register will auto-increment. This allows a FIFO-type access to the Receive Pattern RAM and the host does not need to keep track of the location he is writing to. In addition, MRX_PADR can be set to any address within the Receive Pattern RAM in order to modify a specific location. The sequence the Receive Pattern RAM must be written is LSB (bits 7:0) of the first pattern word, followed by bits 15:8 of the first pattern word. The first pattern word is followed by the second pattern word. The host need not write all 8 pattern words. The last pattern word must have the EOP bit set, even if the pattern word number 8 is the last one. Words following the EOP bit are ignored. The host must make sure that no more than 40 bytes are written to the Receive Pattern RAM. When MRX_ENABLE is set to 1, the MRX_PADR register will clear to 0 and no write to the Receive Pattern RAM may occur. A new acknowledgment frame filter can be written to the Receive Pattern RAM after disabling the receiver by setting MRX_ENABLE to 0.

The table below shows how a sample pattern would be stored in the Receive Pattern RAM. Note that in the 4 columns containing the frame data, the byte in the leftmost column is closer to the start of the frame than the byte in the rightmost column. The 4 columns showing pattern data stored in the RAM show the least significant byte of the RAM word in the rightmost column.

Frame Offset	Offset + 0	Offset + 1	Offset + 2	Offset + 3	RAM Word	EOP	Skip	Mask	Pattern			
0	00	00	1A	00	1	0	0	1111	00	1A	00	00
4	E0	1B	xx	xx	2	0	0	0011	xx	xx	1B	E0
8	xx	xx	xx	xx								
12	08	06	xx	xx	3	0	1	0011	xx	xx	06	08
16	xx	xx	xx	xx								
20	xx	01	xx	xx	4	0	1	0010	xx	xx	01	xx
24	xx	xx	xx	xx								
28	xx	xx	xx	xx								
32	xx	xx	xx	xx								
36	xx	xx	9d	37	5	0	3	1100	37	9d	xx	xx
40	c7	48	xx	xx	6	1	0	0011	xx	xx	48	c7

Once the acknowledgment frame filter has been loaded to the Receive Pattern RAM, the host must set the MRX_ENABLE bit in the Receive Status register to enable the reception of acknowledgment frames.

When an incoming frame passes the acknowledgment frame filter, the Am79C975 controller will store the frame data in the Receive Data memory. Runt frames (frames shorter than 64 bytes) are automatically deleted, unless MRX_RPA (bit 2 in the SMIU Command register) is set to a 1. If the incoming frame is larger than 128 bytes (including FCS), the Am79C975 controller will only store the first 128 bytes and discard the rest. A message length between 129 and 132 bytes indicates that only FCS bytes have been lost and all message data bytes are available in the Receive Data memory. A message length greater than 132 indicates that message data bytes have been lost from the end of the message. The Receive Message Length register (MRX_LEN) will indicate the correct message length up to 255 bytes. It will freeze at 255 for longer frames. Pad stripping is not supported by the SMIU. The setting of the ASTRP_RCV bit in CSR4 only effects the reception of normal frames and not of acknowledgment frames. Note that if the main receiver of the Am79C975 device is enabled, the acknowledgment frame will also be passed up to the network driver and from there to the protocol stack. The frame format of the acknowledgment frame should be designed such that it will be identified by the protocol stack as a special frame and thrown away. There will be no real performance impact

since the frequency of the acknowledgment frames is very low.

Once the receive activity has ended, the Am79C975 controller will set the MRX_DONE bit in the Interrupt register and clear the MRX_ENABLE bit in the Receive Status register. The MIRQ pin will be asserted, if the global interrupt enable bit MIRQEN in the Command register is set to a 1 and the receive interrupt mask bit (MRX_DONEM) is cleared to 0 (default state). Once MIRQ is asserted, the host can read the MRX_DONE bit in the Interrupt register to determine that the interrupt was caused by the end of the reception. The read of the Interrupt register will clear the MRX_DONE bit and cause the deassertion of the MIRQ pin. The Interrupt register also provides a global interrupt bit MIRQ that is the OR of the MRX_DONE and MTX_DONE bits.

Once the MRX_DONE bit indicates that the reception of the acknowledgment frame has ended, the Receive Status register provides the error status for the reception. Two error conditions are reported: Framing Error and Frame Check Sequence Error. There is also an error summary bit (MRX_ERR). The receive status bits remain valid as long as the MRX_ENABLE bit is set to 0. The host has the option to discard an erroneous frame by simply not reading the receive data and unprotecting the Receive Data memory by setting the MRX_ENABLE bit.

The host must read the Receive Message Length register in order to determine the length of the frame. The data from the Receive Data memory is read byte by byte using one or multiple Block Read commands from the Receive Data port. The command code of the Block Read command must be set to 40, the address of the Receive Data port. The Am79C975 controller will indicate in the byte count field the number of bytes that will follow. The byte count field will indicate 32 in all but the last transaction in which the byte count field will indicate the remaining bytes of the frame. The device is capable of transferring data beyond the 32 byte mark. If the host does not assert NACK after the 32nd byte, the Am79C975 controller will continue driving receive data onto the MDATA line until the host asserts NACK. If the master does not have enough buffer space for the incoming data, it can abort the data transfer after any byte. The Am79C975 controller will start the next Block Read command with the remaining data. The location within the Receive Data memory from where the next byte is read is controlled by the Receive Address register. This register will come up cleared to 0 after H_RESET. With every byte read the address register will auto-increment. This allows a FIFO-type access to the Receive Data memory and the host does not need to keep track of the location he is reading from. In addition, MRX_ADR can be set to any address within the Receive Data memory in order to read a specific location or to start the receive data read from an arbitrary address inside the Receive Data memory. Note, that the byte count field in the Block Read command will only reflect the correct amount of transfer data if the access starts at Receive Data memory address 0. If MRX_ADR is manually changed, the byte count field should be ignored. The host must use the Receive Message Length register (MRX_LEN) to determine the length of the data read operation. Data will be undefined, if the host reads further than the Receive Message Length register is indicating.

The Receive Data memory will be protected from overwriting by another frame, until the host enables the next receive by setting the MRX_ENABLE bit the Receive Status register. The operation will also clear the Receive Address register.

Loopback Operation

The SMIU provides a looback mode for diagnostic purposes. If MLOOP in the Command register is set to 1, the receive path is not being blocked while the device is transmitting in half-duplex mode. Receive is never blocked in full-duplex mode and MLOOP has no effect in this mode.

For loopback operation, transmit data must be sent back to the receiver. This is done at the transceiver by either using an external loopback connector or by programming the transceiver for loopback mode. The programming must be done using the Am79C975 CSR/BCR register interface. This limits the SMIU loopback mode to a debug or manufacturing test environment.

User Accessible Registers

The Serial Management Interface Unit (SMIU) of the Am79C975 controller provides four types of user accessible registers: device ID registers, node address registers, device status registers and control and status registers. Most registers are accessible via the Read Byte and/or Write Byte commands. Only the access to the Transmit and Receive Data port as well as to the Receive Pattern RAM Data port is performed as a Block Read or Block Write command. In all commands, the command code is interpreted as the address of the register.

Device ID Registers

The following register allow the unique identification of the Am79C975 device in a system.

SMIU Vendor ID Register 0 (MReg Address 0)

This register is a shadow register of the PCI Vendor ID Register bits 7:0. The PCI Vendor ID Register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	MVENDOR_ID[7:0]

Default: 22h Read only, write has no effect.

These are the lower 8-bits of a register that identifies AMD as the manufacturer of the Am79C975 device.

SMIU Vendor ID Register 1 (MReg Address 1)

This register is a shadow register of the PCI Vendor ID Register bits 15:8. The PCI Vendor ID Register is loaded from the EEPROM.

Bit No. Name and Description

7:0 MVENDOR_ID[15:8]

Default: 10h Read only, write has no effect.

These are the upper 8-bits of a register that identifies AMD as the manufacturer of the Am79C975 device.

SMIU Device ID Register 0 (MReg Address 2)

This register is a shadow register of the PCI Device ID Register bits 7:0.

Bit No. Name and Description

7:0 MDEVICE_ID[7:0]

Default: 00h Read only, write has no effect.

These are the lower 8-bits of a register that identifies the Am79C975 device within AMD's product line.

SMIU Device ID Register 1 (MReg Address 3)

This register is a shadow register of the PCI Device ID Register bits 15:8.

Bit No. Name and Description

7:0 MDEVICE_ID[15:8]

Default: 20h Read only, write has no effect

These are the upper 8-bits of a register that identifies the Am79C975 device within AMD's product line.

SMIU Revision ID Register (MReg Address 4)

This register is a shadow register of the PCI Revision ID Register.

Bit No. Name and Description

7:0 MREVISION_ID

Default: 40h Read only, write has no effect

This 8-bit register specifies the Am79C975 controller revision number. The upper 4 bits of the register value are fixed to 4h, the lower four bits are silicon-revision dependent. The initial revision value will be 40h.

SMIU Subsystem Vendor ID Register 0 (MReg Address 5)

This register is a shadow register of the PCI Subsystem Vendor ID Register bits 7:0. The PCI Subsystem Vendor ID Register is loaded from the EEPROM.

Bit No. Name and Description

7:0 MSUBVEN_ID[7:0]Default: 00hRead only, write has no effect

These are the lower 8-bits of a register that together with the Subsystem ID uniquely identifies the add-in card or subsystem the Am79C975 device is used in.

SMIU Subsystem Vendor ID Register 1 (MReg Address 6)

This register is a shadow register of the PCI Subsystem Vendor ID Register bits 15:8. The PCI Subsystem Vendor ID Register is loaded from the EEPROM.

Bit No. Name and Description

7:0 MSUBVEN_ID[15:8]

Default: 00h Read only, write has no effect

These are the upper 8-bits of a register that together with the Subsystem ID uniquely identifies the add-in card or subsystem the Am79C975 device is used in.

SMIU Subsystem ID Register 0 (MReg Address 7)

This register is a shadow register of the PCI Subsystem ID Register bits 7:0. The PCI Subsystem ID Register is loaded from the EEPROM.

Bit No. Name and Description

7:0 MSUBSYS_ID[7:0]Default: 00hRead only, write has no effect

These are the lower 8-bits of a register that together with the Subsystem Vendor ID uniquely identifies the add-in card or subsystem the Am79C975 device is used in.

SMIU Subsystem ID Register 1 (MReg Address 8)

This register is a shadow register of the PCI Subsystem Vendor ID Register bits 15:8. The PCI Subsystem ID Register is loaded from the EEPROM.

Bit No. Name and Description

7:0 MSUBSYS_ID[15:8]

Default: 00h Read only, write has no effect.

These are the upper 8-bits of a register that together with the Subsystem Vendor ID uniquely identifies the add-in card or subsystem the Am79C975 device is used in.

Node ID Registers

The following registers provide the IEEE and IP addresses of the node using the Am79C975 devices, as well as of the management station.

Node IEEE Address 0 (MReg Address 9)

This register is a shadow register of the APROM location 00h. APROM location 00h is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IEEE_ADR[7:0] Default: 00h Read only, write has no effect. Byte 0 (the LSB) of the Node IEEE Address register. The Node IEEE Address is the unique 48-bit address of the station, the Am79C975 controller is used in.

Node IEEE Address 1 (MReg Address 10)

This register is a shadow register of the APROM location 01h. APROM location 01h is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IEEE_ADR[15:8] Default: 00h Read only, write has no effect. Byte 1 of the Node IEEE Address register. The Node IEEE Address is the unique 48-bit address of the station, the Am79C975 controller is used in.

Node IEEE Address 2 (MReg Address 11)

This register is a shadow register of the APROM location 02h. APROM location 02h is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IEEE_ADR[23:16] Default: 00h Read only, write has no effect. Byte 2 of the Node IEEE Address register. The Node IEEE Address is the unique 48-bit address of the station, the Am79C975 controller is used in.

Node IEEE Address 3 (MReg Address 12)

This register is a shadow register of the APROM location 03h. APROM location 03h is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IEEE_ADR[31:24] Default: 00h Read only, write has no effect. Byte 3 of the Node IEEE Address register. The Node IEEE Address is the unique 48-bit address of the station, the Am79C975 controller is used in.

Node IEEE Address 4 (MReg Address 13)

This register is a shadow register of the APROM location 04h. APROM location 04h is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IEEE_ADR[39:32] Default: 00h Read only, write has no effect. Byte 4 of the Node IEEE Address register. The Node IEEE Address is the unique 48-bit address of the station, the Am79C975 controller is used in.

Node IEEE Address 5 (MReg Address 14)

This register is a shadow register of the APROM location 05h. APROM location 05h is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IEEE_ADR[47:40] Default: 00h Read only, write has no effect. Byte 5 (the MSB) of the Node IEEE Address register. The Node IEEE Address is the unique 48-bit address of the station, the Am79C975 controller is used in.

Node IP Address 0 (MReg Address 15)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IP_ADR[7:0]Default: 00hRead only, write has no effect. Byte 0 (the LSB) of the Node IP Address register. The Node IP Address is the 32-bit address used in the IP protocol header to address the station, the Am79C975 controller is used in. This register can also be used as general pur-

pose 8-bit register that is loaded from the EEPROM.

Node IP Address 1 (MReg Address 16)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IP_ADR[15:8] Default: 00h Read only, write has no effect. Byte 1 of the Node IP Address register. The Node IP Address is the 32-bit address used in the IP protocol header to address the station, the Am79C975 controller is used in. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Node IP Address 2 (MReg Address 17)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IP_ADR[23:16] Default: 00h Read only, write has no effect. Byte 2 of the Node IP Address register. The Node IP Address is the 32-bit address used in the IP protocol header to address the station, the Am79C975 controller is used in. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IP Address 3 (MReg Address 18)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	N_IP_ADR[31:24] Default: 00h Read only, write has no effect. Byte 3 (the MSB) of the Node IP Address register. The Node IP Address is the 32-bit address used in the IP protocol header to address the station, the Am79C975 controller is used in. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IEEE Address 0 (MReg Address 19)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IEEE_ADR[7:0]Default: 00hRead only, write has no effect. Byte 0 (the LSB) of the Management Station IEEE Address register. The Management Station IEEE Address is the unique 48-bit address of the station that is used as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IEEE Address 1 (MReg Address 20)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IEEE_ADR[15:8] Default: 00h Read only, write has no effect. Byte 1 of the Management Station IEEE Address register. The Management Station IEEE Address is the unique 48-bit address of the station that is used as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IEEE Address 2 (MReg Address 21)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IEEE_ADR[23:16] Default: 00h Read only, write has no effect. Byte 2 of the Management Station IEEE Address register. The Management Station IEEE Address is the unique 48-bit address of the station that is used as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IEEE Address 3 (MReg Address 22)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IEEE_ADR[31:24] Default: 00h Read only, write has no effect. Byte 3 of the Management Station IEEE Address register. The Management Station IEEE Address is the unique 48-bit address of the station that is used as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IEEE Address 4 (MReg Address 23)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IEEE_ADR[39:32]

Default: 00h Read only, write has no effect.

Byte 4 of the Management Station IEEE Address register. The Management Station IEEE Address is the unique 48-bit address of the station that is used as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IEEE Address 5 (MReg Address 24)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IEEE_ADR[47:40]

Default: 00h Read only, write has no effect.

Byte 5 (the MSB) of the Management Station IEEE Address register. The Management Station IEEE Address is the unique 48-bit address of the station that is used as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IP Address 0 (MReg Address 25)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IP_ADR[7:0]

Default: 00h Read only, write has no effect.

Byte 0 (the LSB) of the Management Station IP Address register. The Management Station IP Address is the 32-bit address used in the IP protocol header to address the station that functions as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IP Address 1 (MReg Address 26)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IP_ADR[15:8]

Default: 00h Read only, write has no effect.

Byte 1 of the Management Station IP Address register. The Management Station IP Address is the 32-bit address used in the IP protocol header to address the station that functions as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IP Address 2 (MReg Address 27)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IP_ADR[23:16]

Default: 00h Read only, write has no effect.

Byte 2 of the Management Station IP Address register. The Management Station IP Address is the 32-bit address used in the IP protocol header to address the station that functions as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Management Station IP Address 3 (MReg Address 28)

This register is loaded from the EEPROM.

Bit No.	Name and Description
7:0	M_IP_ADR[31:24]

Default: 00hRead only, write has no effect.

Byte 3 (the MSB) of the Management Station IP Address register. The Management Station IP Address is the 32-bit address used in the IP protocol header to address the station that functions as the management console in the network. This register can also be used as general purpose 8-bit register that is loaded from the EEPROM.

Device Status Registers

The following registers allow to determine the status of the Am79C975 device as it relates to the normal mode of operation (i.e. not related to the Serial Management Am79C975 Unit).

SMIU Am79C975 Status (MReg Address 29)

This register is a shadow register of some bits in CSR0, BCR19 and some bits in the PCI PCMCSR register.

Bit No.	Name and Description
7:6	POWERSTATE

Default: 00 Read only, write has no effect.

This two bit field indicates the current power state of the Am79C975 controller. The definition of the field values is given below:

00b - **D0**
01b - **D1**
10b - **D2**
11b - **D3**

5 RESERVED Default: 0Read as ZERO only

Reserved bits. For future use only

4 PVALID Default: 0Read only, write has no effect.

When PVALID is set to a 1, it indicates that there is an EEPROM connected to the Am79C975 controller and that the read operation of the EEPROM has passed the checksum verification. All registers that are loaded from the EEPROM contain data read from the EEPROM. PVALID is cleared by H_RESET.

3 RXON

Default: 0 Read only, write has no effect.

When RXON is set to a 1, it indicates that the receive function for normal operation is enabled. Note that the reception of management frames via the SMIU is only controlled by the MRX_ENABLE bit in the Receive Status register, but not of the state of RXON. RXON is cleared by H_RESET.

2 TXON

Default: 0 Read only, write has no effect.

When TXON is set to a 1, it indicates that the transmit function for normal operation is enabled. Note that the transmission of management frames via the SMIU is always active, independent of the state of TXON. TXON is cleared by H_RESET.

1 STOP

Default: 1 Read only, write has no effect.

When STOP is set to a 1, it indicates that all bus master activity of the Am79C975 controller is disabled. STOP is set to 1 by H_RESET.

0 STRT

Default: 0 Read only, write has no effect.

When STRT is set to a 1, it indicates that the Am79C975 controller is initialized for normal mode of operation and that the device is setup to perform bus master activity. STRT is cleared by H_RESET.

MIU Transceiver Status (MReg Address 30)

This register is a shadow register of some bits in the transceiver status register.

Bit No. Name and Description

7:4 RESERVED

Default: 0000Read as ZERO only

Reserved bits. For future use only

3 LINK

Default: 0 Read only, write has no effect.

When LINK is set to 1, it indicates that the physical layer interface of the Am79C975 controller is in Link Pass state. A value of 0 indicates a Link Fail state.

2 DUPLEX

Default: 0 Read only, write has no effect.

When DULPEX is set to 1, it indicates that the physical layer interface is operating in full-duplex mode. A value of 0 indicates half-duplex mode.

1 SPEED

Default: 0 Read only, write has no effect.

When SPEED is set to 1, it indicates that the physical layer interface is operating at 100 Mbps. A value of 0 indicates a speed of 10 Mbps.

0 AUTONEG_DONEDefault: 0Read only, write has no effect.

When AUTONEG_DONE is set to 1, it indicates that the autonegotiation process of the physical layer interface with the other end of the network cable has completed.

Control and Status Registers

The following registers control the transmission and reception of management frames and provide status of the operation.

SMIU Command Register (MReg Address 31)

Bit No. Name and Description

7 MIRQEN

Default: 0 Read/Write

MIRQEN allows the $\overline{\text{MIRQ}}$ pin to be active if the interrupt flag $\overline{\text{MIRQ}}$ in the SMIU Interrupt register is set. If MIRQEN is cleared to 0, the $\overline{\text{MIRQ}}$ pin will be disabled regardless of the state of $\overline{\text{MIRQ}}$. MRIRQEN is cleared by H_RESET.

6 MTX_DONEM

Default: 0 Read/Write

If MTX_DONEM is set to a 1, the MTX_DONE bit in the SMIU Interrupt register will be masked and unable to set the $\overline{\text{MIRQ}}$ bit. MTX_DONEM is cleared by H_RESET.

5 MRX_DONEM

Default: 0 Read/Write

If MRX_DONEM is set to a 1, the MRX_DONE bit in the SMIU Interrupt register will be masked and unable to set the $\overline{\text{MIRQ}}$ bit. MRX_DONEM is cleared by H_RESET.

4 RESERVED

Default: 0 Read/Write as ZERO only

Reserved bit. For future use only

3 MLOOP

Default: 0 Read/Write

If MLOOP is set to 0, transmit frames will be blocked from being received back, in case the transceiver loops back the data. Setting MLOOP to 1 enables loopback mode. All data that is transmitted will be received back, if the transceiver loop backs the data and the data passes the acknowledgment frame filter. The transceiver loopback can be achieved by programming the device into loopback mode or by using an external loopback connector. MLOOP has no effect., when the Am79C975 controller is configured for full-duplex operation. Receives are never blocked in full-duplex mode. MLOOP is cleared by H_RESET.

2 MRX_RPA

Default: 0 Read/Write

When MRX_RPA is set to a 1, the Am79C975 controller will accept runt frames (frames shorter than 64 bytes) that pass the acknowledgment frame filter. MRX_RPA is cleared by H_RESET.

1:0 RESERVED

Default: 00 Read/Write as ZERO only

Reserved bits. For future use only

SMIU Interrupt Register (MReg Address 32)

Bit No. Name and Description

7 $\overline{\text{MIRQ}}$

Default: 0 Read clear, write has no effect.

MIRQ indicates that one of the following interrupt causing conditions has occurred: MTX_DONE or MRX_DONE and the associated mask bit is programmed to allow the event to cause an interrupt. If the MIRQEN bit in the SMIU Command register is set to 1 and MIRQ is set, the $\overline{\text{MIRQ}}$ pin will be active. MIRQ is cleared by clearing all the active individual interrupt bits that have not been masked out, i.e. MIRQ will clear after reading the Interrupt register. MIRQ is also cleared by H_RESET.

6 MTX_DONE

Default: 0 Read clear, write has no effect.

MTX_DONE is set by the Am79C975 controller after an alert frame has been transmitted. When MTX_DONE is set, the $\overline{\text{MIRQ}}$ pin is asserted if MIRQEN is set to a 1 and the mask bit MTX_DONEM in the SMIU Command register is 0. MTIRQ is automatically cleared after reading the Interrupt register. MTX_DONE is also cleared by H_RESET.

5 MRX_DONE

Default: 0 Read clear, write has no effect.

MRX_DONE is set by the Am79C975 controller after an acknowledgment frame has been received. When MRX_DONE is set, the $\overline{\text{MIRQ}}$ pin is asserted if MIRQEN is set to a 1 and the mask bit MRX_DONEM in the SMIU Command register is 0. MRX_DONE is automatically cleared after reading the Interrupt register. MRX_DONE is also cleared by H_RESET.

4:0 RESERVED

Default: 00000Read as ZERO only

Reserved bits. For future use only

SMIU Receive Pattern RAM Address Register (MReg Address 33)

Bit No. Name and Description

7:0 MRX_PADR

Default: 00h Read/Write

The SMIU Receive Pattern RAM Address register contains the address of the location in the Receive Pattern RAM where the next byte of the Acknowledgment Frame Filter is written to. The register is cleared to 0 by H_RESET and every time the MRX_ENABLE bit is set to 1. The address register autoincrements with every byte write to the Receive Pattern RAM. This allows a FIFO-type access to the Receive Pattern RAM and the host does not need to keep track of the location he is writing to. In addition, MRX_PADR can be set to any address within the Receive Pattern RAM in order to modify a specific location.

MIU Receive Pattern RAM Data Port (MReg Address 34)

Bit No. Name and Description

7:0 MRX_PDATA

Default: undefined Read/Write

This is the 8-bit data port used to write to the Receive Pattern RAM.

SMIU Transmit Address Register (MReg Address 35)

Bit No. Name and Description

7:0 MTX_ADR

Default: 00h Read/Write

The SMIU Transmit Address register contains the address of the location in the Transmit Data memory where the next byte of data is written to. The register is cleared to 0 by H_RESET and every time the MTX_START bit in the Transmit Status register transitions is set to 1. The address register auto-increments with every byte written to the Transmit Data memory. This allows a FIFO-type access to the Transmit Data memory and the host does not need to keep track of the location he is writing to. In addition, MTX_ADR can be set to any address within the Transmit Data memory in order to modify a specific location.

SMIU Transmit Data Port (MReg Address 36)

Bit No. Name and Description

7:0 MTX_DATA

Default: undefined Write only, read has no effect.

This is the 8-bit data port used to write to the Transmit Data memory.

SMIU Transmit Message Length Register (MReg Address 37)

Bit No. Name and Description

7:0 MTX_ LEN

Default: 00h Read/Write

The SMIU Transmit Message Length contains the number of bytes from the Transmit Data memory that will be transmitted by the Am79C975 controller as the alert frame. The 4 bytes of FCS checksum are not part of the memory content, but are calculated by the controller and appended to the frame. The host is responsible to load a valid value into the Transmit Message Length register. Any value below 60 will create a runt frame on the network. Any value above 128 will create unpredictable results.

SMIU Transmit Status Register (MReg Address 38)

Bit No. Name and Description

7 MTX_START

Default: 0 Read/Write

When MTX_START is set to a 1, the Am79C975 controller will take the content of the SMIU Transmit Data memory and transmit the data as the next frame. Setting MTX_START will also clear the Transmit Address register. MTX_START is automatically cleared after every transmission. MTX_START is cleared by H_RESET.

6:4 RESERVED

Default: 000 Read/Write as ZERO only

Reserved bits. For future use only

3 MTX_LCOL

Default: 0 Read only, write has no effect.

Transmit Late Collision indicates that during the transmission of the alert frame a collision has occurred after the first slot time has

elapsed. The Am79C975 controller will not retry the transmission on late collision. MTX_LCOL is valid while MTX_START is set to 0. MTX_LCOL is cleared by H_RESET.

2 MTX_LCAR

Default: 0 Read only, write has no effect.

Transmit Loss of Carrier indicates that the transceiver was in Link Fail state during the transmission of the alert frame. MTX_LCAR is valid while MTX_START is set to 0. MTX_LCAR is cleared by H_RESET.

1 MTX_RTRY

Default: 0 Read only, write has no effect.

Transmit Retry error indicates that the transmission of the alert frame has failed after 16 attempts, due to repeated collisions on the network. MTX_RTRY is valid while MTX_START is set to 0. MTX_RTRY is cleared by H_RESET.

0 MTX_ERR

Default: 0 Read only, write has no effect.

Transmit Error is the OR of the MTX_LCOL, MTX_LCAR and MTX_RTRY error. MTX_ERR is valid while MTX_START is set to 0. MTX_ERR is cleared by H_RESET.

SMIU Receive Address Register (MReg Address 39)

Bit No. Name and Description

7:0 MRX_ADR

Default: 00h Read/Write

The SMIU Receive Address register contains the address within the Receive Data memory from where the next byte of data is read. The register is cleared to 0 by H_RESET and by setting MRX_ENABLE in the Receive Status register, which also unprotects the Receive Data memory from being overwritten with a new frame. The address register autoincrements with every byte read from the Receive Data memory. This allows a FIFO-type access to the Receive Data memory and the host does not need to keep track of the location he is reading from. In addition, MRX_ADR can be set to any address within the Receive Data memory in order to modify a specific location.

SMIU Receive Data Port (MReg Address 40)

Bit No. Name and Description

7:0 MRX_DATA

Default: undefinedRead only, write has no effect.

This is the 8-bit data port used to read from the Receive Data memory.

SMIU Receive Message Length Register (MReg Address 41)

Bit No. Name and Description

7:0 MRX _LEN

Default: undefinedRead only, write has no effect.

The SMIU Receive Message Length contains the length of the acknowledgment frame, including FCS. MRX_LEN only contains valid information after the Am79C975 controller has set the MRX_DONE bit in the Interrupt register. A message length value of larger than 128 indicates, that the Receive Data memory has overflowed. A message length between 129 and 132 bytes indicates that only FCS bytes have been lost and all message data bytes are available in the Receive Data memory. A message length greater than 132 indicates that message data bytes have been lost from the end of the message and the host should discard the frame. MRX_LEN will indicate the correct message length up to 255 bytes. It will freeze at 255 for longer frames. MRX_LEN is not affected by H_RESET.

SMIU Receive Status Register (MReg Address 42)

Bit No. Name and Description

7 MRX_ENABLE

Default: 0 Read/Write

When MRX_ENABLE is set to a 1, the Am79C975 controller is enabled to receive acknowledgment frames from the management station. The host must program the acknowledgment frame filter registers with valid data before setting MRX_ENABLE. Setting MRX_ENABLE to a 1 will also clear the Receive Pattern RAM Address and Receive Address registers. MRX_ENABLE is automatically cleared after every receive. MRX_ENABLE is cleared by H_RESET.

6:3 RESERVED

Default: 0000 Read/Write as ZERO only

Reserved bits. For future use only

2 MRX_FRAM

Default: 0 Read only, write has no effect.

Framing error indicates that the acknowledgment frame contains a non-integer multiple of eight bits and that there was an FCS error. If there was no FCS error on the acknowledgment frame, then MRX_FRAM will not be set even if there was a non-integer multiples of eight bits in the frame. MRX_FRAM is valid while MRX_ENABLE is set to 0. MTX_MRX_FRAM is cleared by H_RESET.

1 MRX_FCS

Default: 0 Read only, write has no effect.

Frame Check Sequence error indicates that the receiver has detected an FCS (CRC) error on the acknowledgment frame. MRX_FCS is valid while MRX_ENABLE is set to 0. MRX_FCS is cleared by H_RESET.

0 MRX_ERR

Default: 0 Read only, write has no effect.

Receive Error is the OR of the MRX_FRAM and MRX_FCS error. MRX_ERR is valid while MRX_ENABLE is set to 0. MRX_ERR is cleared by H_RESET.

Register Summary

MReg Address	Register Name	Access Mode	Access Command	Default Value
0	Vendor ID 0	RO	Read Byte	22h
1	Vendor ID 1	RO	Read Byte	10h
2	Device ID 0	RO	Read Byte	00h
3	Device ID 1	RO	Read Byte	20h
4	Revision ID	RO	Read Byte	40h
5	Subsystem Vendor ID 0	RO	Read Byte	00h
6	Subsystem Vendor ID 1	RO	Read Byte	00h
7	Subsystem ID 0	RO	Read Byte	00h
8	Subsystem ID 1	RO	Read Byte	00
9	Node IEEE Address 0	RO	Read Byte	00h
10	Node IEEE Address 1	RO	Read Byte	00h
11	Node IEEE Address 2	RO	Read Byte	00h
12	Node IEEE Address 3	RO	Read Byte	00h
13	Node IEEE Address 4	RO	Read Byte	00h
14	Node IEEE Address 5	RO	Read Byte	00h
15	Node IP Address 0	RO	Read Byte	00h
16	Node IP Address 1	RO	Read Byte	00h
17	Node IP Address 2	RO	Read Byte	00h
18	Node IP Address 3	RO	Read Byte	00h
19	Management Station IEEE Address 0	RO	Read Byte	00h
20	Management Station IEEE Address 1	RO	Read Byte	00h
21	Management Station IEEE Address 2	RO	Read Byte	00h
22	Management Station IEEE Address 3	RO	Read Byte	00h
23	Management Station IEEE Address 4	RO	Read Byte	00h
24	Management Station IEEE Address 5	RO	Read Byte	00h
25	Management Station IP Address 0	RO	Read Byte	00h
26	Management Station IP Address 1	RO	Read Byte	00h
27	Management Station IP Address 2	RO	Read Byte	00h
28	Management Station IP Address 3	RO	Read Byte	00h
29	Am79C975 Status	RO	Read Byte	02h
30	Transceiver Status	RO	Read Byte	00h
32	Command	RW	Read/Write Byte	00h
32	Interrupt	RC	Read Byte	00h
33	Receive Pattern RAM Address	RC,W	Read/Write Byte	00h
34	Receive Pattern RAM Data	RW	Block Write	undefined
35	Transmit Address	RW	Read/Write Byte	00h
36	Transmit Data	WO	Block Write	undefined
37	Transmit Message Length	RW	Read/Write Byte	00h
38	Transmit Status	RW	Read/Write Byte	00h
39	Receive Address	RW	Read/Write Byte	00h
40	Receive Data	RO	Block Read	undefined
41	Receive Message Length	RO	Read Byte	undefined
42	Receive Status	RW	Read/Write Byte	00h

Note: *RO = Ready Only, RC = Read/Auto-Clear, RW = Read/Write, W = Write, WO = Write Only*

Am79C975 EEPROM Map

The Am79C975 PCnet-*FAST III* controller uses an extended EEPROM map compared to the Am79C973 device (locations reserved in Am79C973 EEPROM map), since there are some new registers in the SMIU that are loaded from the EEPROM.

Am79C975 EEPROM Map

The Am79C975 PCnet-*FAST III* controller uses an extended EEPROM map compared to the Am79C973 device (locations reserved in Am79C973 EEPROM map), since there are some new registers in the SMIU that are loaded from the EEPROM.

Word Addr.	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	CSR116[15:8] (OnNow Misc. Config).	06h	CSR116[7:0] (OnNow Misc. Config.)
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	BCR22[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	BCR36[15:8] (Conf. Sp. byte 43h alias)	30h	BCR36[7:0] (Conf. Sp. byte 42h alias)
19h	33h	BCR37[15:8] (DATA_SCALE alias0)	32h	BCR37[7:0] (Conf. Sp. byte 47h0 alias)
1Ah	35h	BCR38[15:8] (DATA_SCALE alias 1)	34h	BCR38[7:0] (Conf. Sp. byte 47h1 alias)
1Bh	37h	BCR39[15:8] (DATA_SCALE alias 2)	36h	BCR39[7:0] (Conf. Sp. byte 47h2 alias)
1Ch	39h	BCR40[15:8] (DATA_SCALE alias 3)	38h	BCR40[7:0] (Conf. Sp. byte 47h3 alias)
1Dh	3Bh	BCR41[15:8] (DATA_SCALE alias 4)	3Ah	BCR41[7:0] (Conf. Sp. byte 47h4 alias)
1Eh	3Dh	BCR42[15:8] (DATA_SCALE alias 5)	3Ch	BCR42[7:0] (Conf. Sp. byte 47h5 alias)
1Fh	3Fh	BCR43[15:8] (DATA_SCALE alias 6)	3Eh	BCR43[7:0] (Conf. Sp. byte 47h6 alias)
20h	41h	BCR44[15:8] (DATA_SCALE alias 7)	40h	BCR44[7:0] (Conf. Sp. byte 47h7 alias)
21h	43h	BCR48[15:8] N_IP_ADR[15:8]	42h	BCR48[7:0] N_IP_ADR[7:0]
22h	45h	BCR49[15:8] N_IP_ADR[31:24]	44h	BCR49[7:0] N_IP_ADR[23:16]
23h	47h	BCR50[15:8] M_IEEE_ADR[15:8]	46h	BCR50[7:0] M_IEEE_ADR[7:0]
24h	49h	BCR51[15:8] M_IEEE_ADR[31:24]	48h	BCR51[7:0] M_IEEE_ADR[23:16]
25h	4Bh	BCR52[15:8] M_IEEE_ADR[47:40]	4Ah	BCR52[7:0] M_IEEE_ADR[39:32]
26h	4Dh	BCR53[15:8] M_IP_ADR[15:8]	4Ch	BCR53[7:0] M_IP_ADR[7:0]
27h	4Fh	BCR54[15:8] M_IP_ADR[31:24]	4Eh	BCR54[7:0] M_IP_ADR[23:16]
28h	51h	Checksum adjust byte for the 82 bytes of the EEPROM contents, checksum of the 82 bytes of the EEPROM should total to FFh	50h	BCR55[7:0] SMIU Slave Address
3Eh	7Dh	Reserved for Boot ROM usage	7Ch	Reserved for Boot ROM usage
3Fh	7Fh	Reserved for Boot ROM usage	7Eh	Reserved for Boot ROM usage

Note: ** Lowest EEPROM address.*

Absolute Maximum Ratings

Storage Temperature . . . . . –65°C to +150°C

Ambient Temperature. . . . . -65°C to +70°C

Supply voltage with respect to V_{SSB}, V_{SS}, D_{VSSD}, D_{VSSP}, and D_{VSSX} . . . . . –0.3 V to 3.63 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 (V_{DD}, V_{DDB}, V_{DD_PCI}, D_{VDDD}, D_{VDDA}, D_{VDDP}, D_{VDDTX}, D_{VDDRX}, and D_{VDDCO}). . . . . +3.3 V ±10%

All inputs within the range: . . . . . V_{SS} - 0.5 V to 5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC Characteristics

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
SMIU Input Voltage					
V_{IL}	Input LOW Voltage			0.6	V
V_{IH}	Input HIGH Voltage		1.4		V
SMIU Output Voltage					
V_{OL}	Output LOW Voltage	I_{OL} = 6 mA		0.4	V
SMIU Input Leakage Current					
I_{IX}	Input Leakage	V_{IN} = 0 V; V_{DD} = 3.3 V	-10	10	µA

Switching Characteristics

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
Clock Timing (Serial Management Interface)					
F_{MCLOCK}	MCLOCK Frequency		10	100	KHz
t_{HIGH}	MCLOCK High Time	@V_{IHmin}	4.0	50	µs
t_{LOW}	MCLOCK Low Time	@V_{ILmax}	4.7	--	µs
t_{FALL}	MCLOCK Fall Time	From V_{IHmin} to V_{ILmax}	--	300	ns
t_{RISE}	MCLOCK Rise Time	From V_{ILmax} to V_{IHmin}	--	1000	ns
Data Timing (Serial Management Interface)					
t_{SU}	MDATA Setup Time	@V_{ILmax}	250	--	ns
t_{HD}	MDATA Hold Time	@V_{ILmax}	300	--	ns
t_{FALL}	MDATA Fall Time	From V_{IHmin} to V_{ILmax}		300	ns
t_{RISE}	MDATA Rise Time	From V_{ILmax} to V_{IHmin}		1000	ns

Switching Waveforms

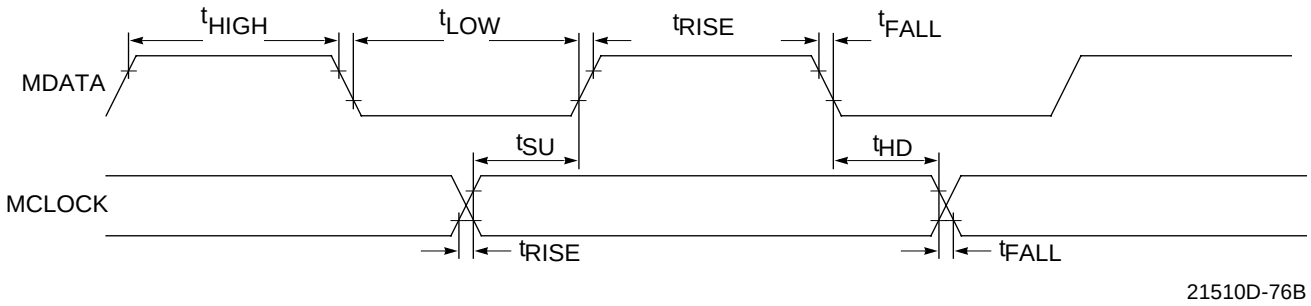


Figure 76. System Management Interface Timing

Media Independent Interface (MII)

APPENDIX C: MEDIA INDEPENDENT INTERFACE (MII)

Introduction

The Am79C973/Am79C975 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 Am79C973/Am79C975 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 or 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 C-77.

Note: *The MII interface is disabled by default. It is enabled by setting PHYSELEN (BCR2 bit 13) = 1 and PHYSEL (BCR18 bit 4, 3) =10.*

Enabling the MII interface automatically disables the internal 10/100 PHY and the Expansion Bus. When in this mode, the Am79C973/Am79C975 MII interface pins are multiplexed with the expansion bus pins. Refer to the connection diagram showing how the pins are multiplexed.

MII Transmit Interface

The MII transmit clock is generated by the external PHY and is sent to the Am79C973/Am79C975 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 to which the external PHY is attached. The data is a nibble-wide (4 bits) data path, TXD(3:0), from the Am79C973/Am79C975 controller to the external PHY and is synchronous to the rising edge of TX_CLK. The transmit process starts when the Am79C973/Am79C975 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 Am79C973/Am79C975 controller does not use this method of signaling errors on the transmit side. The Am79C973/Am79C975 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.

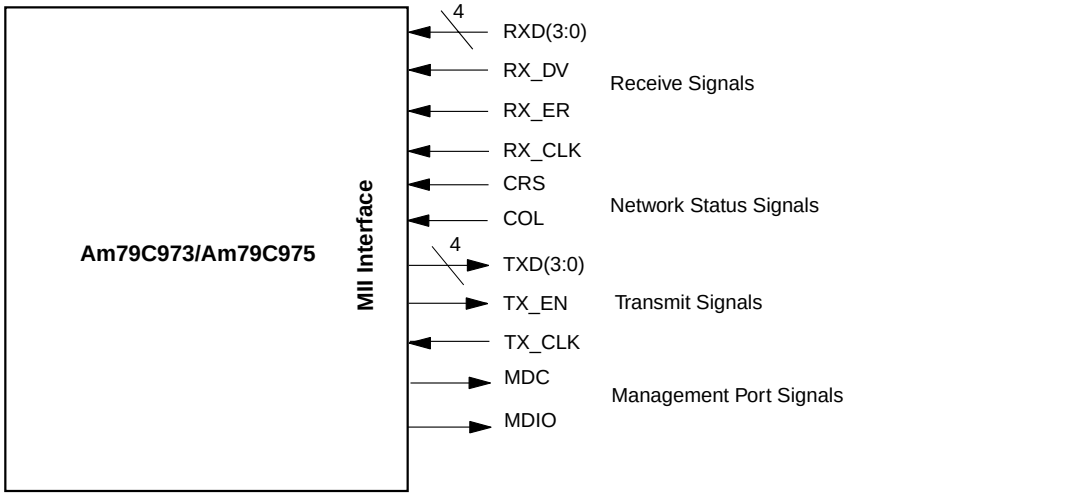


Figure 77. Media Independent Interface

MII Receive Interface

The MII receive clock is also generated by the external PHY and is sent to the Am79C973/Am79C975 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 to which the external PHY is attached.

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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller can control and receive status from external PHY devices.

The Am79C973/Am79C975 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 PHYADD after the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller is unique by offering direct hardware support of the external PHY device without software support. The PHY address of 1Fh is reserved and should not be used. 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 PHYs. The Am79C973/Am79C975 controller 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.

MII Management Frames

MII management frames are automatically generated by the Am79C973/Am79C975 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 C-78. 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 Am79C973/Am79C975 controller will then drop the creation of the preamble stream until a reset occurs, receives a read error, or the external PHY is disconnected.

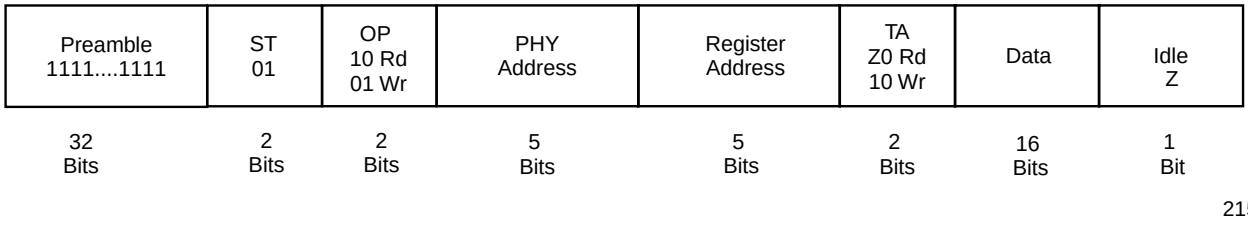


Figure 78. Frame Format at the MII Interface Connection

This is followed by a start field (ST) and an operation field (OP). The operation field (OP) indicates whether the Am79C973/Am79C975 controller is initiating a read or write operation. This is followed by the external PHY address (PHYADD) and the register address (REGAD) programmed in BCR33. The PHY address of 1Fh is reserved and should not be used. 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 Am79C973/Am79C975 controller will tri-state the MDIO for both MDC cycles.

During the second cycle, if the external PHY is synchronized to the Am79C973/Am79C975 controller, the external PHY will drive a 0. If the external PHY does not drive a 0, the Am79C973/Am79C975 controller will signal a MREINT (CSR7, bit 9) interrupt, if MREINTE (CSR7, bit 8) is set to a 1, indicating the Am79C973/Am79C975 controller had an MII management frame read error and that the data in BCR34 is not valid. The data field to/from the 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 Am79C973/Am79C975 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 of writing 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 Am79C973/Am79C975 controller's MII has no way of communicating important timely status information back to Am79C973/Am79C975 controller. The Am79C973/Am79C975 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 Am79C973/Am79C975 controller will Auto-Poll when APEP (BCR32, bit 11) is set to 1 to insure that the most current information is available. See *MII Management Registers* section 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 5) interrupt will be generated on $\overline{\text{INTA}}$ if the MAPINTE (CSR7, bit 4) 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 Am79C973/Am79C975 controller is unique in that is 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 Am79C973/Am79C975 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.

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

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 as defined in the IEEE 802.3u standard and according to Table C-68.

By default, the link partner must be at least 10BASE-T half-duplex capable. The Am79C973/Am79C975 controller can automatically negotiate with the network and yield the highest performance possible without software support. See the section on *Network Port Manager* for more details.

Table 68. 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

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 Am79C973/Am79C975 unless the DANAS (BCR32, bit 10) is selected and the software driver is capable of controlling the external PHY.* A complete bit description of the MII and Auto-Negotiation registers can be found in the *MII Management Registers* section.

Automatic Network Port Selection

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.

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 Am79C973/Am79C975 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).

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 clear the XPHYANE (BCR32, bit 5) and set either XPHYFD (BCR32, bit 4) or XPHYSP (BCR32, bit 3) or both. The Network Port Manager will send a few MII frames to the PHY to validate the configuration.

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 Am79C973/Am79C975 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.*

Am79C973/Am79C975's Automatic Network Port selection mechanism falls within the following general categories:

- External PHY *Not* Auto-Negotiable
- External PHY Auto-Negotiable

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

This case occurs when the MIIPD (BCR32, bit 14) bit is 1. This indicates that there is an external PHY attached to Am79C973/Am79C975 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 Am79C973/Am79C975 controller will read the register of the external PHY to determine its status and network capabilities. See *the MII Management Registers* section 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 Am79C973/Am79C975 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 Am79C973/Am79C975 controller will fail the external PHY link. The Network Port Man-

ager will periodically query the external PHY for active links.

Automatic Network Selection: External PHY Auto-Negotiable

This case occurs when the MIIPD (BCR32, bit 14) bit is 1. This indicates that there is an external PHY attached to Am79C973/Am79C975 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 Am79C973/Am79C975 controller will read the MII Status register of the external PHY to determine its status and network capabilities. If the external PHY is Auto-Negotiation capable and/or the XPHYANE (BCR32, bit 5) bit is set to 1, then the Am79C973/Am79C975 controller will start the external PHY's Auto-Negotiation process. The Am79C973/Am79C975 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 from the external PHY MII Status register read, and set the Type Selector field to the IEEE 802.3 standard. The Am79C973/Am79C975 controller will then write to the external PHY's MII Control register instructing the external PHY to negotiate the link. The Am79C973/Am79C975 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 Am79C973/Am79C975 controller will then wait a specific time and then again read the external PHY's MII Status register. If the Am79C973/Am79C975 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 Am79C973/Am79C975 controller will fail the external PHY link and start the process again.

Automatic Network Selection: Force External Reset

If the XPHYRST bit (BCR32, bit 6) is set to 1, then the flow changes slightly. The Am79C973/Am79C975 controller will write to the external PHY's MII Control register with the RESET bit set to 1 (See *the MII Management Registers* section for the MII register bit descriptions). This will force a complete reset of the external PHY. The Am79C973/Am79C975 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

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 nibble as it arrives at the Am79C973/Am79C975 controller, to compare the captured address with a table of stored addresses or identifiers, and then to determine whether or not the Am79C973/Am79C975 controller should accept the packet.

If an address match is detected by comparison with either the Physical Address or Logical Address Filter registers contained within the Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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 Am79C973/Am79C975 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. In order for the $\overline{EAR}$ pin to be functional in full-duplex mode, FDRPAD bit (BCR9, bit 2) needs to be set. 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-powerdown mode in that the Am79C973/Am79C975 control-

ler 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 C-69 summarizes the operation of the EADI interface.

Table 69. EADI Operations

PROM	EAR	Required Timing	Received Frames
1	X	No timing requirements	All received frames
0	1	No timing requirements	All received frames
0	0	Low for two bit times plus 10 ns	Frame rejected if in address match mode

When using the MII, 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. Data for the EADI is the RXD(3:0) receive data provided to the MII. 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 Am79C973/Am79C975 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 should be driven LOW by the external address comparison logic to reject a frame.

External Address Detection Interface: Receive Frame Tagging

The Am79C973/Am79C975 controller supports receive frame tagging in the MII mode. The receive frame tagging implementation will be a two--wire chip interface, respectively, added to the existing EADI.

The Am79C973/Am79C975 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.

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 Am79C973/Am79C975 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 C-79.

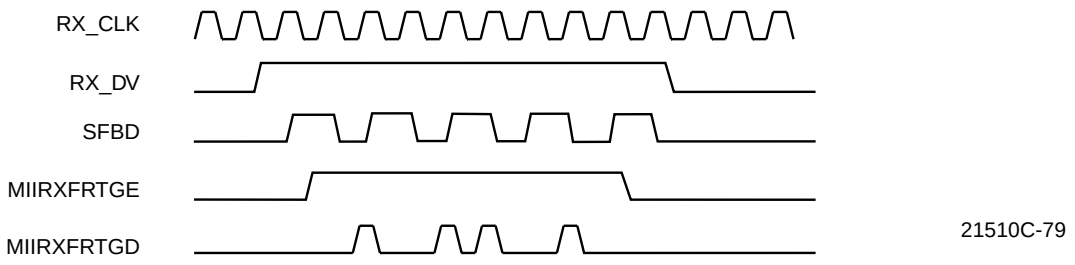


Figure 79. MII Receive Frame Tagging

MII management registers

As specified in the IEEE standard, the basic register set consists of the Control Register (Register 0) and the Status Register (Register 1). The extended register set consists of Registers 2 to 31 (decimal). All PHYs that provide an MII shall incorporate the basic register set. Both sets of registers are accessible through the MII Management Interface.

Table C-70 lists the most interesting registers.

Table 70. MII Management Register Set

Register Address	Register Name	Basic/ Extended
0	MII Control	B
1	MII Status	B
2-3	PHY Identifier	E
4	Auto-Negotiation Advertisement	E
5	Auto-Negotiation Link Partner Ability	E

Control Register (Register 0)

Table C-71 shows the MII Management Control Register (Register 0).

Table 71. MII Management Control Register (Register 0)

Bits	Name	Description	Read/Write (Note 1)
15	Soft Reset	When write: 1 = PHY software reset 0 = normal operation. When read: 1 = reset in process 0 = reset done.	R/W, SC
14	Loopback	1 = enables Loopback mode 0 = disables Loopback mode	R/W
13	Speed Selection	1 = 100 Mbps 0 = 10 Mbps	R/W
12	Auto-Negotiation Enable	1 = enable Auto-Negotiation 0 = disable Auto-Negotiation	R/W
11	Power Down	1 = power down, 0 = normal operation	R/W
10	Isolate	1 = electrically isolate PHY from MII 0 = normal operation	R/W
9	Restart Auto-Negotiation	1 = restart Auto-Negotiation 0 = normal operation	R/W, SC
8	Duplex Mode	1 = full duplex 0 = half duplex	R/W
7	Collision Test	1 = enable COL signal test 0 = disable COL signal test	R/W
6-0	Reserved	Write as 0, ignore on read	RO

Note:

1. R/W = Read/Write, SC = Self Clearing, RO = Read only.

Status Register (Register 1)

The MII Management Status Register identifies the physical and auto-negotiation capabilities of the PHY. This register is read only; a write will have no effect. See Table C-72.

Table 72. MII Management Status Register (Register 1)

Bits	Name	Description	Read/Write (Note 1)
15	100BASE-T4	1 = PHY able to perform 100BASE-T4 0 = PHY not able to perform 100BASE-T4	RO
14	100BASE-X Full Duplex	1 = PHY able to perform full-duplex 100BASE-X 0 = PHY not able to perform full-duplex 100BASE-X	RO
13	100BASE-X Half Duplex	1 = PHY able to perform half-duplex 100BASE-X 0 = PHY not able to perform half-duplex 100BASE-X	RO
12	10 Mbps Full Duplex	1 = PHY able to operate at 10 Mbps full-duplex mode 0 = PHY not able to operate at 10 Mbps full-duplex mode	RO
11	10 Mbps Half Duplex	1 = PHY able to operate at 10 Mbps full-duplex mode 0 = PHY not able to operate at 10 Mbps full-duplex mode	RO
10-7	Reserved	Ignore when read	RO
6	MF Preamble Suppression	1 = PHY will accept management frames with preamble suppressed 0 = PHY not able to accept management frames with preamble suppressed	RO
5	Auto-Negotiation Complete	1 = Auto-Negotiation process completed 0 = Auto-Negotiation process not completed	RO
4	Remote Fault	1 = remote fault condition detected 0 = no remote fault condition detected	RO, LH
3	Auto-Negotiation Ability	1 = PHY is able to perform Auto-Negotiation 0 = PHY is not able to perform Auto-Negotiation	RO
2	Link Status	1 = link is up 0 = link is down	RO, LL
1	Jabber Detect	1 = jabber condition detected, 0 = no jabber condition detected	RO
0	Extended Capability	1 = extended register capabilities, 0 = basic register set capabilities only	RO

Note:

1. RO = Read Only, LH = Latching High, LL = Latching Low.

Auto-Negotiation Advertisement Register (Register 4)

The purpose of this register is to advertise the technology ability to the link partner device. See Table C-73.

When this register is modified, Restart Auto-Negotiation (Register 0, bit 9) must be enabled to guarantee the change is implemented.

Table 73. Auto-Negotiation Advertisement Register (Register 4)

Bit(s)	Name	Description	Read/ Write
15	Next Page	When set, the device wishes to engage in next page exchange. If clear, the device does not wish to engage in next page exchange.	R/W
14	Reserved		RO
13	Remote Fault	When set, a remote fault bit is inserted into the base link code word during the Auto Negotiation process. When cleared, the base link code word will have the bit position for remote fault as cleared.	R/W
12:5	Technology Ability	Link partner technology ability field.	RO
4:0	Selector Field	Link partner selector field.	RO

Technology Ability Field Bit Assignments

The technology bit field consists of bits A0-A7 in the IEEE 802.3 Selector Base Page. Table C-74 summarizes the bit assignments.

Table 74. Technology Ability Field Bit Assignments

Bit	Technology
A0	10BASE-T
A1	10BASE-T Full Duplex
A2	100BASE-TX
A3	100BASE-TX Full Duplex
A4	100BASE-T4
A5	Reserved for future technology
A6	Reserved for future technology
A7	Reserved for future technology

Auto-Negotiation Link Partner Ability Register (Register 5)

The Auto-Negotiation Link Partner Ability Register is Read Only. The register contains the advertised ability of the link partner. The bit definitions represent the received link code word. This register contains either the base page or the link partner's next pages. See Table C-75.

Table 75. Auto-Negotiation Link Partner Ability Register (Register 5) - Base Page Format

Bit(s)	Name	Description	Read/ Write
15	Next Page	Link partner next page request.	RO
14	Acknowledge	Link partner acknowledgment	RO
13	Remote Fault	Link partner remote fault request	RO
12:5	Technology Ability	Link partner technology ability field	RO
4:0	Selector Field	Link partner selector field.	RO

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 Waveforms: Media Independent Interface

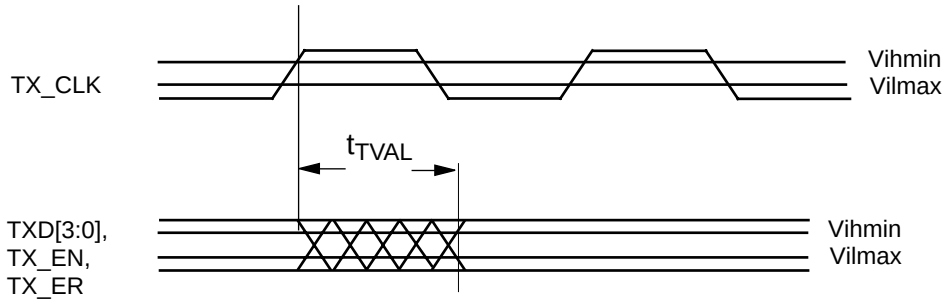


Figure 80. Transmit Timing

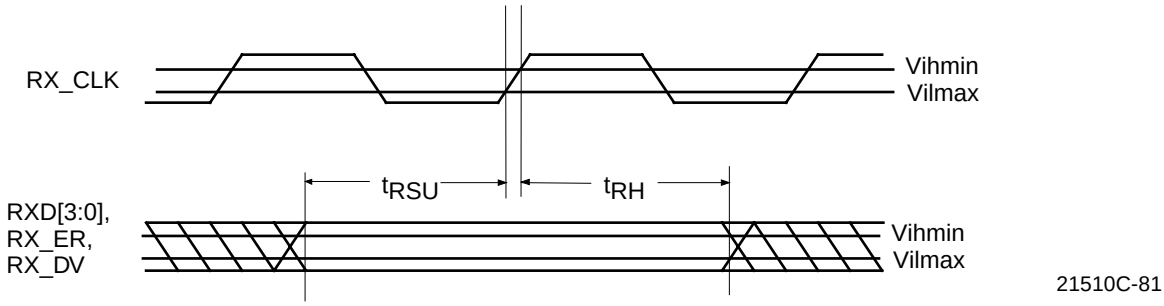


Figure 81. Receive Timing

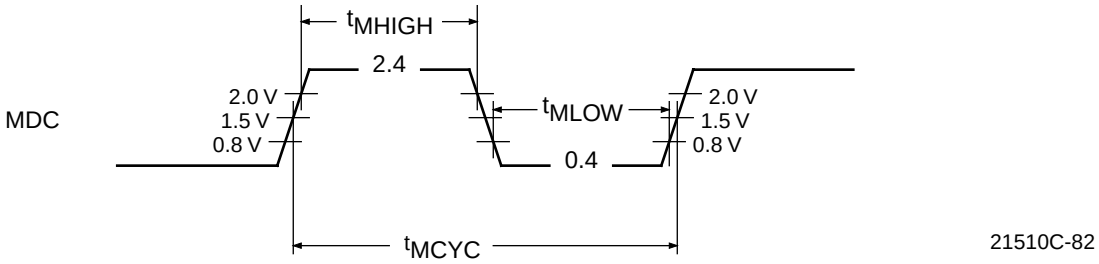


Figure 82. MDC Waveform

Switching Waveforms: Media Independent Interface (Concluded)

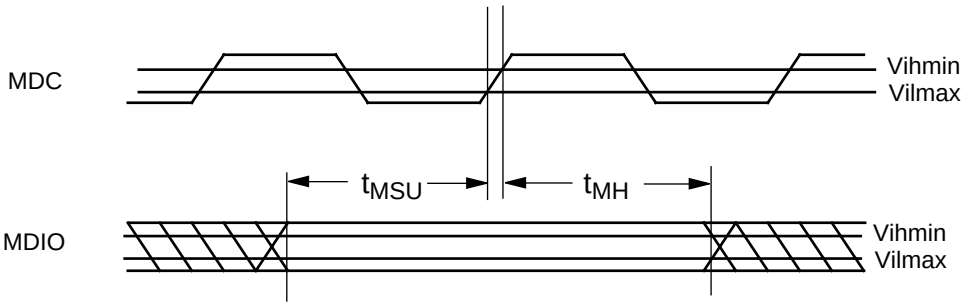


Figure 83. Management Data Setup and Hold Timing

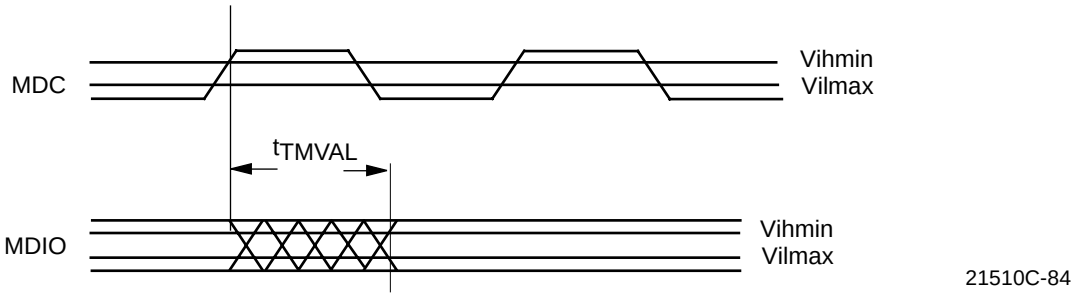


Figure 84. Management Data Output Valid Delay Timing

Switching Waveforms: External Address Detection Interface

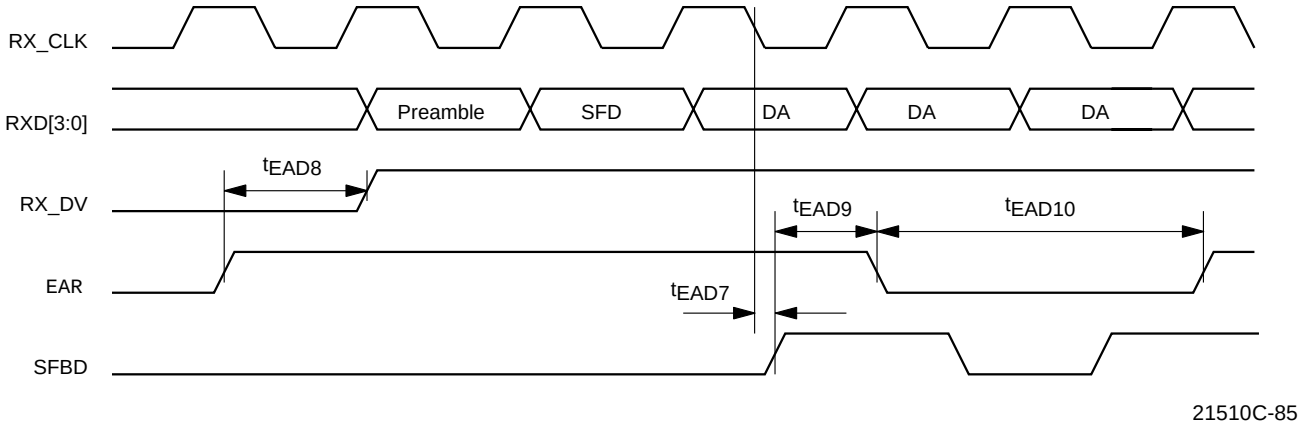


Figure 85. Reject Timing - External PHY MII @ 25 MHz

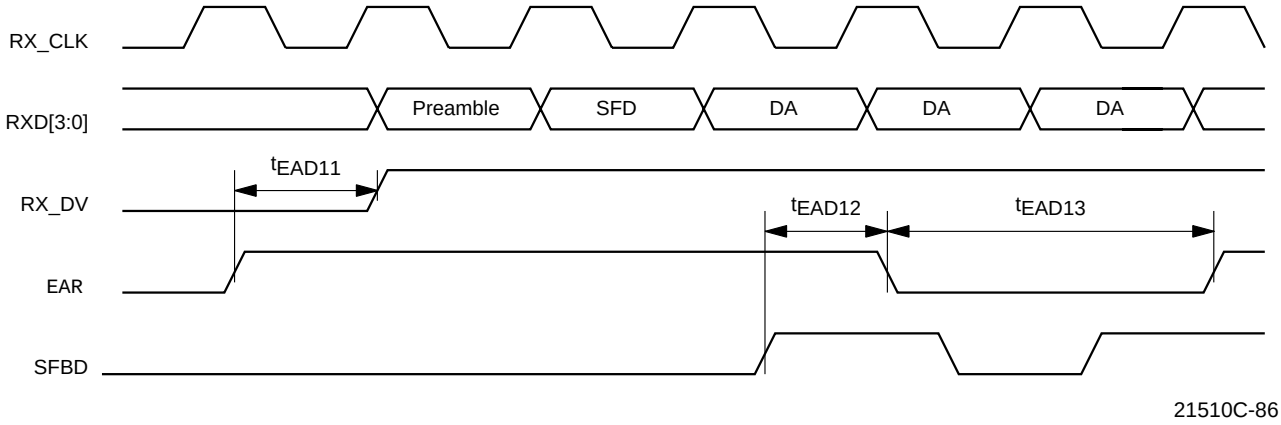


Figure 86. Reject Timing - External PHY MII @ 2.5 MHz

Switching Waveforms: Receive Frame Tag

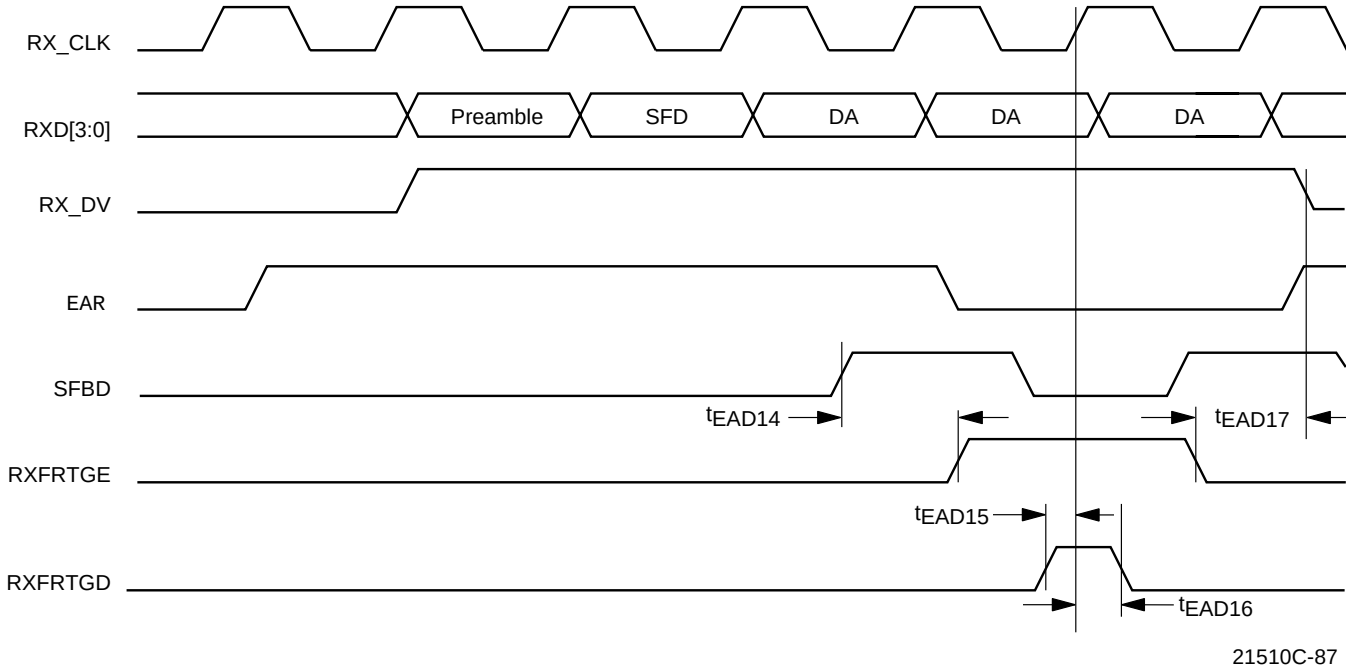


Figure 87. Receive Frame Tag Timing with Media Independent Interface

Alternative Method for Initialization

APPENDIX D: Alternative Method for Initialization

The Am79C973/Am79C975 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 Table D-76. These register writes are followed by writing the START bit in CSR0.

Table 76. Registers for Alternative Initialization Method (Note 1)

Control and Status Register	Comment
CSR2	IADR[31:16] (Note 2)
CSR8	LADRF[15:0]
CSR9	LADRF[31:16]
CSR10	LADRF[47:32]
CSR11	LADRF[63:48]
CSR12	PADR[15:0] (Note 3)
CSR13	PADR[31:16] (Note 3)
CSR14	PADR[47:32] (Note 3)
CSR15	Mode
CSR24-25	BADR
CSR30-31	BADX
CSR47	TXPOLLINT
CSR49	RXPOLLINT
CSR76	RCVRL
CSR78	XMTRL

Note:

1. *The INIT bit must not be set or the initialization block will be accessed instead.*
2. *Needed only if SSIZE32 =0.*
3. *Needed only if the physical address is different from the one stored in EEPROM or if there is no EEPROM present.*

AMD

Look-Ahead Packet Processing (LAPP) Concept

APPENDIX E: LOOK-AHEAD PACKET PROCESSING (LAPP) CONCEPT

Introduction

A driver for the Am79C973 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 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 Am79C973 controller's data transfers to its buffer space are such that the system bus is needed by the Am79C973 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 Am79C973 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 Am79C973 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 Am79C973 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 Am79C973 controller.

Note: *The labels in the following text are used as references in the timeline diagram that follows (Figure B-1).*

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 Am79C973 controller to generate an INTERRUPT when STP has been written to a receive descriptor by the Am79C973 controller.

Flow

The Am79C973 controller polls the current receive descriptor at some point in time before a message arrives. The Am79C973 controller determines that this receive buffer is OWNed by the Am79C973 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 Am79C973 controller to begin frame data DMA operations to the first buffer.

C0 When the 64th byte of the message arrives, the Am79C973 controller performs a lookahead operation to the next receive descriptor. This descriptor should be owned by the Am79C973 controller.

C1 The Am79C973 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 Am79C973 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 Am79C973 controller to the CPU, the Am79C973 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 Am79C973 controller's driver to run.

C4 During the CPU interrupt-generated task switching, the Am79C973 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 Am79C973 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 Am79C973 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 Am79C973 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 Am79C973 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 Am79C973 controller.

C5 Interleaved with S2, S3, and S4 driver activity, the Am79C973 controller will write frame data to buffer number 2.

S4 The driver will next proceed to copy the contents of the Am79C973 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 Am79C973 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 Am79C973 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 appli-

cation pointer into this descriptor and then changed the ownership to give the descriptor to the Am79C973 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 Am79C973 controller will write the status and change the ownership bit of descriptor number 2.

S6 After the ownership of descriptor number 2 has been changed by the Am79C973 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 Am79C973 controller will perform data DMA to the last buffer, whose pointer is pointing to application space. Data entering the least 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.

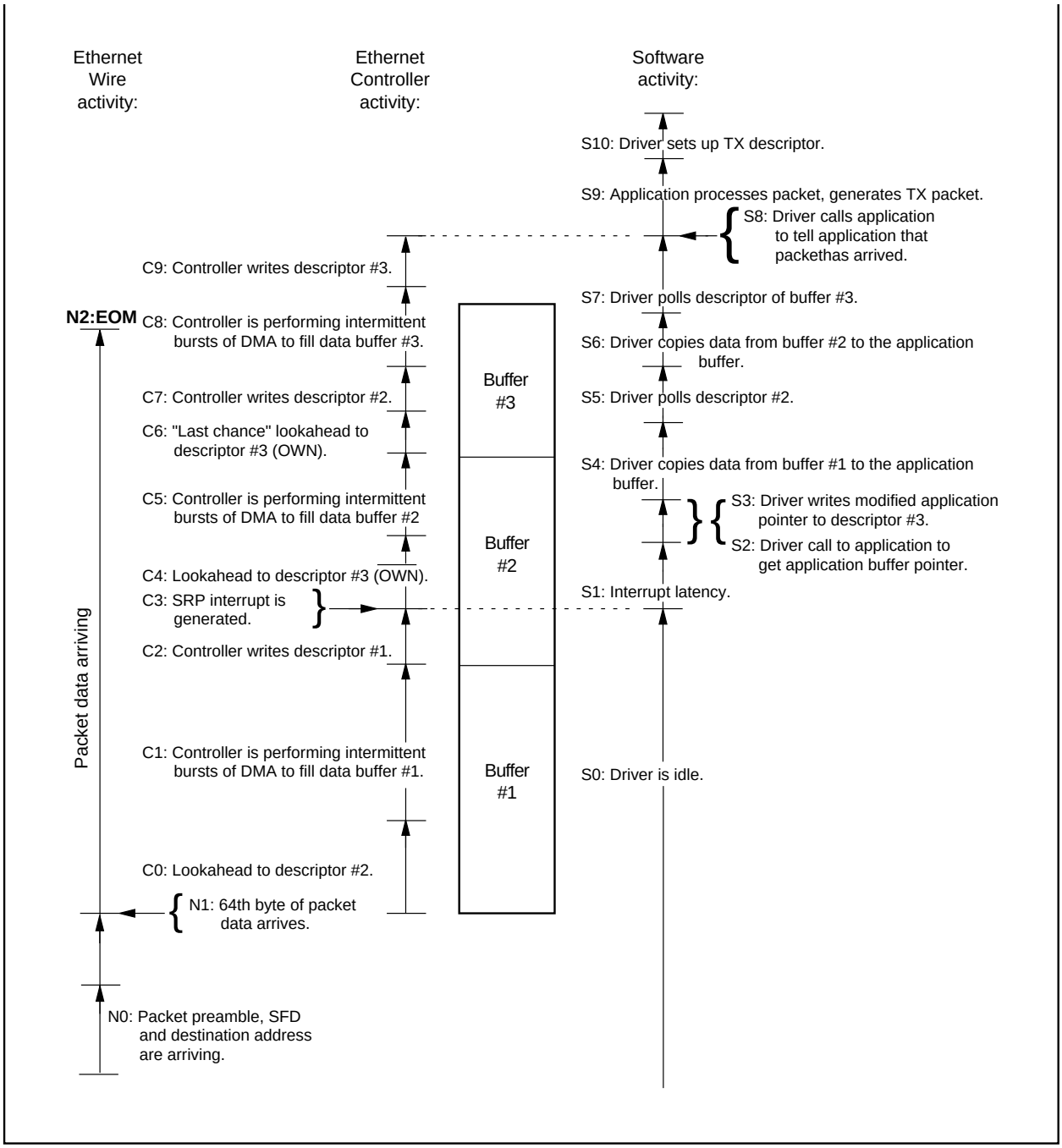
S7 When the driver completes the copy of buffer number 2 data to the application buffer space, it begins polling descriptor number 3.

C9 When the Am79C973 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 Am79C973 controller.



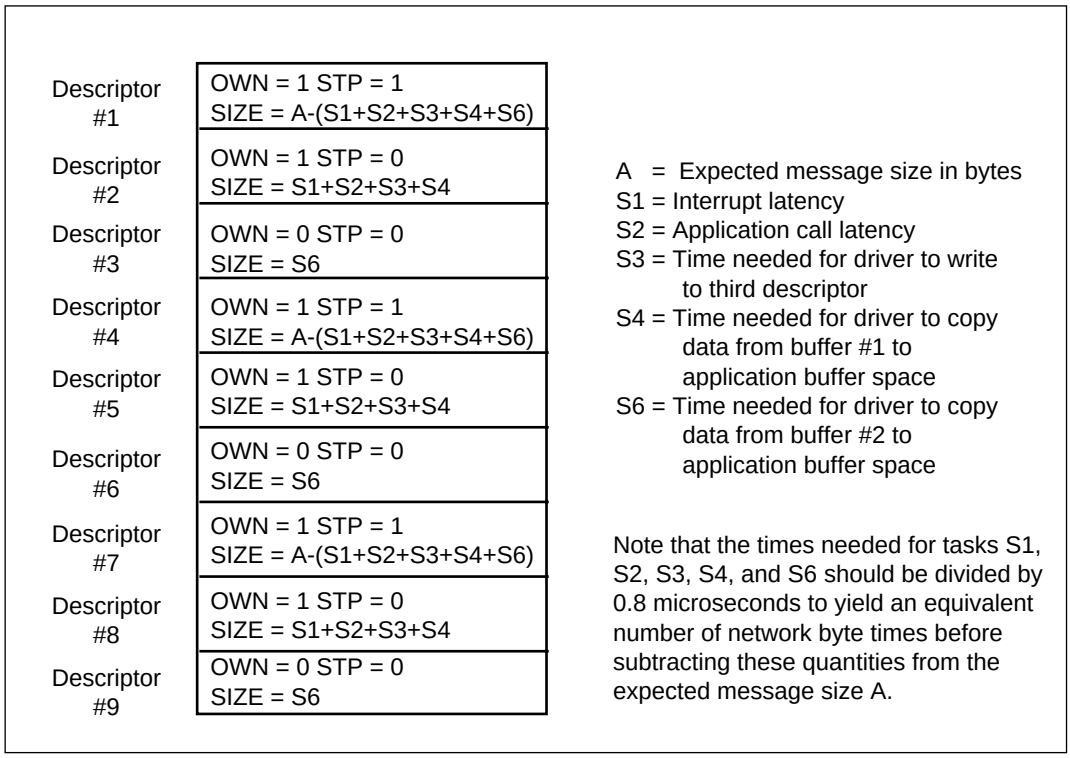
21510B-B1

Figure 88. 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 B-2 illustrates this setup for a receive ring size of 9.



21510B-B2

Figure 89. 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: *The Am79C973 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 Am79C973 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 Am79C973 controller has completed its poll of the next descriptors. This means that for almost all occurrences of this case, the Am79C973 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 Am79C973 controller will not have had an opportunity to modify it.*

***Note that even though the Am79C973 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 Am79C973 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

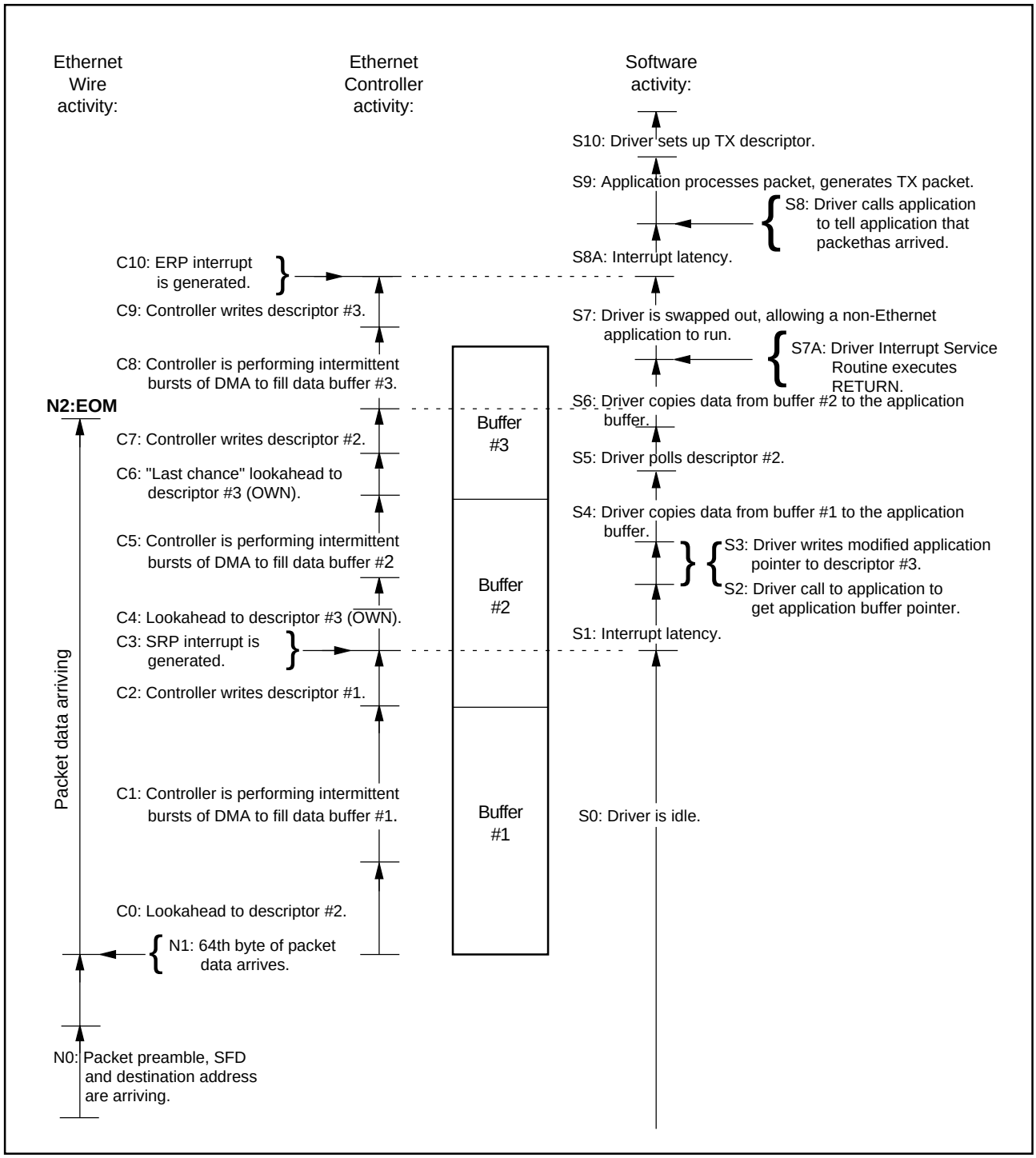
An alternative to the above suggested flow is to use two interrupts, one at the start of the receive frame and the other at the end of the receive frame, instead of just looking for the SRP interrupt as described above. This alternative attempts to reduce the amount of time that the software wastes while 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 B-3.

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 B-4 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.



21510B-B3

Figure 90. LAPP Timeline for Two-Interrupt Method

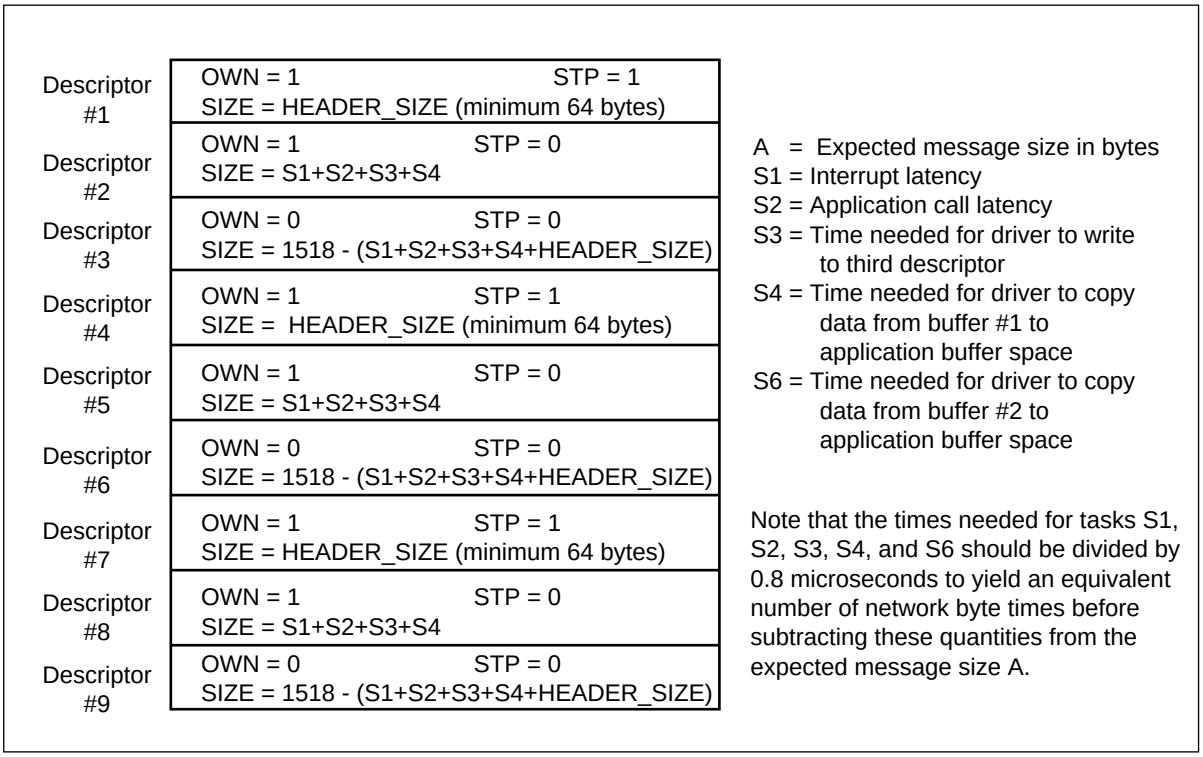


Figure 91. LAPP 3 Buffer Grouping for Two-interrupt Method

INDEX

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X