



Am79C961

PCnet™-ISA+ Jumperless Single-Chip Ethernet Controller for ISA

DISTINCTIVE CHARACTERISTICS

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

GENERAL DESCRIPTION

The PCnet-ISA+ controller, a single-chip Ethernet controller, is a highly integrated system solution for the PC-AT Industry Standard Architecture (ISA) architecture. It is designed to provide flexibility and compatibility with any existing PC application. This highly integrated 132-pin VLSI device is specifically designed to reduce parts count and cost, and addresses applications where higher system throughput is desired. The PCnet-ISA+

controller is fabricated with AMD's advanced low-power CMOS process to provide low standby current for power sensitive applications.

The PCnet-ISA+ controller is a DMA-based device with a dual architecture that can be configured in two different operating modes to suit a particular PC application. In the Bus Master Mode all transfers are performed using

the integrated DMA controller. This configuration enhances system performance by allowing the PCnet-ISA⁺ controller to bypass the platform DMA controller and directly address the full 24-bit memory space. The implementation of Bus Master Mode allows minimum parts count for the majority of PC applications. The PCnet-ISA⁺ controller can be configured to perform Shared Memory operations for compatibility with low-end machines, such as PC/XTs that do not support Bus Master and high-end machines that require local packet buffering for increased system latency.

The PCnet-ISA⁺ controller is designed to directly interface with the ISA or EISA system bus. It contains an ISA Plug and Play bus interface unit, DMA Buffer Management Unit, 802.3 Media Access Control function, individual 136-byte transmit and 128-byte receive FIFOs, IEEE 802.3 defined Attachment Unit Interface (AUI), and a Twisted Pair Transceiver Media Attachment Unit. The PCnet-ISA⁺ controller is also register compatible with the LANCE (Am7990) Ethernet controller and PCnet-ISA. The DMA Buffer Management Unit supports the LANCE descriptor software model.

External remote boot and Ethernet physical address PROMs and Electrically Erasable Proms are also supported.

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

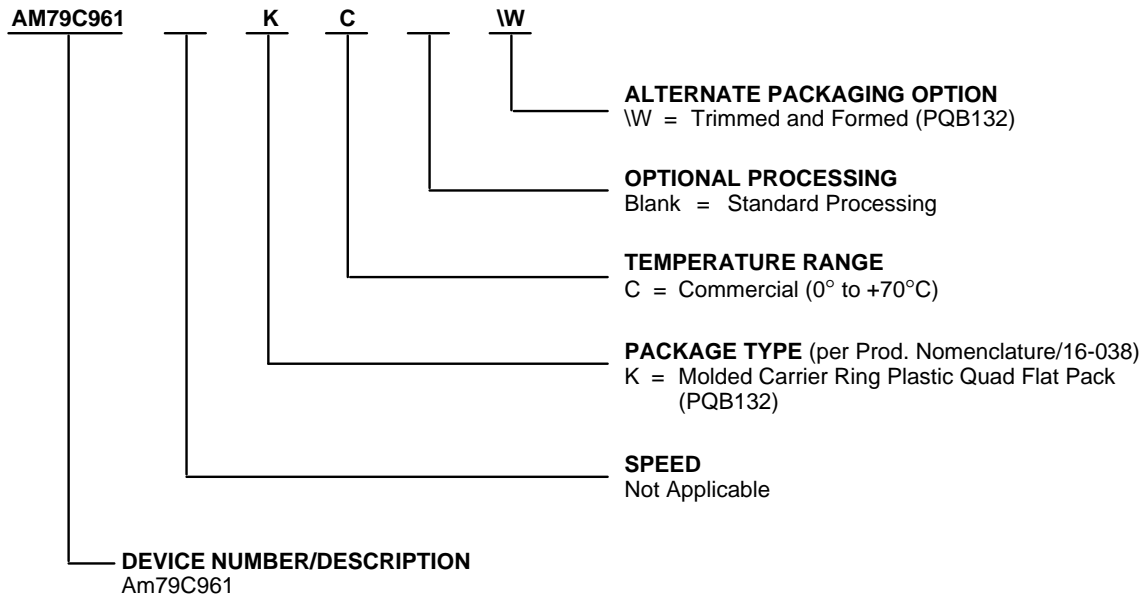
RELATED PRODUCTS

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

ORDERING INFORMATION

Standard Products

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



| Valid Combinations | |
|--------------------|---------|
| AM79C961 | KC, KCW |

Valid Combinations

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

| | |
|---|-------|
| TABLE OF CONTENTS | |
| DISTINCTIVE CHARACTERISTICS | 1-475 |
| GENERAL DESCRIPTION | 1-475 |
| RELATED PRODUCTS | 1-476 |
| ORDERING INFORMATION | 1-477 |
| BLOCK DIAGRAM: BUS MASTER MODE | 1-484 |
| CONNECTION DIAGRAM: BUS MASTER | 1-484 |
| PIN DESIGNATIONS: BUS MASTER | |
| LISTED BY PIN NUMBER | 1-486 |
| LISTED BY PIN NAME | 1-487 |
| LISTED BY GROUP | 1-488 |
| PIN DESCRIPTION: BUS MASTER MODE | 1-490 |
| ISA INTERFACE | 1-490 |
| BOARD INTERFACE | 1-491 |
| BLOCK DIAGRAM: SHARED MEMORY | 1-493 |
| CONNECTION DIAGRAM: SHARED MEMORY | 1-494 |
| PIN DESIGNATIONS: SHARED MEMORY | |
| LISTED BY PIN NUMBER | 1-495 |
| LISTED BY PIN NAME | 1-496 |
| LISTED BY GROUP | 1-497 |
| PIN DESCRIPTION: SHARED MEMORY MODE | 1-499 |
| ISA INTERFACE | 1-499 |
| BOARD INTERFACE | 1-500 |
| PIN DESCRIPTION: NETWORK INTERFACES (mode independent) | 1-502 |
| AUI | 1-502 |
| TWISTED PAIR INTERFACE | 1-502 |
| IEEE 1149.1 TEST ACCESS PORT INTERFACE | 1-502 |
| PIN DESCRIPTION: POWER SUPPLIES (mode independent) | 1-502 |
| FUNCTIONAL DESCRIPTION | 1-503 |
| BUS MASTER MODE | 1-503 |
| SHARED MEMORY MODE | 1-505 |
| NETWORK INTERFACE | 1-505 |
| PLUG AND PLAY | 1-507 |
| DETAILED FUNCTIONS | 1-514 |
| EEPROM | 1-514 |
| SERIAL EEPROM BYTE MAP | 1-515 |
| PLUG AND PLAY REGISTER MAP | 1-517 |
| PLUG AND PLAY REGISTER LOCATIONS DETAILED DESCRIPTION | 1-518 |
| SHARED MEMORY CONFIGURATION BITS | 1-520 |
| USE WITHOUT EEPROM | 1-521 |
| EXTERNAL SCAN CHAIN | 1-521 |
| FLASH PROM | 1-521 |
| OPTIONAL IEEE ADDRESS PROM | 1-521 |
| EISA CONFIGURATION REGISTERS | 1-521 |

| | |
|--|-------|
| BUS INTERFACE UNIT (BIU) | 1-521 |
| DMA Transfers | 1-521 |
| 1. Initialization Block DMA Transfers | 1-521 |
| 2. Descriptor DMA Transfers | 1-522 |
| 3. Burst-Cycle DMA Transfers | 1-522 |
| BUFFER MANAGEMENT UNIT (BMU) | 1-522 |
| Initialization | 1-522 |
| Reinitialization | 1-522 |
| Buffer Management | 1-522 |
| Descriptor Rings | 1-522 |
| Descriptor Ring Access Mechanism | 1-523 |
| Polling | 1-523 |
| Transmit Descriptor Table Entry (TDTE) | 1-525 |
| Receive Descriptor Table Entry (RDTE) | 1-526 |
| MEDIA ACCESS CONTROL | 1-527 |
| Transmit and Receive Message Data Encapsulation | 1-527 |
| Media Access Management | 1-528 |
| MANCHESTER ENCODER/DECODER (MENDEC) | 1-530 |
| External Crystal Characteristics | 1-530 |
| External Clock Drive Characteristics | 1-530 |
| MENDEC Transmit Path | 1-530 |
| Transmitter Timing and Operation | 1-530 |
| Receive Path | 1-531 |
| Input Signal Conditioning | 1-531 |
| Clock Acquisition | 1-531 |
| PLL Tracking | 1-531 |
| Carrier Tracking and End of Message | 1-532 |
| Data Decoding | 1-532 |
| Differential Input Terminations | 1-532 |
| Collision Detection | 1-532 |
| Jitter Tolerance Definition | 1-532 |
| Attachment Unit Interface (AUI) | 1-532 |
| TWISTED PAIR TRANSCEIVER (T-MAU) | 1-532 |
| Twisted Pair Transmit Function | 1-533 |
| Twisted Pair Receive Function | 1-533 |
| Link Test Function | 1-533 |
| Polarity Detection and Reversal | 1-533 |
| Twisted Pair Interface Status | 1-534 |
| Collision Detect Function | 1-534 |
| Signal Quality Error (SQE) Test (Heartbeat) Function | 1-534 |
| Jabber Function | 1-534 |
| Power Down | 1-534 |
| EADI™ (External Address Detection Interface™) | 1-534 |
| GENERAL PURPOSE SERIAL INTERFACE (GPSI) | 1-536 |
| IEEE 1149.1 TEST ACCESS PORT INTERFACE | 1-537 |
| Boundary Scan Circuit | 1-537 |

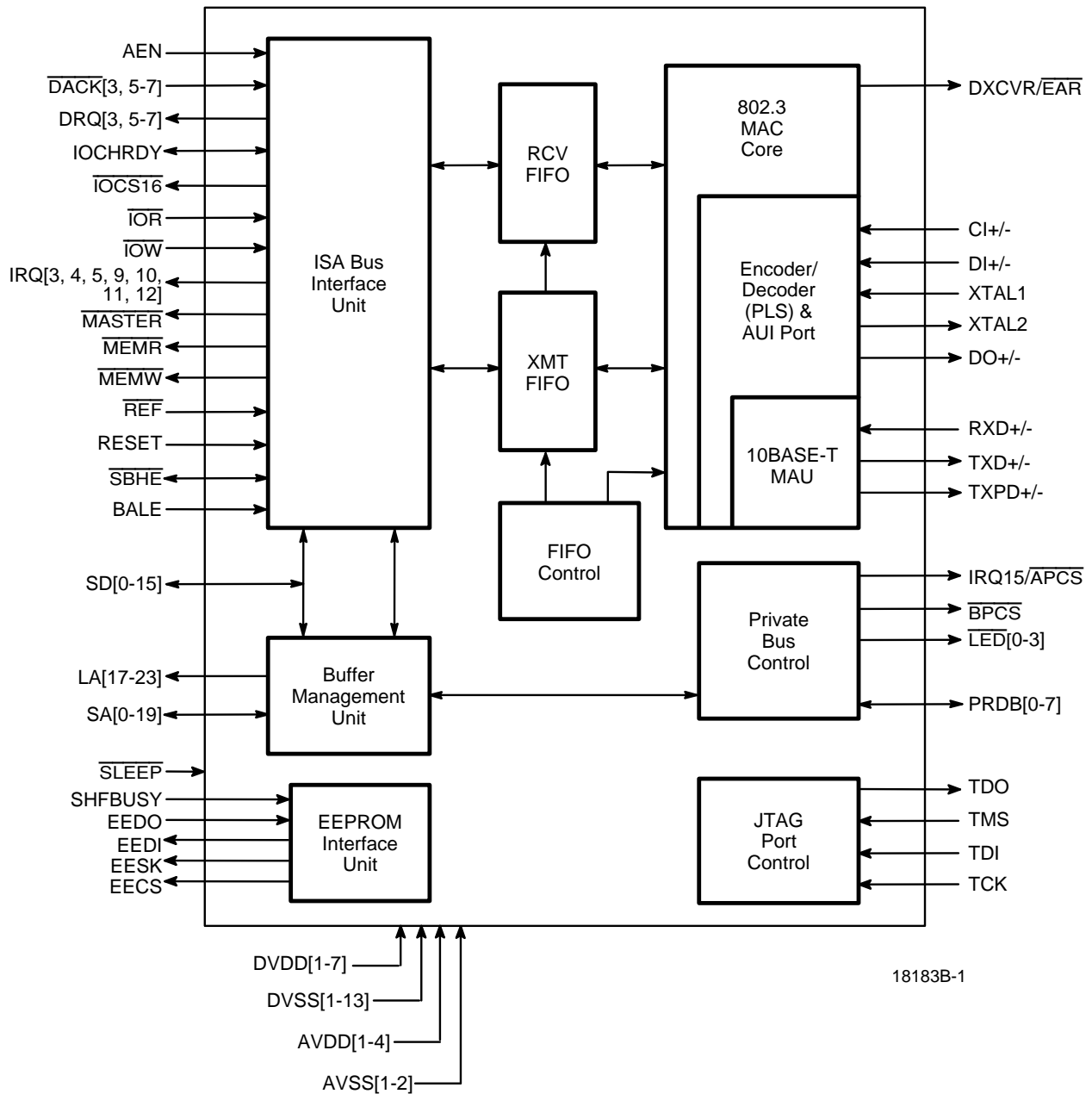
| | |
|---|-------|
| TAP FSM | 1-537 |
| Supported Instructions | 1-537 |
| Instruction Register and Decoding Logic | 1-537 |
| Boundary Scan Register (BSR) | 1-537 |
| Other Data Registers | 1-537 |
| POWER SAVING MODES | 1-538 |
| ACCESS OPERATIONS (SOFTWARE) | |
| I/O Resources | 1-538 |
| I/O Register Access | 1-538 |
| IEEE Address Access | 1-538 |
| Boot PROM Access | 1-538 |
| Static RAM Access | 1-538 |
| BUS CYCLES (HARDWARE) | 1-538 |
| Bus Master Mode | 1-539 |
| Refresh Cycles | 1-539 |
| Address PROM Cycles External PROM | 1-539 |
| Ethernet Controller Register Cycles | 1-540 |
| RESET Cycles | 1-540 |
| ISA Configuration Register Cycles | 1-540 |
| Boot PROM Cycles | 1-540 |
| Current Master Operation | 1-540 |
| Master Mode Memory Read Cycle | 1-541 |
| Master Mode Memory Write Cycle | 1-541 |
| Shared Memory Mode | 1-541 |
| Address PROM Cycles | 1-541 |
| Ethernet Controller Register Cycles | 1-542 |
| RESET Cycles | 1-542 |
| ISA Configuration Register Cycles | 1-542 |
| Boot PROM Cycles | 1-542 |
| Static RAM Cycles | 1-542 |
| TRANSMIT OPERATION | 1-544 |
| Transmit Function Programming | 1-544 |
| Automatic Pad Generation | 1-544 |
| Transmit FCS Generation | 1-544 |
| Transmit Exception Conditions | 1-544 |
| Loss of Carrier | 1-545 |
| RECEIVE OPERATION | 1-545 |
| Receive Function Programming | 1-545 |
| Automatic Pad Stripping | 1-546 |
| Receive FCS Checking | 1-546 |
| Receive Exception Conditions | 1-546 |
| LOOPBACK OPERATION | 1-547 |
| LEDs | 1-548 |
| PCnet-ISA⁺ CONTROLLER REGISTERS | 1-549 |
| REGISTER ACCESS | 1-549 |
| CONTROL AND STATUS REGISTERS | 1-549 |

| | |
|---|-------|
| CSR0: PCnet-ISA+ Controller Status Register | 1-549 |
| CSR1: IADR[15:0] | 1-551 |
| CSR2: IADR[23:16] | 1-551 |
| CSR3: Interrupt Masks and Deferral Control | 1-551 |
| CSR4: Test and Features Control | 1-552 |
| CSR6: RCV/XMT Descriptor Table Length | 1-554 |
| CSR8: Logical Address Filter, LADRF[15:0] | 1-554 |
| CSR9: Logical Address Filter, LADRF[31:16] | 1-554 |
| CSR10: Logical Address Filter, LADRF[47:32] | 1-554 |
| CSR11: Logical Address Filter, LADRF[63:48] | 1-554 |
| CSR12: Physical Address Register, PADR[15:0] | 1-554 |
| CSR13: Physical Address Register, PADR[31:16] | 1-555 |
| CSR14: Physical Address Register, PADR[47:32] | 1-555 |
| CSR15: Mode Register | 1-555 |
| CSR16: Initialization Block Address | 1-557 |
| CSR17: Initialization Block Address | 1-557 |
| CSR18–19: Current Receive Buffer Address | 1-557 |
| CSR20–21: Current Transmit Buffer Address | 1-557 |
| CSR22–23: Next Receive Buffer Address | 1-557 |
| CSR24–25: Base Address of Receive Ring | 1-557 |
| CSR26–27: Next Receive Descriptor Address | 1-558 |
| CSR28–29: Current Receive Descriptor Address | 1-558 |
| CSR30–31: Base Address of Transmit Ring | 1-558 |
| CSR32–33: Next Transmit Descriptor Address | 1-558 |
| CSR34–35: Current Transmit Descriptor Address | 1-558 |
| CSR36–37: Next Next Receive Descriptor Address | 1-558 |
| CSR38–39: Next Next Transmit Descriptor Address | 1-558 |
| CSR40–41: Current Receive Status and Byte Count | 1-558 |
| CSR42–43: Current Transmit Status and Byte Count | 1-558 |
| CSR44–45: Next Receive Status and Byte Count | 1-559 |
| CSR46: Poll Time Counter | 1-559 |
| CSR47: Polling Interval | 1-559 |
| CSR48–49: Temporary Storage | 1-559 |
| CSR50–51: Temporary Storage | 1-559 |
| CSR52–53: Temporary Storage | 1-559 |
| CSR54–55: Temporary Storage | 1-559 |
| CSR56–57: Temporary Storage | 1-560 |
| CSR58–59: Temporary Storage | 1-560 |
| CSR60–61: Previous Transmit Descriptor Address | 1-560 |
| CSR62–63: Previous Transmit Status and Byte Count | 1-560 |
| CSR64–65: Next Transmit Buffer Address | 1-560 |
| CSR66–67: Next Transmit Status and Byte Count | 1-560 |
| CSR68–69: Transmit Status Temporary Storage | 1-560 |
| CSR70–71: Temporary Storage | 1-560 |
| CSR72: Receive Ring Counter | 1-561 |
| CSR74: Transmit Ring Counter | 1-561 |
| CSR76: Receive Ring Length | 1-561 |

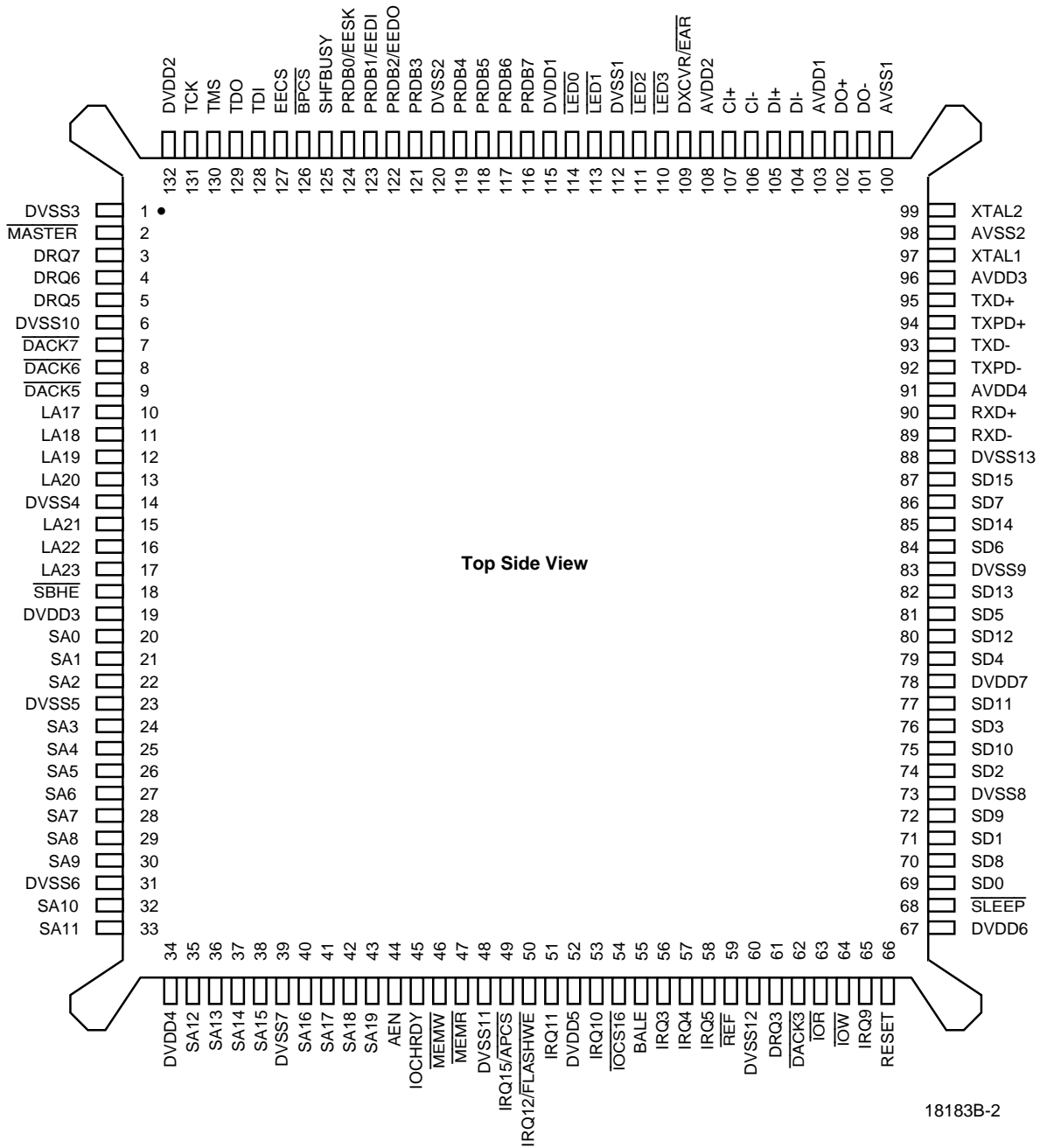
| | |
|--|--------------|
| CSR78: Transmit Ring Length | 1-561 |
| CSR80: Burst and FIFO Threshold Control | 1-561 |
| CSR82: Bus Activity Timer | 1-561 |
| CSR84–85: DMA Address | 1-562 |
| CSR86: Buffer Byte Counter | 1-563 |
| CSR88–89: Chip ID | 1-563 |
| CSR92: Ring Length Conversion | 1-563 |
| CSR94: Transmit Time Domain Reflectometry Count | 1-563 |
| CSR96–97: Bus Interface Scratch Register 0 | 1-563 |
| CSR98–99: Bus Interface Scratch Register 1 | 1-563 |
| CSR104–105: SWAP | 1-563 |
| CSR108–109: Buffer Management Scratch | 1-564 |
| CSR112: Missed Frame Count | 1-564 |
| CSR114: Receive Collision Count | 1-564 |
| CSR124: Buffer Management Unit Test | 1-564 |
| ISA BUS CONFIGURATION REGISTERS | 1-564 |
| ISACSR0: Master Mode Read Active | 1-565 |
| ISACSR1: Master Mode Write Active | 1-565 |
| ISACSR2: Miscellaneous Configuration | 1-565 |
| ISACSR3: EEPROM Configuration | 1-566 |
| ISACSR4: Link Integrity | 1-567 |
| ISACSR5: Default: RCV | 1-567 |
| ISACSR6: Default: RCVPOL | 1-568 |
| ISACSR7: Default: XMT | 1-568 |
| ISACSR8: Software Configuration (Read-Only Register) | 1-569 |
| INITIALIZATION BLOCK | 1-569 |
| RLEN and TLEN | 1-569 |
| RDRA and TDRA | 1-570 |
| LADRF | 1-570 |
| PADR | 1-570 |
| MODE | 1-570 |
| RECEIVE DESCRIPTORS | 1-571 |
| RMD0 | 1-571 |
| RMD1 | 1-571 |
| RMD2 | 1-572 |
| RMD3 | 1-572 |
| TRANSMIT DESCRIPTORS | 1-572 |
| TMD0 | 1-572 |
| TMD1 | 1-572 |
| TMD2 | 1-573 |
| TMD3 | 1-573 |
| REGISTER SUMMARY | 1-575 |
| SYSTEM APPLICATION | 1-578 |
| ISA BUS INTERFACE | 1-578 |
| Compatibility Consideration | 1-578 |

| | |
|---|-------|
| Bus Masters | 1-578 |
| Shared Memory | 1-578 |
| OPTIONAL ADDRESS PROM INTERFACE | 1-582 |
| BOOT PROM INTERFACE | 1-582 |
| STATIC RAM INTERFACE | 1-582 |
| AUI | 1-582 |
| EEPROM INTERFACE | 1-582 |
| 10BASE-T INTERFACE | 1-582 |
| ABSOLUTE MAXIMUM RATINGS | 1-584 |
| OPERATING RANGES | 1-584 |
| DC CHARACTERISTICS | 1-584 |
| SWITCHING CHARACTERISTICS | 1-587 |
| BUS MASTER MODE | 1-587 |
| SHARED MEMORY MODE | 1-591 |
| EADI | 1-595 |
| JTAG (IEEE 1149.1) INTERFACE | 1-595 |
| GPSI | 1-596 |
| AUI | 1-597 |
| 10BASE-T INTERFACE | 1-598 |
| SERIAL EEPROM INTERFACE | 1-598 |
| KEY TO SWITCHING WAVEFORMS | 1-699 |
| SWITCHING TEST CIRCUITS | 1-600 |
| SWITCHING WAVEFORMS | 1-602 |
| BUS MASTER MODE | 1-602 |
| SHARED MEMORY MODE | 1-612 |
| GPSI | 1-622 |
| EADI | 1-623 |
| JTAG (IEEE 1149.1) INTERFACE | 1-623 |
| AUI | 1-624 |
| 10BASE-T INTERFACE | 1-627 |
| APPENDIX A: PCnet-ISA+ COMPATIBLE MEDIA INTERFACE MODULES | |
| 10BASE-T FILTERS and TRANSFORMERS | 1-629 |
| AUI ISOLATION TRANSFORMERS | 1-629 |
| MANUFACTURER CONTACT INFORMATION | 1-630 |
| APPENDIX B: LAYOUT RECOMMENDATION FOR REDUCING NOISE | |
| DECOUPLING LOW-PASS RC FILTER DESIGN | 1-631 |
| APPENDIX C: SAMPLE CONFIGURATION FILE | 1-633 |
| APPENDIX D: ALTERNATIVE METHOD FOR INITIALIZATION | 1-635 |
| APPENDIX E: INTRODUCTION TO THE CONCEPT OF LOOK AHEAD PACKET PROCESSING (LAPP) | 1-636 |
| APPENDIX F: SOME CHARACTERISTICS OF THE XXC56 SERIAL EEPROMs | 1-646 |

BLOCK DIAGRAM: BUS MASTER MODE



CONNECTION DIAGRAM: BUS MASTER



PIN DESIGNATIONS: BUS MASTER
Listed by Pin Number

| Pin # | Name | Pin # | Name | Pin # | Name |
|-------|--------|-------|---------------|-------|------------|
| 1 | DVSS3 | 45 | IOCHRDY | 89 | RXD- |
| 2 | MASTER | 46 | MEMW | 90 | RXD+ |
| 3 | DRQ7 | 47 | MEMR | 91 | AVDD4 |
| 4 | DRQ6 | 48 | DVSS11 | 92 | TXPD- |
| 5 | DRQ5 | 49 | IRQ15/APCS | 93 | TXD- |
| 6 | DVSS10 | 50 | IRQ12/FlashWE | 94 | TXPD+ |
| 7 | DACK7 | 51 | IRQ11 | 95 | TXD+ |
| 8 | DACK6 | 52 | DVDD5 | 96 | AVDD3 |
| 9 | DACK5 | 53 | IRQ10 | 97 | XTAL1 |
| 10 | LA17 | 54 | IOCS16 | 98 | AVSS2 |
| 11 | LA18 | 55 | BALE | 99 | XTAL2 |
| 12 | LA19 | 56 | IRQ3 | 100 | AVSS1 |
| 13 | LA20 | 57 | IRQ4 | 101 | DO- |
| 14 | DVSS4 | 58 | IRQ5 | 102 | DO+ |
| 15 | LA21 | 59 | REF | 103 | AVDD1 |
| 16 | LA22 | 60 | DVSS12 | 104 | DI- |
| 17 | LA23 | 61 | DRQ3 | 105 | DI+ |
| 18 | SBHE | 62 | DACK3 | 106 | CI- |
| 19 | DVDD3 | 63 | IOR | 107 | CI+ |
| 20 | SA0 | 64 | IOW | 108 | AVDD2 |
| 21 | SA1 | 65 | IRQ9 | 109 | DXCVR/EAR |
| 22 | SA2 | 66 | RESET | 110 | LED3 |
| 23 | DVSS5 | 67 | DVDD6 | 111 | LED2 |
| 24 | SA3 | 68 | SLEEP | 112 | DVSS1 |
| 25 | SA4 | 69 | SD0 | 113 | LED1 |
| 26 | SA5 | 70 | SD8 | 114 | LED0 |
| 27 | SA6 | 71 | SD1 | 115 | DVDD1 |
| 28 | SA7 | 72 | SD9 | 116 | PRDB7 |
| 29 | SA8 | 73 | DVSS8 | 117 | PRDB6 |
| 30 | SA9 | 74 | SD2 | 118 | PRDB5 |
| 31 | DVSS6 | 75 | SD10 | 119 | PRDB4 |
| 32 | SA10 | 76 | SD3 | 120 | DVSS2 |
| 33 | SA11 | 77 | SD11 | 121 | PRDB3 |
| 34 | DVDD4 | 78 | DVDD7 | 122 | PRDB2/EEDO |
| 35 | SA12 | 79 | SD4 | 123 | PRDB1/EEDI |
| 36 | SA13 | 80 | SD12 | 124 | PRDB0/EESK |
| 37 | SA14 | 81 | SD5 | 125 | SHFBUSY |
| 38 | SA15 | 82 | SD13 | 126 | BPCS |
| 39 | DVSS7 | 83 | DVSS9 | 127 | EECS |
| 40 | SA16 | 84 | SD6 | 128 | TDI |
| 41 | SA17 | 85 | SD14 | 129 | TDO |
| 42 | SA18 | 86 | SD7 | 130 | TMS |
| 43 | SA19 | 87 | SD15 | 131 | TCK |
| 44 | AEN | 88 | DVSS13 | 132 | DVDD2 |

PIN DESIGNATIONS: BUS MASTER**Listed by Pin Name**

| Name | Pin # | Name | Pin # | Name | Pin # |
|--------------------------------|-------|----------------------------|-------|---------------------------|-------|
| AEN | 44 | EECS | 127 | SA13 | 36 |
| AVDD1 | 103 | IOCHRDY | 45 | SA14 | 37 |
| AVDD2 | 108 | $\overline{\text{IOCS16}}$ | 54 | SA15 | 38 |
| AVDD3 | 96 | $\overline{\text{IOR}}$ | 63 | SA16 | 40 |
| AVDD4 | 91 | $\overline{\text{IOW}}$ | 64 | SA17 | 41 |
| AVSS1 | 100 | IRQ10 | 53 | SA18 | 42 |
| AVSS2 | 98 | IRQ11 | 51 | SA19 | 43 |
| BALE | 55 | IRQ12/FlashWE | 50 | SA2 | 22 |
| BPCS | 126 | IRQ15/APCS | 49 | SA3 | 24 |
| CI- | 106 | IRQ3 | 56 | SA4 | 25 |
| CI+ | 107 | IRQ4 | 57 | SA5 | 26 |
| $\overline{\text{DACK3}}$ | 62 | IRQ5 | 58 | SA6 | 27 |
| $\overline{\text{DACK5}}$ | 9 | IRQ9 | 65 | SA7 | 28 |
| $\overline{\text{DACK6}}$ | 8 | LA17 | 10 | SA8 | 29 |
| $\overline{\text{DACK7}}$ | 7 | LA18 | 11 | SA9 | 30 |
| DI- | 104 | LA19 | 12 | $\overline{\text{SBHE}}$ | 18 |
| DI+ | 105 | LA20 | 13 | SD0 | 69 |
| DO- | 101 | LA21 | 15 | SD1 | 71 |
| DO+ | 102 | LA22 | 16 | SD10 | 75 |
| DRQ3 | 61 | LA23 | 17 | SD11 | 77 |
| DRQ5 | 5 | LED0 | 114 | SD12 | 80 |
| DRQ6 | 4 | LED1 | 113 | SD13 | 82 |
| DRQ7 | 3 | LED2 | 111 | SD14 | 85 |
| DVDD1 | 115 | LED3 | 110 | SD15 | 87 |
| DVDD2 | 132 | $\overline{\text{MASTER}}$ | 2 | SD2 | 74 |
| DVDD3 | 19 | $\overline{\text{MEMR}}$ | 47 | SD3 | 76 |
| DVDD4 | 34 | $\overline{\text{MEMW}}$ | 46 | SD4 | 79 |
| DVDD5 | 52 | PRDB0/EESK | 124 | SD5 | 81 |
| DVDD6 | 67 | PRDB1/EEDI | 123 | SD6 | 84 |
| DVDD7 | 78 | PRDB2/EEDO | 122 | SD7 | 86 |
| DVSS1 | 112 | PRDB3 | 121 | SD8 | 70 |
| DVSS10 | 6 | PRDB4 | 119 | SD9 | 72 |
| DVSS11 | 48 | PRDB5 | 118 | SHFBUSY | 125 |
| DVSS12 | 60 | PRDB6 | 117 | $\overline{\text{SLEEP}}$ | 68 |
| DVSS13 | 88 | PRDB7 | 116 | TCK | 131 |
| DVSS2 | 120 | $\overline{\text{REF}}$ | 59 | TDI | 128 |
| DVSS3 | 1 | RESET | 66 | TDO | 129 |
| DVSS4 | 14 | RXD- | 89 | TMS | 130 |
| DVSS5 | 23 | RXD+ | 90 | TXD- | 93 |
| DVSS6 | 31 | SA0 | 20 | TXD+ | 95 |
| DVSS7 | 39 | SA1 | 21 | TXPD- | 92 |
| DVSS8 | 73 | SA10 | 32 | TXPD+ | 94 |
| DVSS9 | 83 | SA11 | 33 | XTAL1 | 97 |
| DXCVR/ $\overline{\text{EAR}}$ | 109 | SA12 | 35 | XTAL2 | 99 |

PIN DESIGNATIONS: BUS MASTER
Listed by Group

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

PIN DESIGNATIONS: BUS MASTER (continued)**Listed by Group**

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

Output Driver Types

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

PIN DESCRIPTION: BUS MASTER MODE

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

IEEE P996 Terminology

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

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

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

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

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

ISA Interface

AEN

Address Enable *Input*

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

BALE

Used to latch the LA20–23 address lines.

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

DMA Acknowledge *Input*

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

DRQ 3, 5-7

DMA Request *Output*

When the PCnet-ISA+ controller needs to perform a DMA transfer, it asserts DRQ. The Permanent Master acknowledges DRQ with assertion of $\overline{\text{DACK}}$. When the PCnet-ISA+ controller does not need the bus it deasserts DRQ.

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

$\overline{\text{IOCHRDY}}$

I/O Channel Ready *Input/Output*

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

$\overline{\text{IOCS16}}$

I/O Chip Select 16 *Output*

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

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

$\overline{\text{IOR}}$

I/O Read *Input*

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

$\overline{\text{IOW}}$

I/O Write *Input*

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

IRQ 3, 4, 5, 9, 10, 11, 12, 15**Interrupt Request** *Output*

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

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

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

LA17-23**Unlatched Address Bus** *Input/Output*

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

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

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

MASTER**Master Mode** *Input/Output*

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

MEMR**Memory Read** *Input/Output*

MEMR goes LOW to perform a memory read operation.

MEMW**Memory Write** *Input/Output*

MEMW goes LOW to perform a memory write operation.

REF**Memory Refresh** *Input*

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

RESET**Reset** *Input*

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

SA0-19**System Address Bus** *Input/Output*

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

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

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

SD0-15**System Data Bus** *Input/Output*

These pins are used to transfer data to and from the PCnet-ISA⁺ controller to system resources via the ISA data bus. SD0-15 is driven by the PCnet-ISA⁺ controller when performing bus master writes and slave read operations. Likewise, the data on SD0-15 is latched by the PCnet-ISA⁺ controller when performing bus master reads and slave write operations.

Board Interface**IRQ12/FlashWE****Flash Write Enable** *Output*

Optional interface to the Flash memory boot PROM Write Enable.

IRQ15/APCS
Address PROM Chip Select *Output*

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

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

BPCS
Boot PROM Chip Select *Output*

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

DXCVR/ $\overline{\text{EAR}}$
**Disable Transceiver/
External Address Reject** *Input/Output*

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

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

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

LED0-3
LED Drivers *Output*

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

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

| LED | EADI Function |
|-----|---------------|
| 1 | SF/BD |
| 2 | SRD |
| 3 | SRDCLK |

PRDB3-7
Private Data Bus *Input/Output*

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

PRDB2/EEDO
Private data bus bit 2/Data Out *Input/Output*

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

PRDB1/EEDI
Private data bus bit 1/Data In *Input/Output*

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

PRDB0/EESK
**Private data bus bit 0/
Serial Clock** *Input/Output*

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

SHFBUSY
Input/Output

An output from PCnet-ISA⁺ which indicates that a read from the external EEPROM is in progress. It is active only when the hardware reconfigure is running (when data is being shifted out of the EEPROM due to a hardware RESET or the EELOAD command being issued). This pin should have a pull-up resistor (10 K Ω) to VCC.

EECS
EEPROM CHIPSELECT *Output*

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

SLEEP
Sleep *Input*

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

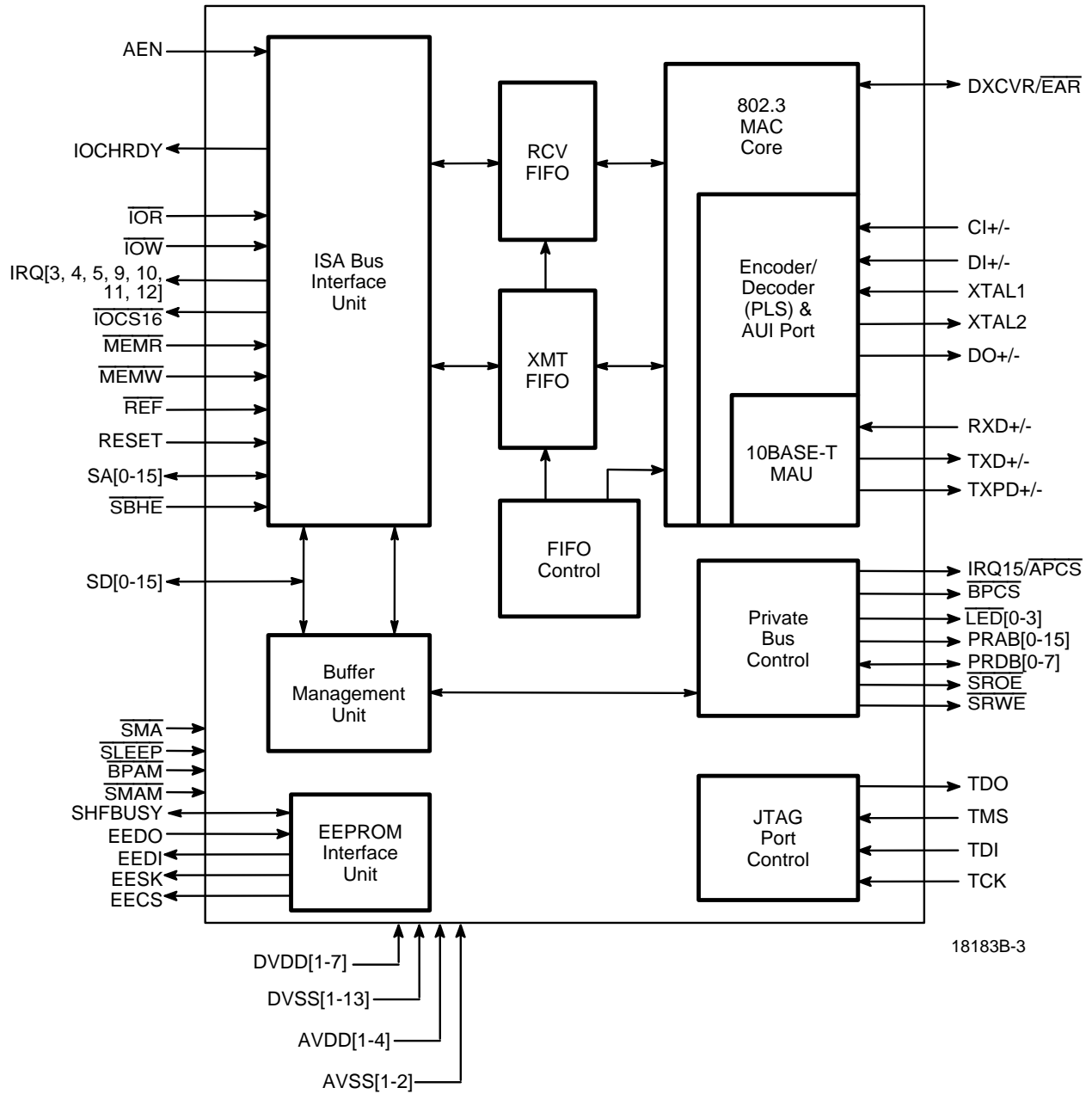
XTAL1
Crystal Connection *Input*

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

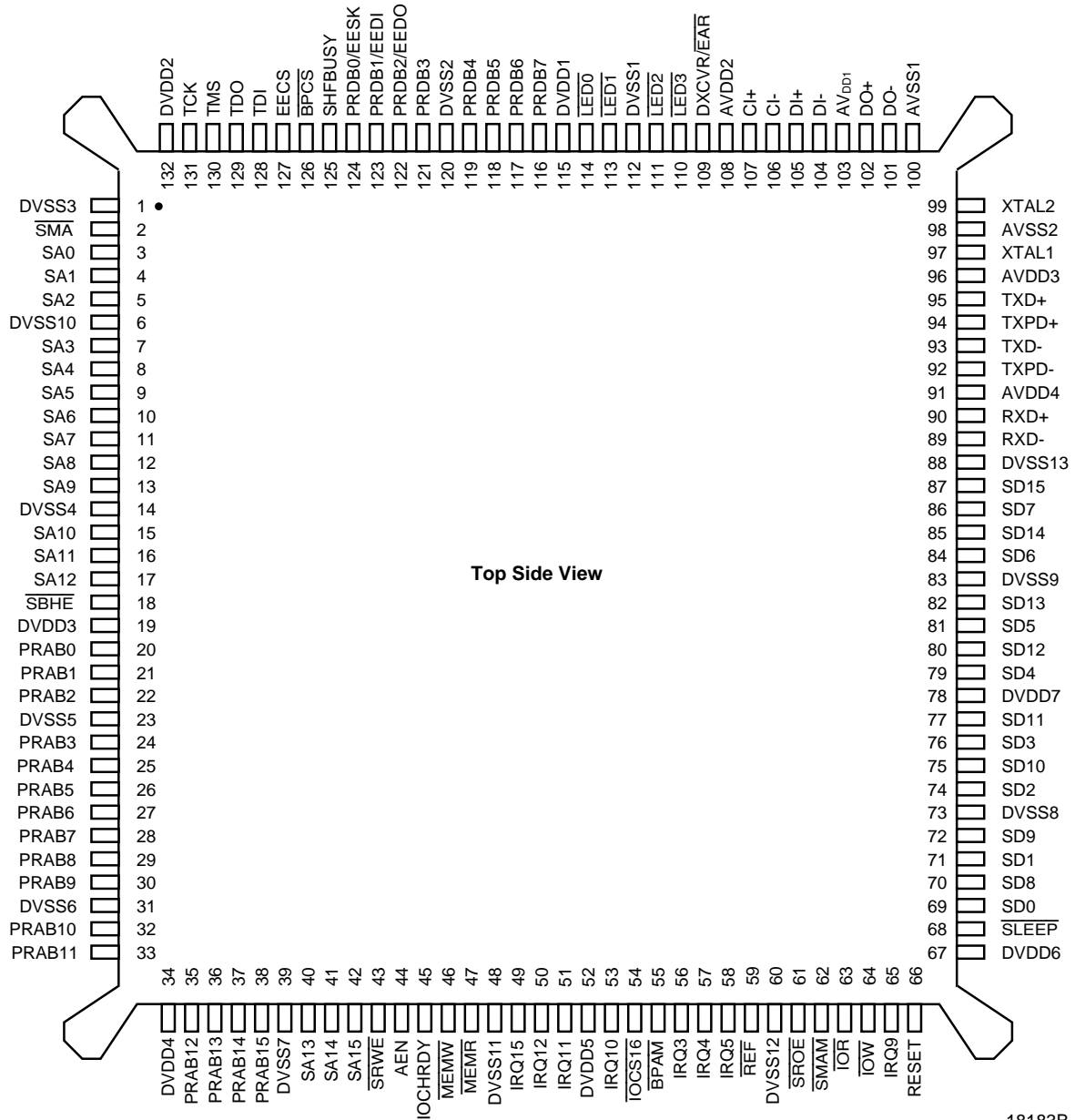
XTAL2
Crystal Connection *Output*

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

BLOCK DIAGRAM: SHARED MEMORY MODE



CONNECTION DIAGRAM: SHARED MEMORY



18183B-4

PIN DESIGNATIONS: SHARED MEMORY**Listed by Pin Number**

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

PIN DESIGNATIONS: SHARED MEMORY

Listed by Pin Name

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

PIN DESIGNATIONS: SHARED MEMORY**Listed by Group**

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

PIN DESIGNATIONS: SHARED MEMORY (continued)
Listed by Group

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

Output Driver Types

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

PIN DESCRIPTION: SHARED MEMORY MODE

ISA Interface

AEN

Address Enable *Input*

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

IOCHRDY

I/O Channel Ready *Output*

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

IOCS16

I/O Chip Select 16 *Input/Output*

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

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

$\overline{\text{IOR}}$

I/O Read *Input*

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

$\overline{\text{IOW}}$

I/O Write *Input*

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

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

Interrupt Request *Output*

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

which allows for suppression of IRQ assertion. These flags have the following meaning:

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

$\overline{\text{MEMR}}$

Memory Read *Input*

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

$\overline{\text{MEMW}}$

Memory Write *Input*

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

RESET

Reset *Input*

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

SA0-15

System Address Bus *Input*

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

$\overline{\text{SBHE}}$

System Bus High Enable *Input*

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

SD0-15

System Data Bus *Input/Output*

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

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

BOARD INTERFACE
APCS/IRQ15
Address PROM Chip Select *Output*

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

BPAM
Boot PROM Address Match *Input*

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

BPCS
Boot PROM Chip Select *Output*

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

DXCVR/ $\overline{\text{EAR}}$
**Disable Transceiver/
External Address Reject** *Input/Output*

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

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

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

LED0-3
LED Drivers *Output*

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

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

| LED | EADI Function |
|-----|---------------|
| 1 | SF/BD |
| 2 | SRD |
| 3 | SRDCLK |

PRAB0-15
Private Address Bus *Input/Output*

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

PRDB3-7
Private Data Bus *Input/Output*

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

PRDB2/EEDO
Private Data Bus Bit 2/Data Out *Input/Output*

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

PRDB1/EEDI
Private Data Bus Bit 1/Data In *Input/Output*

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

PRDB0/EESK
**Private Data Bus Bit 0/
Serial Clock** *Input/Output*

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

SHFBUSY
Shift Busy *Input/Output*

An output from PCnet-ISA+ which indicates that a read from the external EEPROM is in progress. It is active only when the hardware reconfigure is running (when data is being shifted out of the EEPROM due to a hardware RESET or the EELOAD command being issued).

SHFBUSY should be connected to V_{CC} with a 10K Ω resistor.

EECS
EEPROM CHIPSELECT *Output*

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

SLEEP
Sleep *Input*

When $\overline{\text{SLEEP}}$ input is asserted (active LOW), the PCnet-ISA+ controller performs an internal system reset

and proceeds into a power savings mode. All outputs will be placed in their normal reset condition. All PCnet-ISA+ controller inputs will be ignored except for the SLEEP pin itself. Deassertion of SLEEP results in wake-up. The system must delay the starting of the network controller by 0.5 seconds to allow internal analog circuits to stabilize.

SMA**Shared Memory Architecture** *Input*

This pin is sampled after the hardware RESET sequence. The pin must be pulled permanently LOW for operation in the shared memory mode.

SMAM**Shared Memory Address Match** *Input*

This pin indicates an access to shared memory when active. The type of access is decided by MEMR or MEMW.

SROE**Static RAM Output Enable** *Output*

This pin directly controls the external SRAM's \overline{OE} pin.

SRCS/IRQ12**Static RAM Chip Select** *Output*

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

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

SRWE/WE**Static RAM Write Enable/
Write Enable** *Output*

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

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

XTAL1**Crystal Connection** *Input*

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

XTAL2**Crystal Connection** *Output*

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

**PIN DESCRIPTION:
NETWORK INTERFACES**
AUI

CI+, CI-
Control Input *Input*

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

DI+, DI-
Data In *Input*

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

DO+, DO-
Data Out *Output*

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

Twisted Pair Interface

RXD+, RXD-
Receive Data *Input*

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

TXD+, TXD-
Transmit Data *Output*

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

TXP+, TXP-
Transmit Predistortion Control *Output*

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

**PIN DESCRIPTION:
IEEE 1149.1 (JTAG) TEST ACCESS PORT**

TCK
Test Clock *Input*

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

TDI
Test Data Input *Input*

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

TDO
Test Data Output *Output*

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

TMS
Test Mode Select *Input*

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

**PIN DESCRIPTION:
POWER SUPPLIES**

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

AVDD1-4
Analog Power (4 Pins) *Power*

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

AVSS1-2
Analog Ground (2 Pins) *Power*

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

DVDD1-7
Digital Power (7 Pins) *Power*

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

DVSS1-13
Digital Ground (13 Pins) *Power*

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

FUNCTIONAL DESCRIPTION

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

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

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

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

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

The PCnet-ISA+ controller's advanced bus mastering architecture also provides high data throughput and low CPU utilization for even better performance.

To offer greater flexibility, the PCnet-ISA+ controller has a shared memory mode to meet varying application needs. The shared memory architecture is compatible with very low-end machines, such as PC/XTs that do not support bus mastering, and very high end machines which require local packet buffering for increased system latency.

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

Bus Master Mode

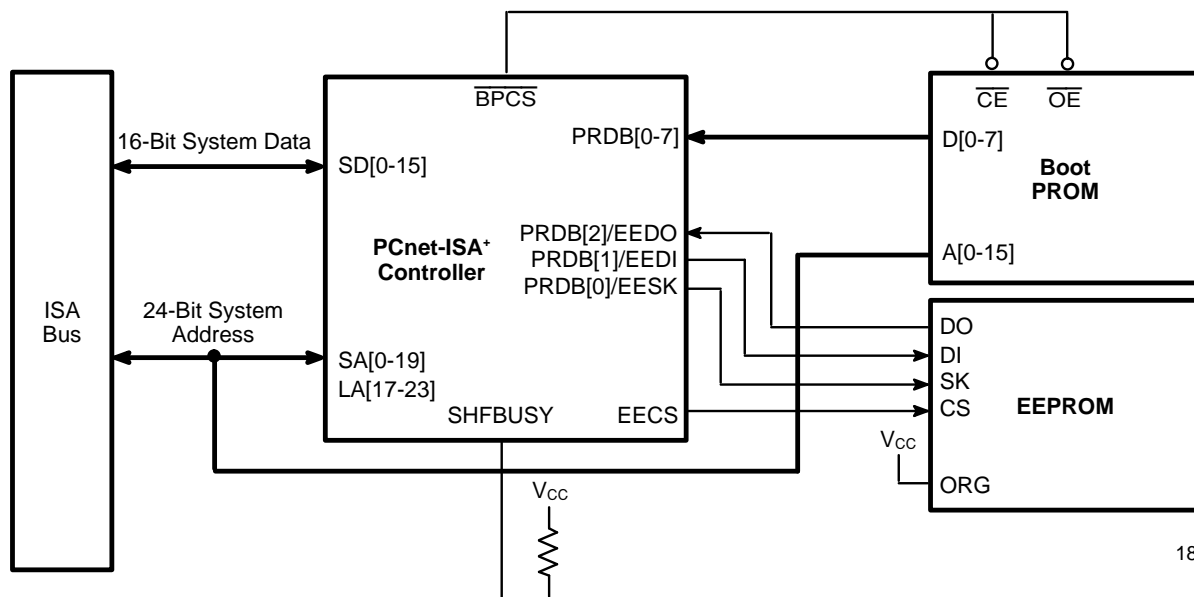
System Interface

The PCnet-ISA+ controller has two fundamental operating modes, Bus Master and Shared Memory. The selection of either the Bus Master mode or the Shared Memory mode must be done through hard wiring; it is not software configurable. The Bus Master mode provides an Am7990 (LANCE) compatible Ethernet controller, an Ethernet Address EEPROM or PROM, a Boot PROM, and a set of device configuration registers.

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

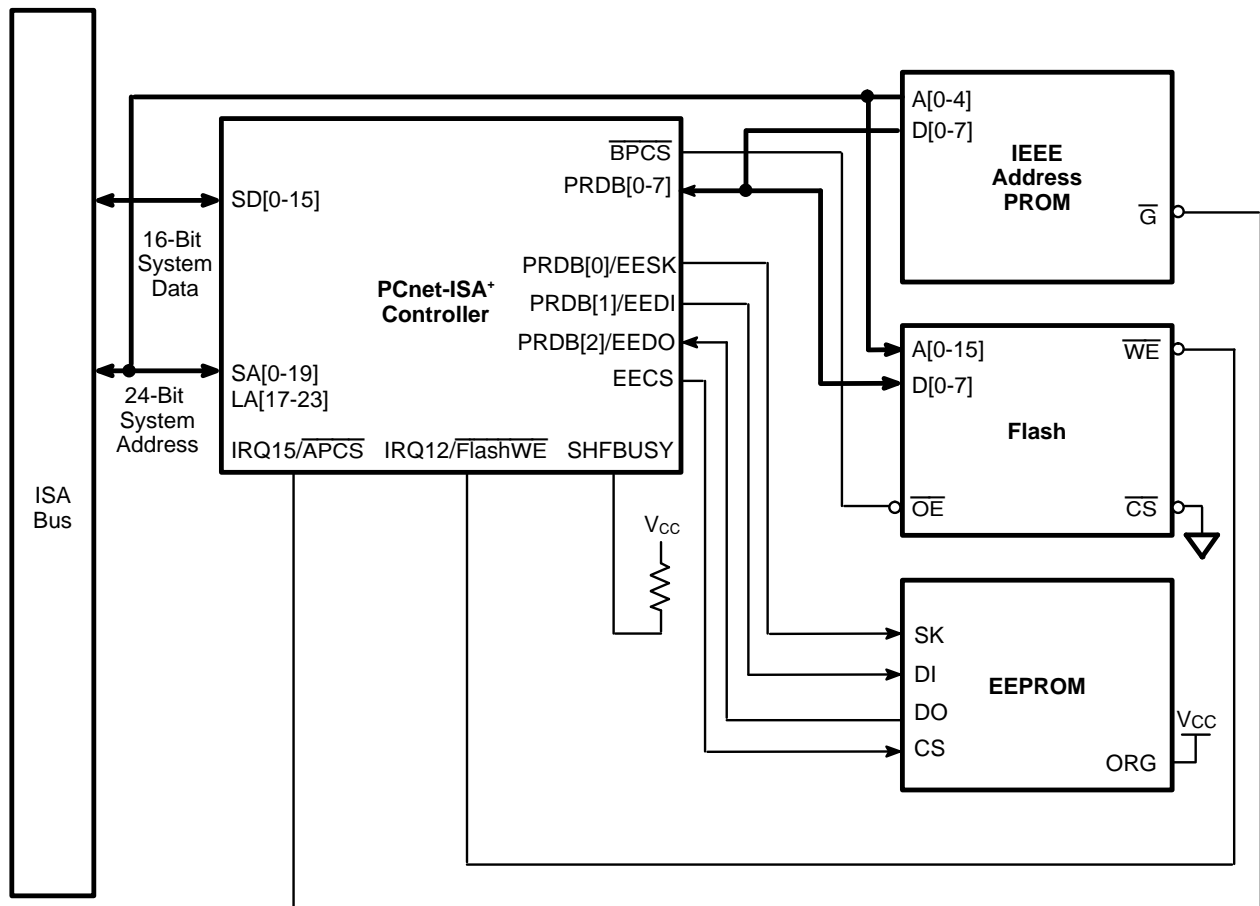
The address PROM, board configuration registers, and the Ethernet controller occupy 24 bytes of I/O space and can be located on 16 different starting addresses.

Data buffers are located in system memory and can be accessed by the PCnet-ISA+ controller when the device becomes the Current Master.



18183B-6

**Bus Master Block Diagram
Plug and Play Compatible**



18183B-7

**Bus Master Block Diagram
Plug and Play Compatible with Flash Support**

Shared Memory Mode

System Interface

The Shared Memory mode is the other fundamental operating mode available on the PCnet-ISA+ controller. The PCnet-ISA+ controller uses the same descriptor and buffer architecture as the LANCE, but these data structures are stored in static RAM controlled by the PCnet-ISA+ controller. The static RAM is visible as a memory resource to the PC. The other resources look the same as in the Bus Master mode.

The Boot PROM is selected by an external device which drives the Boot PROM Address Match (\overline{BPAM}) input to the PCnet-ISA+ controller. The PCnet-ISA+ controller can perform two 8-bit accesses from the 8-bit Boot PROM and presents 16-bits of data. The shared memory works the same way, with an external device generating Shared Memory Address Match and the PCnet-ISA+ controller performing the read or write and the 8 to 16-bit data conversion.

Converting shared memory accesses from 8-bit cycles to 16-bit cycles allows use of the much faster 16-bit cycle timing while cutting the number of bus cycles in half.

This raises performance to more than 400% of what could be achieved with 8-bit cycles. Converting boot PROM accesses to 16-bit cycles allows the two memory resources to be in the same 128 Kbyte block of memory without a clash between two devices with different data widths.

The PCnet-ISA+ controller uses an internal address comparator to perform SRAM prefetches on the Private Data Bus; the SA0-15 signals are used internally to determine whether a SRAM read cycle prefetch is a match or a miss.

Access to the Ethernet controller registers, board configuration registers, and Address PROM is done with on-chip address comparators.

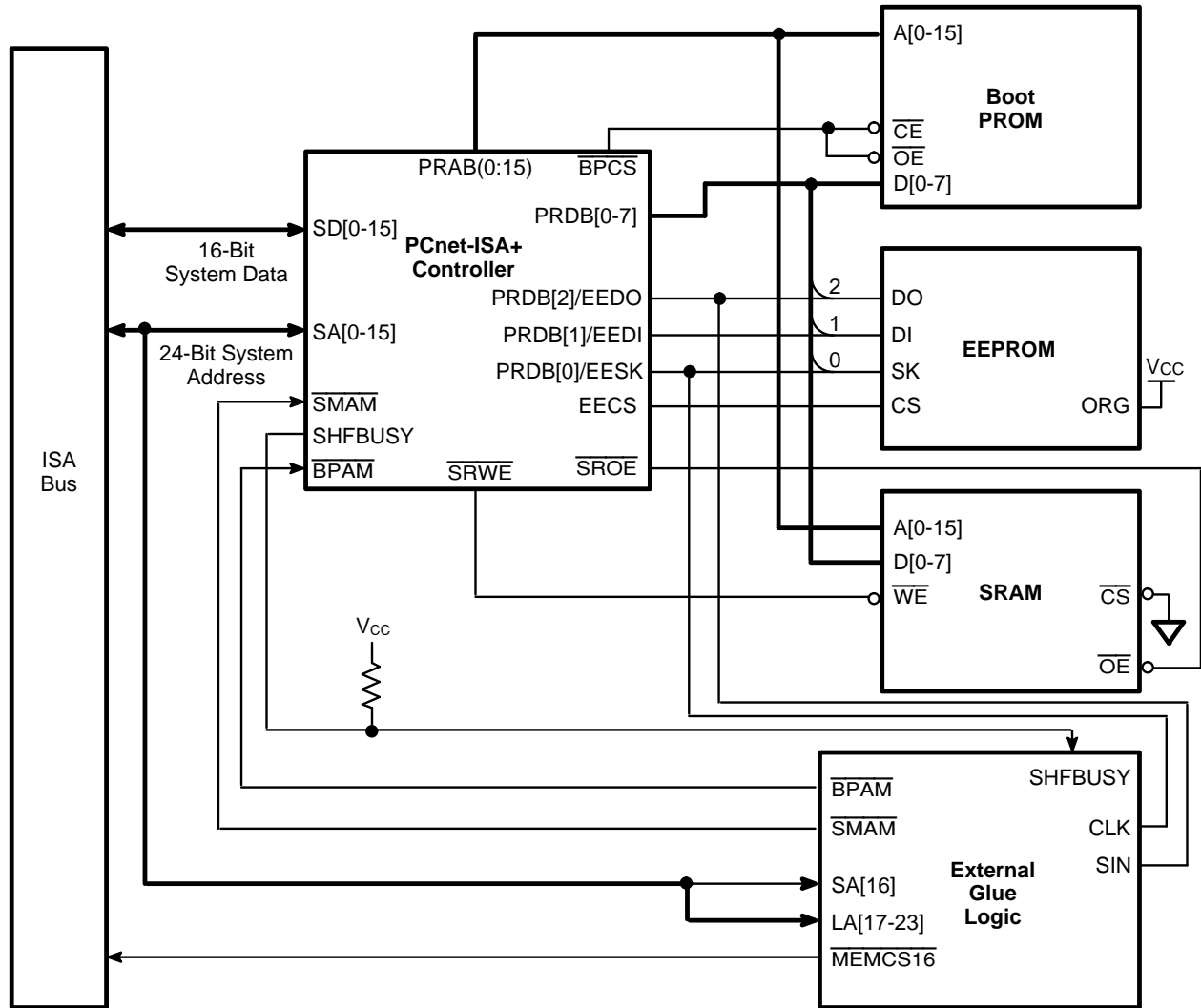
Network Interface

The PCnet-ISA+ controller can be connected to an IEEE 802.3 network via one of two network interface ports. The Attachment Unit Interface (AUI) provides an IEEE 802.3 compliant differential interface to a remote MAU or an on-board transceiver. The 10BASE-T interface provides a twisted-pair Ethernet port. The PCnet-ISA+ controller provides three modes of network interface

selection: automatic selection, software selection, and jumper selection of AUI or 10BASE-T interface.

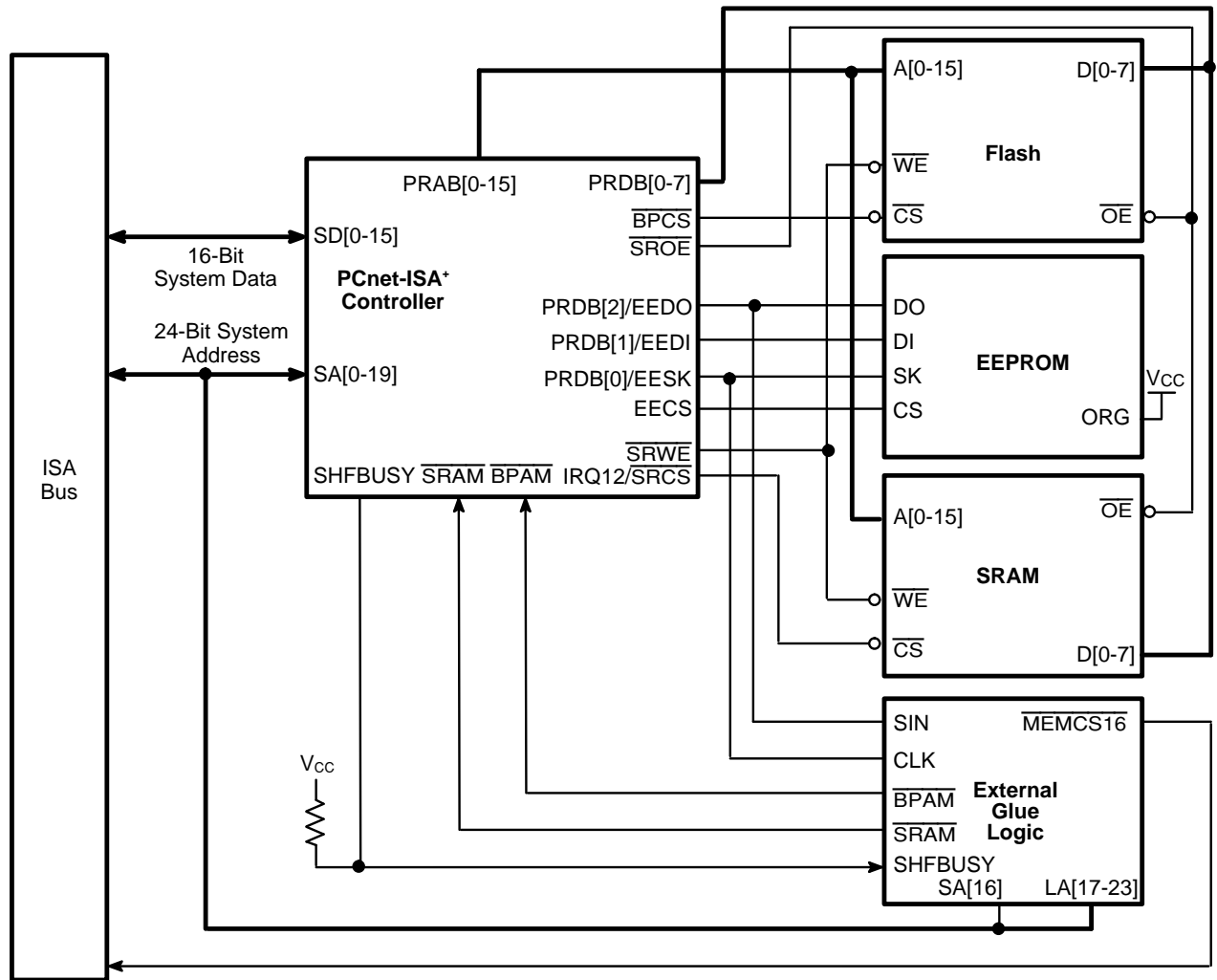
In the automatic selection mode, the PCnet-ISA+ controller will select the interface that is connected to the network by checking the Link Status state machine. If

both AUI and 10BASE-T interfaces are connected, the 10BASE-T interface is selected over AUI. If the PCnet-ISA+ controller is initialized for software selection of network interface, it will read the PORTSEL [1:0] bits in the Mode register (CSR15.8 and CSR15.7) to determine which interface needs to be activated.



18183B-9

Shared Memory Block Diagram
Plug and Play Compatible



18183B-10

Shared Memory Block Diagram
Plug and Play Compatible with Flash Memory Support

PLUG AND PLAY

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

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

Operation

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

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

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

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

Auto-Configuration Ports

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

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

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

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

ADDRESS PORT

The internal Plug and Play registers are accessed by writing the address to the ADDRESS PORT and then

either reading the READ_DATA PORT or writing to the WRITE_DATA PORT. Once the ADDRESS PORT has been written, any number of reads or writes can occur without having to rewrite the ADDRESS PORT.

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

WRITE_DATA PORT

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

READ_DATA PORT

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

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

Initiation Key

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

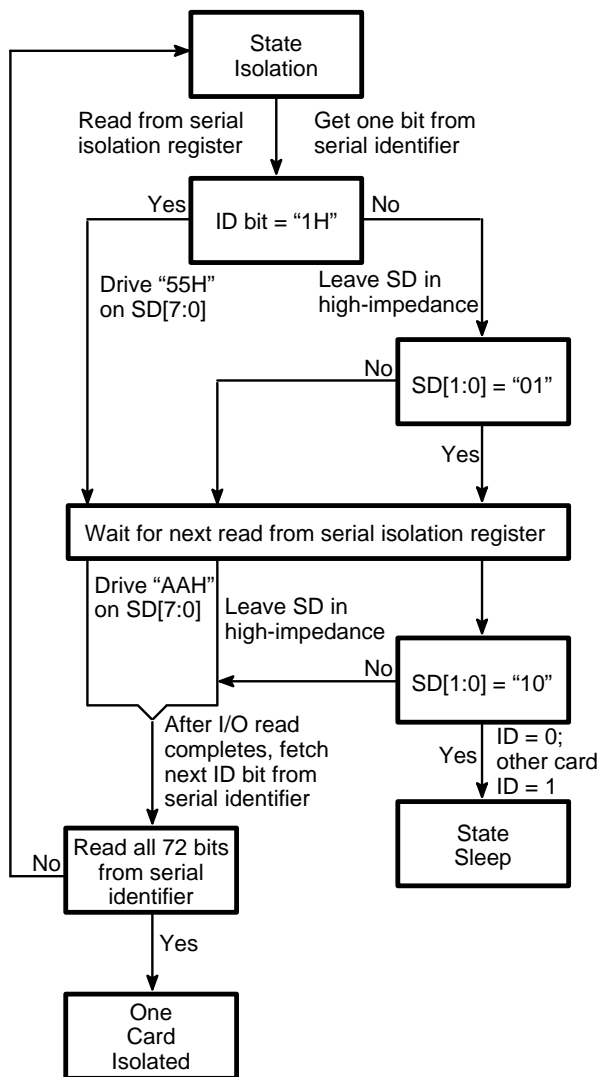
The initiation key places the PCnet-ISA+ device into the configuration mode. This is done by writing a predefined pattern to the ADDRESS PORT. If the proper sequence of I/O writes are detected by the PCnet-ISA+ device, the Plug and Play auto-configuration ports are enabled. This sequence must be sequential, i.e., any other I/O access to this I/O port will reset the state machine which is checking the pattern. Interrupts should be disabled during this time to eliminate any extraneous I/O cycles.

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

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

Isolation Protocol

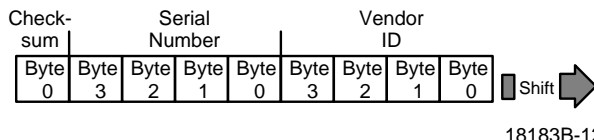
A simple algorithm is used to isolate each Plug and Play card. This algorithm uses the signals on the ISA bus and requires lock-step operation between the Plug and Play hardware and the isolation software.



18183B-11

Plug and Play ISA Card Isolation Algorithm

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



18183B-12

Shifting of Serial Identifier

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

Hardware Protocol

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

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

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

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

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

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

Cards must be assigned a CSN before they will respond to the other commands defined in the specification.

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

Software Protocol

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

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

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

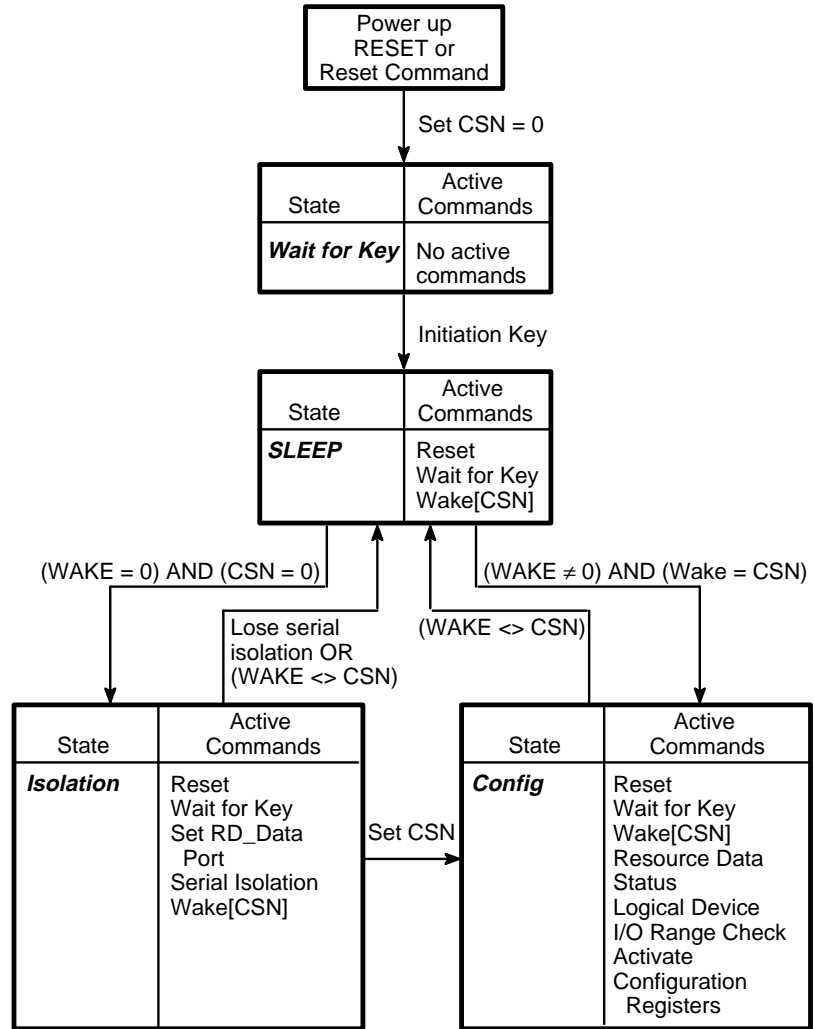
There are two other special considerations for the software protocol. During an iteration, it is possible that the 0x55 and 0xAA combination is never detected. It is also possible that the checksum does not match. If either of these cases occur on the first iteration, it must be assumed that the READ_DATA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

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

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

Plug and Play Card Control Registers

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



18183B-13

Notes

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

Plug and Play ISA Card State Transitions

Plug and Play Registers

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

Plug and Play Standard Registers

| Name | Address Port Value | Definition |
|-----------------------|--------------------|--|
| Set RD_DATA Port | 0x00 | Writing to this location modifies the address of the port used for reading from the Plug and Play ISA cards. Bits[7:00] become I/O read port address bits [9:02]. Reads from this register are ignored. I/O Address bits 11:10 should = 00, and 1:0 = 11. |
| Serial Isolation | 0x01 | A read to this register causes a Plug and Play card in the Isolation state to compare one bit of the board's ID. This process is fully described above. This register is read only. |
| Config Control | 0x02 | Bit[0] - Reset all logical devices and restore configuration registers to their power-up values. Bit[1] - Return to the Wait for Key state Bit[2] - Reset CSN to 0 A write to bit[0] of this register performs a reset function on all logical devices. This resets the contents of configuration registers to their default state. All card's logical devices enter their default state and the CSN is preserved. A write to bit[1] of this register causes all cards to enter the Wait for Key state but all CSNs are preserved and logical devices are not affected. A write to bit[2] of this register causes all cards to reset their CSN to zero. This register is write-only. The values are not sticky, that is, hardware will automatically clear them and there is no need for software to clear the bits. |
| Wake[CSN] | 0x03 | A write to this port will cause all cards that have a CSN that matches the write data[7:0] to go from the Sleep state to either the Isolation state if the write data for this command is zero or the Config state if the write data is not zero. This register is write-only. Writing to this register resets the EEPROM pointer to the beginning of the Plug and Play Data Structure. |
| Resource Data | 0x04 | A read from this address reads the next byte of resource information. The Status register must be polled until bit[0] is set before this register may be read. This register is read-only. |
| Status | 0x05 | Bit[0] when set indicates it is okay to read the next data byte from the Resource Data register. This register is read-only. |
| Card Select Number | 0x06 | A write to this port sets a card's CSN. The CSN is a value uniquely assigned to each ISA card after the serial identification process so that each card may be individually selected during a Wake [CSN] command. This register is read/write. |
| Logical Device Number | 0x07 | Selects the current logical device. This register is read only. The PCnet-ISA+ controller has only 1 logical device, and this register contains a value of 0x00 |

Plug and Play Logical Device Configuration Registers

The PCnet-ISA+ controller supports a subset of the defined Plug and Play logical device control registers. The reason for only supporting a subset of the registers

is that the PCnet-ISA+ controller does not require as many system resources as Plug and Play allows. For instance, Memory Descriptor 2 is not used, as the PCnet-ISA+ controller only requires two memory descriptors, one for the Boot PROM/Flash, and one for the SRAM in Shared Memory Mode.

Plug and Play Logical Device Control Registers

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

Memory Space Configuration

| Name | Register Index | Definition |
|---|----------------|---|
| Memory base address bits[23:16] descriptor 0 | 0x40 | Read/write value indicating the selected memory base address bits[23:16] for memory descriptor 0. This is the Boot Prom Space. |
| Memory base address bits[15:08] descriptor 0 | 0x41 | Read/write value indicating the selected memory base address bits[15:08] for memory descriptor 0. |
| Memory control | 0x42 | Bits[2:1] specifies 8/16-bit control. The encoding is identical to memory control (bits[4:3]) of the information field in the memory descriptor. Bit[0], =0, indicates the next field is used as a range length for decode (implies range length and base alignment of memory descriptor are equal). Bit[0] is read-only. |
| Memory upper limit address; bits[23:16] or range length; bits[23:16] for descriptor 0 | 0x43 | Read/write value indicating the selected memory high address bits[23:16] for memory descriptor 0. If bit[0] of memory control is 0, this is the range length. If bit[0] of memory control is 1, this is considered invalid. |
| Memory upper limit bits[15:08] or range length; bits[15:08] for descriptor 0 | 0x44 | Read/write value indicating the selected memory high address bits[15:08] for memory descriptor 0, either a memory address or a range length as described above. |
| Memory descriptor 1 | 0x48-0x4C | Memory descriptor 1. This is the SRAM Space for Shared Memory. |

I/O Space Configuration

| Name | Register Index | Definition |
|--|----------------|--|
| I/O port base address bits[15:08] descriptor 0 | 0x60 | Read/write value indicating the selected I/O lower limit address bits[15:08] for I/O descriptor 0. If a logical device indicates it only uses 10 bit encoding, then bits[15:10] do not need to be supported. |
| I/O port base address bits[07:00] descriptor 0 | 0x61 | Read/write value indicating the selected I/O lower limit address bits[07:00] for I/O descriptor 0. |

I/O Interrupt Configuration

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

DMA Channel Configuration

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

DETAILED FUNCTIONS

EEPROM

Interface

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

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

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

| | |
|---------------------------|-----------|
| IEEE address | 6 bytes |
| Reserved | 10 bytes |
| EISA ID | 4 bytes |
| ISACSRs | 12 bytes |
| Plug and Play Defaults | 19 bytes |
| 8-Bit Checksum | 1 byte |
| External Shift Chain | 2 bytes |
| Plug and Play Config Info | 192 bytes |

Serial EEPROM Byte Map

This byte map is for the case where a non-PCnet Family compatible software driver is implemented.

The following is a byte map of the XXC56 series of EEPROMs used by the PCnet-ISA+ Ethernet Controller.

| | | Word Location | |
|-----------------------------|---------------------------------|---------------|----------|
| IEEE Address (Bytes 0-5) | Byte 1 | Byte 0 | 0 |
| | Byte 3 | Byte 2 | 1 |
| | Byte 5 | Byte 4 | 2 |
| | Byte 7 | Byte 6 | 3 |
| | Byte 9 | Byte 8 | 4 |
| | Byte 11 | Byte 10 | 5 |
| | Byte 13 | Byte 12 | 6 |
| | Byte 15 | Byte 14 | 7 |
| EISA Config Reg. | EISA Byte 1 | EISA Byte 0 | 8 |
| | EISA Byte 3 | EISA Byte 2 | 9 |
| Internal Registers | MSRDA, ISACSR0 | | A |
| | MSWRA, ISACSR1 | | B |
| | MISC Config, ISACSR2 | | C |
| | LED1 Config, ISACSR5 | | D |
| | LED2 Config, ISACSR6 | | E |
| | LED3 Config, ISACSR7 | | F |
| | Plug and Play Reg. | PnP 0x61 | PnP 0x60 |
| PnP 0x71 | | PnP 0x70 | 11 |
| Unused | | PnP 0x74 | 12 |
| PnP 0x41 | | PnP 0x40 | 13 |
| PnP 0x43 | | PnP 0x42 | 14 |
| Unused | | PnP 0x44 | 15 |
| PnP 0x49 | | PnP 0x48 | 16 |
| PnP 0x4b | | PnP 0x4A | 17 |
| Unused | | PnP 0x4C | 18 |
| 8-bit Checksum | | PnP 0xF0 | 19 |
| External Shift Chain | | 1A | |
| Unused Locations | | 1B ⋮ 1F | |
| See Appendix C | Plug and Play Starting Location | | 20 |

Note:

Checksum is calculated on words 0 through 0x1Ah (first 54 Bytes).

Serial EEPROM Byte Map

The following is a byte map of the XXC56 series of EEPROMs used by the PCnet-ISA+ Ethernet Controller.

This byte map is for the case where a PCnet Family compatible software driver is implemented.

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

| Word Location | Byte 1 | Byte 0 | |
|-----------------------|---------------------------------|------------------|----------------------------|
| 0 | Byte 1 | Byte 0 | } IEEE Address (Bytes 0–5) |
| 1 | Byte 3 | Byte 2 | |
| 2 | Byte 5 | Byte 4 | |
| 3 | Reserved | Reserved | |
| 4 | HWID (01H) | Reserved | |
| 5 | User Space 1 | | |
| 6 | 16-Bit Checksum 1 | | |
| 7 | ASCII W(0 x 57H) | ASCII W(0 x 57H) | |
| 8 | EISA Byte 1 | EISA Byte 0 | |
| EISA Config Reg. 9 | EISA Byte 3 | EISA Byte 2 | |
| A | MSRDA, ISACSR0 | | |
| B | MSWRA, ISACSR1 | | |
| C | MISC Config, ISACSR2 | | |
| Internal Registers D | LED1 Config, ISACSR5 | | |
| E | LED2 Config, ISACSR6 | | |
| F | LED3 Config, ISACSR7 | | |
| 10 | PnP 0x61 | PnP 0x60 | I/O Ports |
| 11 | PnP 0x71 | PnP 0x70 | Interrupts |
| 12 | Unused | PnP 0x74 | DMA Channels |
| Plug and Play Reg. 13 | PnP 0x41 | PnP 0x40 | ROM Memory |
| 14 | PnP 0x43 | PnP 0x42 | |
| 15 | Unused | PnP 0x44 | |
| 16 | PnP 0x49 | PnP 0x48 | RAM Memory |
| 17 | PnP 0x4b | PnP 0x4A | |
| 18 | Unused | PnP 0x4C | |
| 19 | 8-bit Checksum 2 | PnP 0xF0 | Vendor Byte |
| 1A | External Shift Chain | | |
| 1B ⋮ 1F | Unused Locations | | |
| See Appendix C 20 | Plug and Play Starting Location | | See Appendix C |

Note:

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

Checksum 2 is calculated on words 0 through 0x1Ah (first 54 Bytes).

Plug and Play Register Map

The following chart and its bit descriptions show the internal configuration registers associated with the Plug

and Play operation. These registers control the configuration of the PCnet-ISA+ controller.

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

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

The following chart and its bit descriptions show the internal command registers associated with the Plug and

Play operation. These registers control the PCnet-ISA+ controller Plug and Play operation.

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

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

IOAM[3:0]

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

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

IRQ[3:0]

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

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

IRQ_TYPE

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

IRQ_LVL

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

DMA[2:0] DMA Channel Select (PnP 0x74). Controls the DRQ and DMA selection of PCnet-ISA+. The DMA[2:0] register will be written with a value from the EEPROM. {For Bus Master Mode Only} The DRQ signal will not be driven unless EE_VALID is set or Non-EEPROM sequential write process is complete.

| DMA[2:0] | DMA Channel (DRQ/DACK Pair) |
|----------|-----------------------------|
| 0 1 1 | Channel 3 |
| 1 0 1 | Channel 5 |
| 1 1 0 | Channel 6 |
| 1 1 1 | Channel 7 |

BPAM[3:0] Boot PROM Address Match to bits [23:16] of SA bus (PnP 0x40–0x41). Selects the location where the Boot PROM Address match decode is started. The BPAM will be written with a value from the EEPROM.

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

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

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

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

SRAM[3:0] Static RAM Address Match to bits [16:13] of SA bus (PnP 0x48–0x49). Selects the starting location of the Shared memory by using SA[16:13] for performing address comparisons. The shared memory address match, the **SMAM** is asserted low. SRAM[3] value must reflect the external address match logic for SA[16].

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

SR_16B Static RAM 16-bit access (PnP 0x4A). Asserted if SRAM cycles should respond as an 16-bit device.

SRSZ[3:0] Static RAM Size (PnP 0x4B–0x4C). Selects the size of the static RAM selected.

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

Vendor Defined Byte (PnP 0x0F)

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

| | |
|----------|--|
| | selection will respond as an 8-bit port. |
| AEN_CS | External Decode Logic for I/O Registers. When written with a one, the PCnet-ISA+ will use the AEN pin as I/O chip select bar, to allow for external decode logic for the upper address bit of SA [9:5]. The purpose of this pin is to allow I/O locations, not supported with the IOAM[3:0], selection, to be defined outside the range 0x200–0x3F7. When set to a zero, (Default), I/O Selection will use IOAM[3:0]. |
| APROM_EN | External Parallel IEEE Address PROM. When set, the IRQ15 pin is reconfigured to be an Address Chip Select low, similar to APCS pin in the existing PCnet-ISA (Am79C960) device. The purpose of this bit is to allow for both a serial EEPROM and parallel PROM to coexist. When APROM_EN is set, the IEEE address located in the serial EEPROM will be ignored and parallel access will occur over the PRDB bus. When APROM_EN is cleared, default state, the IEEE address will be read in from the serial device and written to an internal RAM. When the I/O space of the IEEE PROM is selected, PCnet-ISA+, will access the contents of this RAM for I/O read cycles. I/O write cycles will be ignored. |
| BP_CS | Boot PROM Chip Select. When BP_CS is set to one, BALE will act as an external chip select (active low) above bit 15 of the address bus. BALE = 0, will select the boot PROM when MEMR is asserted low if the BP_CS bit is set and BPAM[2:0] match SA[15:13] and BPSZ[3:0] matches the selected size. When BP_CS is set to zero. BALE will act as the normal address latch strobe to capture the upper address bits for memory access to the boot PROM. BP_CS is by default low. The primary purpose of this bit is to allow non-ISA bus applications to support larger Boot PROMS or non-standard Boot PROM/Flash locations. |
| FL_SEL | Flash Memory Device Selected. When set, the Boot PROM is replaced with an external Flash |

memory device. In Bus Master Mode, \overline{BPCS} is replaced with $\overline{Flash_OE}$. IRQ12 becomes $\overline{Flash_WE}$. The Flash's CS pin is grounded. In shared memory mode, \overline{BPCS} is replaced with $\overline{Flash_RAM_CS}$. IRQ12 becomes $\overline{Static_RAM_CS}$ pin. The SROE and SRWE signals are connected to both the SRAM and Flash memory devices. FL_SEL is cleared by a reset, which is the default.

Shared Memory Configuration Bits (Not Defined for Bus Master Mode)

In Shared Memory Mode, the address comparison above the 15th bit must be performed by external logic. All address comparisons for bit 15th and below will use the internal compare logic.

SRAM[3:0],

SR_16B, SRSZ[3:0] These are not defined in bus-master mode. BP_16B must be written with a zero in bus-master mode.

Note: In Bus Master Mode, the BP_16B is always considered an 8-bit device. If \overline{SBHE} signal is left unconnected, in shared memory mode (i.e. 8-bit Slot), all memory and I/O access will assume 8-bit accesses. It is the responsibility of external logic to drive $\overline{MEMCS16}$ signal for the appropriate 128 Kbit segment decoded from the LA[23:17] signals. $\overline{MEMCS16}$ should be driven when accessing an 8-bit memory resource.

Checksum Failure

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

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

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

Use Without EEPROM

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

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

External Scan Chain

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

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

For more information on the use of this function, please refer to the technical reference manual.

Flash PROM

Use

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

Interface

To use a Flash-type device with the PCnet-ISA⁺ controller, Flash Select is set in register 0F0h of the Plug and Play registers. Flash Select is cleared by RESET (default).

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

In shared memory mode, \overline{BPCS} becomes $\overline{Flash_CS}$ and IRQ12 becomes the static RAM Chip Select, and

the \overline{SROE} and \overline{SRWE} signals are connected to both the SRAM and Flash devices.

Optional IEEE Address PROM

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

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

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

EISA Configuration Registers

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

Bus Interface Unit (BIU)

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

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

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

DMA Transfers

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

1. Initialization Block DMA Transfers

Once the BIU has been granted bus mastership, it will perform four data transfer cycles (eight bytes) before relinquishing the bus. The four transfers within the mastership period will always be read cycles to contiguous addresses. There are 12 words to transfer so there will be three bus mastership periods.

2. Descriptor DMA Transfers

Once the BIU has been granted bus mastership, it will perform the appropriate number of data transfer cycles before relinquishing the bus. The transfers within the mastership period will always be of the same type (either all read or all write), but may be to non-contiguous addresses. Only the bytes which need to be read or written are accessed.

3. Burst-Cycle DMA Transfers

Once the BIU has been granted bus mastership, it will perform a series of consecutive data transfer cycles before relinquishing the bus. Each data transfer will be performed sequentially, with the issue of the address, and the transfer of the data with appropriate output signals to indicate selection of the active data bytes during the transfer. All transfers within the mastership cycle will be either read or write cycles, and will be to contiguous addresses. The number of data transfer cycles within the burst is dependent on the programming of the DMAPLUS option (CSR4, bit 14).

If DMAPLUS = 0, a maximum of 16 transfers will be performed. This may be changed by writing to the burst register (CSR80), but the default takes the same amount of time as the Am2100 family of LANCE-based boards, a little over 5 μ s.

If DMAPLUS = 1, the burst will continue until the FIFO is filled to its high threshold (32 bytes in transmit operation) or emptied to its low threshold (16 bytes in receive operation). The exact number of transfer cycles in this case will be dependent on the latency of the system bus to the BIU's mastership request and the speed of bus operation.

Buffer Management Unit (BMU)

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

Initialization

PCnet-ISA+ controller initialization includes the reading of the initialization block in memory to obtain the operating parameters. The initialization block is read when the INIT bit in CSR0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure correct operation. Four words at a time are read and the bus is released at the end of each block of reads, for a total of three arbitration cycles. Once the initialization block has been read in and processed, the BMU knows where the receive and transmit descriptor rings are. On completion of the read operation and after internal registers have been updated, IDON will be set in CSR0, and an interrupt generated if IENA is set.

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

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

Reinitialization

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

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

Buffer Management

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

Descriptor Rings

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

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

Each ring entry contains the following information:

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

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

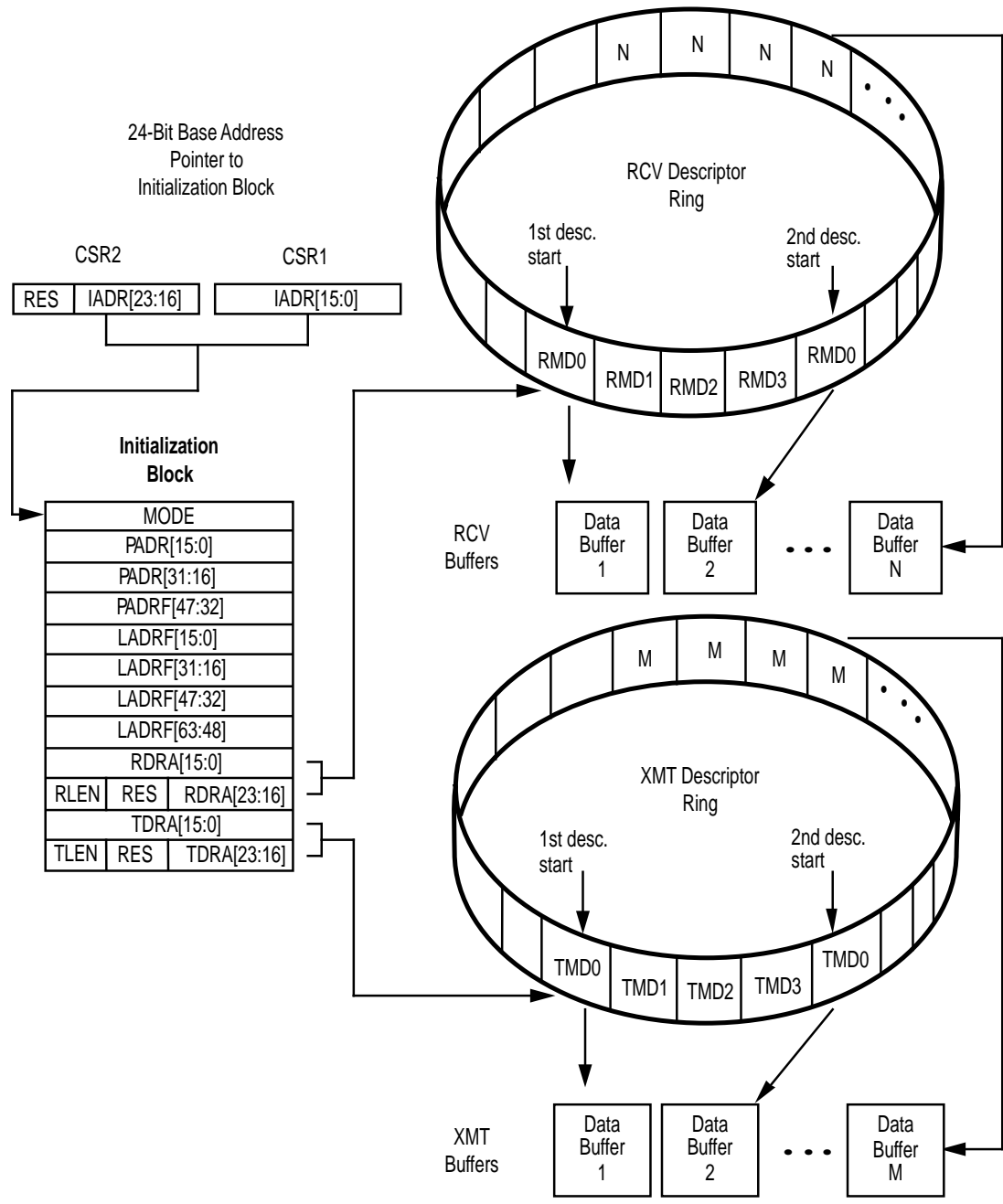
To permit the queuing and de-queuing of message buffers, ownership of each buffer is allocated to either the PCnet-ISA+ controller or the host. The OWN bit within the descriptor status information, either TMD or RMD (see section on TMD or RMD), is used for this purpose. “Deadly Embrace” conditions are avoided by the ownership mechanism. Only the owner is permitted to

relinquish ownership or to write to any field in the descriptor entry. A device that is not the current owner of a descriptor entry cannot assume ownership or change any field in the entry.

Descriptor Ring Access Mechanism

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

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



18183B-14

Initialization Block and Descriptor Rings

Polling

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

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

TMD1 and TMD0 of the current TDTE at periodic polling intervals. All information collected during polling activity will be stored internally in the appropriate CSRs. (i.e. CSR18–19, CSR40, CSR20–21, CSR42, CSR50, CSR52). Unowned descriptor status will be internally ignored.

A typical receive poll occurs under the following conditions:

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

or

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

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

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

A typical transmit poll occurs under the following conditions:

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

or

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

or

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

The poll time interval is nominally defined as 32,768 crystal clock periods, or 1.6 ms. However, the poll time register is controlled internally by microcode, so any other microcode controlled operation will interrupt the incrementing of the poll count register. For example, when a receive packet is accepted by the PCnet-ISA+ controller, the device suspends execution of the poll-time-incrementing microcode so that a receive microcode routine may instead be executed. Poll-time-incrementing code is resumed when the receive operation has completely finished. Note, however, that following the completion of any receive or transmit operation, a poll operation will always be performed. The

poll time count register is never reset. Note that if a non-default is desired, then a strict sequence of setting the INIT bit in CSR0, waiting for the IDON bit in CSR0, then writing to CSR47, and then setting STRT in CSR0 must be observed, otherwise the default value will not be overwritten. See the CSR47 section for details.

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

Transmit Descriptor Table Entry (TDTE)

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

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

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

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

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

If the PCnet-ISA+ controller does not own the next TDTE (i.e. the second TDTE for this packet), then it will complete transmission of the current buffer and then update the status of the current (first) TDTE with the BUFF and UFLO bits being set. This will cause the transmitter to be disabled (CSR0, TXON=0). The PCnet-ISA+ controller will have to be restarted to restore the transmit function. The situation that matches this description implies that the system has not been able to stay ahead of the

PCnet-ISA+ controller in the transmit descriptor ring and therefore, the condition is treated as a fatal error. To avoid this situation, the system should always set the transmit chain descriptor own bits in reverse order.

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

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

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

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

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

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

operation poll avoids inserting poll time counts between successive transmit packets.

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

Receive Descriptor Table Entry (RDTE)

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

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

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

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

If the PCnet-ISA+ controller sees that the last poll (either a normal poll or the last-ditch effort described in the above paragraph) of the current RDTE shows valid ownership, then it proceeds to a poll of the next RDTE. Following this poll, and regardless of the outcome of this poll, transfers of receive data from the FIFO may begin.

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

If the packet length exceeds the length of the first (current) buffer, and the PCnet-ISA+ controller does own the

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

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

Media Access Control

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

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).

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

The two primary attributes of the MAC engine are:

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

Transmit And Receive Message Data Encapsulation

The MAC engine provides minimum frame size enforcement for transmit and receive packets. When

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

Framing (frame boundary delimitation, frame synchronization)

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

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

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

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

Addressing (source and destination address handling)

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

Error detection (physical medium transmission errors).

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

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

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

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

The status of each receive message is available in the appropriate RMD and CSR areas. FCS and Framing errors (FRAM) are reported, although the received frame is still passed to the host. The FRAM error will only be reported if an FCS error is detected and there are a non-integral number of bits in the message. The MAC engine

will ignore up to seven additional bits at the end of a message (dribbling bits), which can occur under normal network operating conditions. The reception of eight additional bits will cause the MAC engine to de-serialize the entire byte, and will result in the received message and FCS being modified.

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

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

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

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

Media Access Management

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

Medium allocation (collision avoidance)

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

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

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

Note: It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the interpacket gap based on this indication it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in 4.2.8, are recommended when InterFrameSpacingPart1 is other than zero:

- (1) Upon completing a transmission, start timing the interpacket gap, as soon as transmitting and carrierSense are both false.
- (2) When timing an interpacket gap following reception, reset the interpacket gap timing if carrier Sense becomes true during the first 2/3 of the interpacket gap timing interval. During the final 1/3 of the interval the timer shall not be reset to ensure fair access to the medium. An initial period shorter than 2/3 of the interval is permissible including zero."

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

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

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

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

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

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

The PCnet-ISA+ controller implements a carrier sense "blinding" period within 0 - 4.0 μ s from de-assertion of carrier sense after transmission. This effectively means that when transmit two part deferral is enabled (DXMT2PD is cleared) the IFS1 time is from 4 μ s to 6 μ s after a transmission. However, since IPG shrinkage below 4 μ s will rarely be encountered on a correctly configured network, and since the fragment size will be larger than the 4 μ s blinding window, then the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the PCnet-ISA+ controller will defer its transmission. In addition, the PCnet-ISA+ controller will not restart the "blinding" period if carrier is detected within the 4.0 μ s – 6.0 μ s IFS1 period, but will commence timing of the entire IFS1 period.

Contention resolution (collision handling)

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

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

The MAC Engine will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of collision will cause the transmission to be re-scheduled, dependent on the backoff time that the MAC Engine computes. If a single retry was required, the ONE bit will be set in the Transmit Frame Status (TMD1 in the Transmit Descriptor Ring). If more than one retry was

required, the MORE bit will be set. If all 16 attempts experienced collisions, the RTRY bit (in TMD2) will be set (ONE and MORE will be clear), and the transmit message will be flushed from the FIFO. If retries have been disabled by setting the DRTY bit in the MODE register (CSR15), the MAC Engine will abandon transmission of the frame on detection of the first collision. In this case, only the RTRY bit will be set and the transmit message will be flushed from the FIFO.

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

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

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

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

$$0 \leq r < 2^k, \text{ where } k = \min(n, 10).”$$

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

Manchester Encoder/Decoder (MENDEC)

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

External Crystal Characteristics

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

External Crystal Characteristics

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

* Requires trimming crystal spec; no trim is 50 ppm total

External Clock Drive Characteristics

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

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

MENDEC Transmit Path

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

Transmitter Timing and Operation

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

controller and Internal Transmit Enable (ITXEN). The internal transmit clock is also used as a stable bit-rate clock by the receive section of the MENDEC and controller.

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

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

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

Receive Path

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

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

Input Signal Conditioning

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

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

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

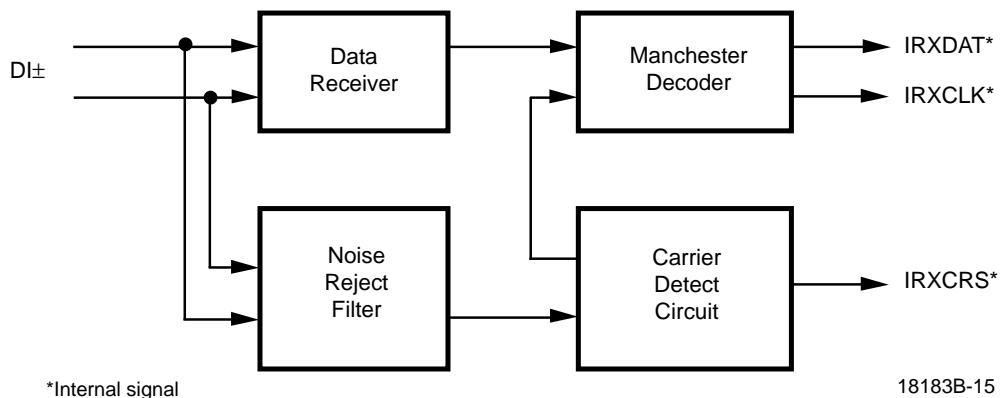
Clock Acquisition

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

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

PLL Tracking

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



Receiver Block Diagram

18183B-15

Carrier Tracking and End of Message

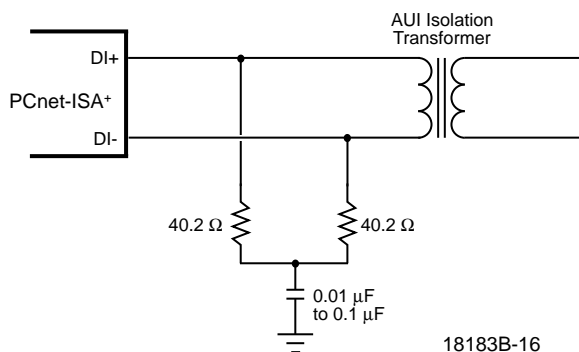
The carrier detection circuit monitors the DI± inputs after IRXCRS is asserted for an end of message. IRXCRS de-asserts 1 to 2 bit times after the last positive transition on the incoming message. This initiates the end of reception cycle. The time delay from the last rising edge of the message to IRXCRS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

Data Decoding

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

Differential Input Terminations

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



Differential Input Termination

Collision Detection

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

Jitter Tolerance Definition

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

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

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

Attachment Unit Interface (AUI)

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

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

Twisted Pair Transceiver (T-MAU)

The T-MAU implements the Medium Attachment Unit (MAU) functions for the Twisted Pair Medium, as specified by the supplement to IEEE 802.3 standard (Type 10BASE-T). The T-MAU provides twisted pair driver and receiver circuits, including on-board transmit digital predistortion and receiver squelch, and a number of additional features including Link Status indication, Automatic Twisted Pair Receive Polarity Detection/Correction and Indication, Receive Carrier Sense, Transmit Active and Collision Present indication.

Twisted Pair Transmit Function

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

Twisted Pair Receive Function

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

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

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

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

Link Test Function

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

When the link test function is enabled (DLNKTST bit in CSR15 is cleared), the absence of link beat pulses and receive data on the RXD \pm pair will cause the TMAU to go into the Link Fail state. In the Link Fail state, data transmission, data reception, data loopback and the collision detection functions are disabled and remain disabled

until valid data or greater than 5 consecutive link pulses appear on the RXD \pm pair. During Link Fail, the Link Status (LNKST indicated by $\overline{\text{LED0}}$) signal is inactive. When the link is identified as functional, the LNKST signal is asserted, and $\overline{\text{LED0}}$ output will be activated.

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

Polarity Detection and Reversal

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

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

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

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

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

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

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

Twisted Pair Interface Status

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

The T-MAU will power up in the Link Fail state and the normal algorithm will apply to allow it to enter the Link Pass state. In the Link Pass state, transmit or receive activity will be indicated by assertion of RCV signal going active. If T-MAU is selected using the PORTSEL bits in CSR15, when moving from AU1 to T-MAU selection, the T-MAU will be forced into the Link Fail state.

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

Collision Detect Function

Activity on both twisted pair signals RXD \pm and TXD \pm constitutes a collision, thereby causing the COL signal to be asserted. (COL is used by the LED control circuits) COL will remain asserted until one of the two colliding signals changes from active to idle. COL stays active for 2 bit times at the end of a collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

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

Jabber Function

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

before the T-MAU deasserts COL and re-enables the transmit circuitry.

Power Down

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

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

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

In **Snooze** mode, the T-MAU receive circuitry will remain enabled even while the SLEEP pin is driven LOW.

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

EADI (EXTERNAL ADDRESS DETECTION INTERFACE)

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

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

SRDCLK is provided to allow clocking of the receive bit stream into the external address detection logic. SRDCLK runs only during frame reception activity. Once a received frame commences and data and clock are available, the EADI logic will monitor the alternating (“1,0”) preamble pattern until the two ones of the Start Frame Delimiter (“1,0,1,0,1,0,1,1”) are detected, at which point the SF/BD output will be driven HIGH.

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

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

If an internal address match is detected by comparison with either the Physical or Logical Address field, the frame will be accepted regardless of the condition of $\overline{\text{EAR}}$. Incoming frames which do not pass the internal address comparison will continue to be received. This allows approximately 58 byte times after the last destination address bit is available to generate the $\overline{\text{EAR}}$ signal, assuming the device is not configured to accept runt packets. $\overline{\text{EAR}}$ will be ignored after 64 byte times after the SFD, and the frame will be accepted if $\overline{\text{EAR}}$ has not been asserted before this time. If Runt Packet Accept is configured, the $\overline{\text{EAR}}$ signal must be generated prior to the receive message completion, which could be as short as 12 byte times (assuming 6 bytes for source

address, 2 bytes for length, no data, 4 bytes for FCS) after the last bit of the destination address is available. $\overline{\text{EAR}}$ must have a pulse width of at least 200 ns.

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

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

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

The table below summarizes the operation of the EADI features.

Internal/External Address Recognition Capabilities

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

General Purpose Serial Interface (GPSI)

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

The PCnet-ISA+ GPSI signals are consistent with the LANCE digital serial interface. Since the GPSI functions can be accessed only through a special test mode, expect some loss of functionality to the device when the GPSI is invoked. The AUI and 10BASE-T analog interfaces are disabled along with the internal MENDEC logic. The LA (unlatched address) pins are removed and become the GPSI signals, therefore, only 20 bits of address space is available. The table below shows the GPSI pin configuration:

To invoke the GPSI signals, follow the procedure below:

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

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

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

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

GPSI Pin Configurations

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

Note:

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

IEEE 1149.1 Test Access Port Interface

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

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

Boundary Scan Circuit

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

TAP FSM

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

Supported Instructions

In addition to the minimum IEEE 1149.1 requirements (BYPASS, EXTEST and SAMPLE instructions), three additional instructions (IDCODE, TRIBYP and SETBYP) are provided to further ease board-level testing.

All unused instruction codes are reserved. See the table below for a summary of supported instructions.

Instruction Register and Decoding Logic

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

Boundary Scan Register (BSR)

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

There are four possible operational modes in the BSR cell:

| | |
|---|-----------------|
| 1 | Capture |
| 2 | Shift |
| 3 | Update |
| 4 | System Function |

Other Data Registers

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

| | |
|-------------|--|
| Bits 31–28: | Version |
| Bits 27–12: | Part number (2260) |
| Bits 11–1: | Manufacturer ID. The 11 bit manufacturer ID code for AMD is 0000000001 according to JEDEC Publication 106-A. |
| Bit 0: | Always a logic 1 |

IEEE 1149.1 Supported Instruction Summary

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

Power Saving Modes

The PCnet-ISA⁺ controller supports two hardware power-savings modes. Both are entered by asserting the SLEEP pin LOW.

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

In **Snooze** mode, enabled by setting the AWAKE bit in ISACSR2 and driving the SLEEP pin LOW, the T-MAU receive circuitry will remain enabled even while the SLEEP pin is driven LOW. The LED0 output will also continue to function, indicating a good 10BASE-T link if there are link beat pulses or valid frames present. This LED0 pin can be used to drive a LED and/or external hardware that directly controls the SLEEP pin of the PCnet-ISA⁺ controller. This configuration effectively wakes the system when there is any activity on the 10BASE-T link.

Access Operations (Software)

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

I/O Resources

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

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

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

I/O Register Access

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

separate 8-bit hardware bus cycles. The motherboard accesses the low byte before the high byte and the PCnet-ISA⁺ controller has circuitry to specifically support this type of access.

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

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

**Note that the RAP is cleared on Reset.*

IEEE Address Access

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

Boot PROM Access

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

Static RAM Access

The static RAM is only present in the shared memory mode. It is located at the address selected by the SMAM input. It may be accessed as an 8- or 16-bit resource but the latter is recommended for best performance.

Bus Cycles (Hardware)

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

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

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

All accesses to 8-bit resources (which do not return $\overline{\text{MEMCS16}}$ or $\overline{\text{IOCS16}}$) use SD0-7. If an odd byte is accessed, the Current Master swap buffer turns on. During an odd byte read the swap buffer copies the data from SD0-7 to the high byte. During an odd byte write the Current Master swap buffer copies the data from the high byte to SD0-7. The PCnet-ISA+ controller can be configured to be an 8-bit I/O resource even in a 16-bit system; this is set by the EEPROM. It is recommended that the PCnet-ISA+ controller be configured for 8-bit only I/O bus cycles for maximum compatibility with PC/AT clone motherboards.

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

Accesses to 16-bit resources (which do return $\overline{\text{MEMCS16}}$ or $\overline{\text{IOCS16}}$) use either or both SD0-7 and SD8-15. A word access is indicated by A0=0 and $\overline{\text{SBHE}}=0$ and data is transferred on all 16 data lines. An even byte access is indicated by A0=0 and $\overline{\text{SBHE}}=1$ and data is transferred on SD0-7. An odd-byte access is indicated by A0=1 and $\overline{\text{SBHE}}=0$ and data is transferred on

SD8-15. It is illegal to have A0=1 and $\overline{\text{SBHE}}=1$ in any bus cycle. The PCnet-ISA+ controller returns only $\overline{\text{IOCS16}}$; $\overline{\text{MEMCS16}}$ must be generated by external hardware if desired. The use of $\overline{\text{MEMCS16}}$ applies only to Shared Memory Mode.

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

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

Bus Master Mode

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

ISA Bus Accesses

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

*Motherboard SWAP logic drives

Refresh Cycles

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

because some motherboards generate a false $\overline{\text{DACK}}$ at that time.

Address PROM Cycles External PROM

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

arbitrates for the Private Data Bus (PRDB) if necessary. IOCHRDY is driven LOW during accesses to the address PROM.

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

Address PROM Cycles Using EEPROM Data

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

Ethernet Controller Register Cycles

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

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

RESET Cycles

A read to the reset address causes a PCnet-ISA+ controller reset. This has the same effect as asserting the RESET pin on the PCnet-ISA+ controller, such as happens during a system power-up or hard boot. The subsequent write cycle needed in the NE2100 LANCE

based family of Ethernet cards is not required but does not have any harmful effects. IOCS16 is not asserted in this cycle.

ISA Configuration Register Cycles

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

Boot PROM Cycles

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

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

Current Master Operation

Current Master operation only occurs in the bus master mode. It does not occur in shared memory mode.

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

Obtain Phase

A Master Mode Transfer Cycle begins by asserting DRQ. When the Permanent Master asserts DACK, the PCnet-ISA+ controller asserts MASTER, signifying it has taken control of the ISA bus. The Permanent Master tristates the address, command, and data lines within 60 ns of DACK going active. The Permanent Master drives AEN inactive within 71 ns of MASTER going active.

Access Phase

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

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

Release Phase

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

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

Master Mode Memory Read Cycle

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

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

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

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

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

Master Mode Memory Write Cycle

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

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

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

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

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

Shared Memory Mode

Address PROM Cycles External PROM

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

When the Private Data Bus becomes available, the PCnet-ISA⁺ controller drives \overline{APCS} active, releases $\overline{IOCHRDY}$, turns on the data path from PRD0-7, and enables the SD0-7 drivers (but not SD8-15). During this bus cycle, $\overline{IOCS16}$ is not driven active. This condition is

maintained until $\overline{\text{IOR}}$ goes inactive, at which time the access cycle ends. Data is removed from SD0-7 within 30 ns.

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

Ethernet Controller Register Cycles

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

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

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

RESET Cycles

A read to the reset address causes an PCnet-ISA+ controller reset. This has the same effect as asserting the RESET pin on the PCnet-ISA+ controller, such as happens during a system power-up or hard boot. The subsequent write cycle needed in the NE2100 LANCE-based family of Ethernet cards is not required but does not have any harmful effects. $\overline{\text{IOCS16}}$ is not asserted in this cycle.

ISA Configuration Register Cycles

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

Boot PROM Cycles

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

The PCnet-ISA+ controller assumes 16-bit ISA memory bus cycles for the boot PROM. A 16-bit boot PROM bus cycle begins with the Permanent Master driving the addresses valid and $\overline{\text{MEMR}}$ active. (AEN is not involved in memory cycles). External hardware would assert $\overline{\text{BPAM}}$ and $\overline{\text{MEMCS16}}$. The PCnet-ISA+ controller detects this combination of signals, drives IOCHRDY LOW, and reads two bytes out of the boot PROM. The data bytes read from the PROM are driven by the PCnet-ISA+ controller onto SD0-15 and IOCHRDY is released. This condition is maintained until $\overline{\text{MEMR}}$ goes inactive, at which time the access cycle ends.

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

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

Static RAM Cycles

The shared memory SRAM is an 8-bit device connected to the PCnet-ISA+ controller Private Bus, and can occupy up to 64 Kbytes of address space. In Shared Memory Mode, an external address comparator is responsible for asserting $\overline{\text{SMAM}}$ to the PCnet-ISA+ controller. $\overline{\text{SMAM}}$ is intended to be a perfect decode of the SRAM address space, i.e. LA17-23, SA16 for 64 Kbytes of SRAM. The LA signals must be latched by BALE in order to provide a stable decode for $\overline{\text{SMAM}}$.

The PCnet-ISA⁺ controller assumes 16-bit ISA memory bus cycles for the SRAM, so this same logic must assert $\overline{\text{MEMCS16}}$ to the ISA bus if 16-bit bus cycles are to be supported.

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

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

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

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

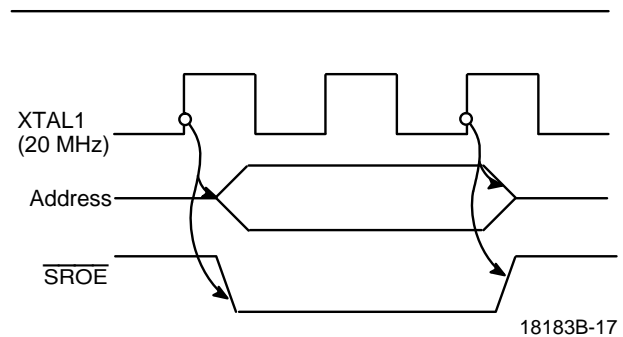
If the internal prefetch buffer contains the correct data, then the pre-fetch buffer data is driven on the System Data bus. If IOCHRDY was previously driven LOW due to either Private Data Bus arbitration or SRAM access, then it is released HIGH. The PCnet-ISA⁺ controller remains in this state until $\overline{\text{MEMR}}$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

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

The only way to configure the PCnet-ISA⁺ controller for 8-bit ISA bus cycles for SRAM accesses is to configure the entire PCnet-ISA⁺ controller to support only 8-bit ISA bus cycles. This is accomplished by leaving the $\overline{\text{SBHE}}$ pin disconnected. The PCnet-ISA⁺ controller will perform 8-bit ISA bus cycle operation for all resources

(registers, PROMs, SRAM) if $\overline{\text{SBHE}}$ has never been driven active since the last RESET, such as in the case of an 8-bit system like the PC/XT. In this case, the external address decode logic must not assert $\overline{\text{MEMCS16}}$ to the ISA bus, which will be the case if $\overline{\text{MEMCS16}}$ is left unconnected. It is possible to manufacture a dual 8/16 bit PCnet-ISA⁺ controller adapter card, as the $\overline{\text{MEMCS16}}$ and $\overline{\text{SBHE}}$ signals do not exist in the PC/XT environment.

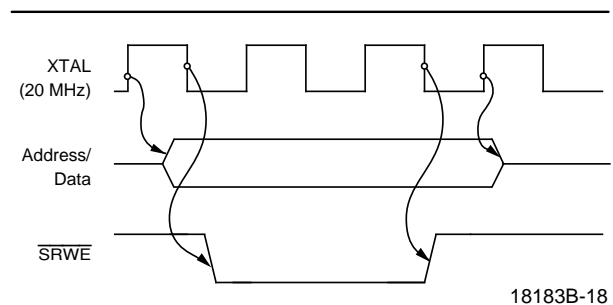
At the memory device level, each SRAM Private Bus read cycle takes two 50 ns clock periods for a maximum read access time of 75 ns. The timing looks like this:



Static RAM Read Cycle

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

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



Static RAM Write Cycle

Address and data are valid 20 ns after the rising edge of the first clock period. $\overline{\text{SRWE}}$ goes active 20 ns after the falling edge of the first clock period. $\overline{\text{SRWE}}$ goes inactive 20 ns after the falling edge of the third clock period. Address and data remain valid until the end of the third clock period. Rise and fall times are nominally 5 ns.

Non-negative setup and hold times for address and data with respect to $\overline{\text{SRWE}}$ are guaranteed. $\overline{\text{SRWE}}$ has a pulse width of typically 100 ns, minimum 75 ns.

Transmit Operation

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

Transmit Function Programming

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

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

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

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

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

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

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

Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (excluding preamble). This allows the minimum frame size of 64 bytes (512 bits) for 802.3/Ethernet to be guaranteed with no software intervention from the host/controlling process. Setting the APAD_XMT bit in CSR4 enables the automatic padding feature. The pad is placed between the LLC data field and FCS field in the 802.3 frame. FCS is always added if the frame is padded, regardless of the state of DXMTFCS. The transmit frame will be padded by bytes

with the value of 00h. The default value of APAD_XMT is 0, and this will disable auto pad generation after RESET.

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

The 544 bit count is derived from the following:

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

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

$$\text{Preamble} + (\text{Min Frame Size} + \text{FCS}) \text{ bits}$$

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

$$\begin{array}{rcl} \text{Preamble} & + & (\text{Min Frame Size} - \text{FCS}) \text{ bits} \\ 64 & + & (512 - 32) \text{ bits} \end{array}$$

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

Transmit FCS Generation

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

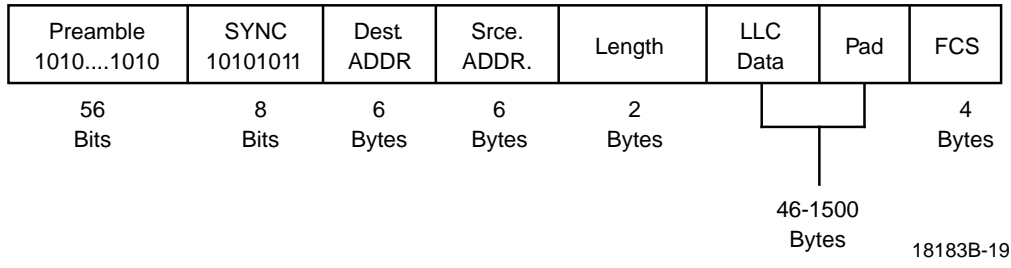
Transmit Exception Conditions

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

Normal events which may occur and which are handled autonomously by the PCnet-ISA+ controller are

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

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



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

Abnormal network conditions include:

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

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

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

Loss of Carrier

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

Late Collision

A late collision will be reported if a collision condition occurs after one slot time (512 bit times) after the transmit process was initiated (first bit of preamble commenced).

The PCnet-ISA+ controller will abandon the transmit process for the particular frame, set Late Collision (LCOL) in the associated TMD3, and process the next transmit frame in the ring. Frames experiencing a late collision will not be re-tried. Recovery from this condition must be performed by upper-layer software.

SQE Test Error

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

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

Receive Operation

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

Receive Function Programming

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

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

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

Automatic Pad Stripping

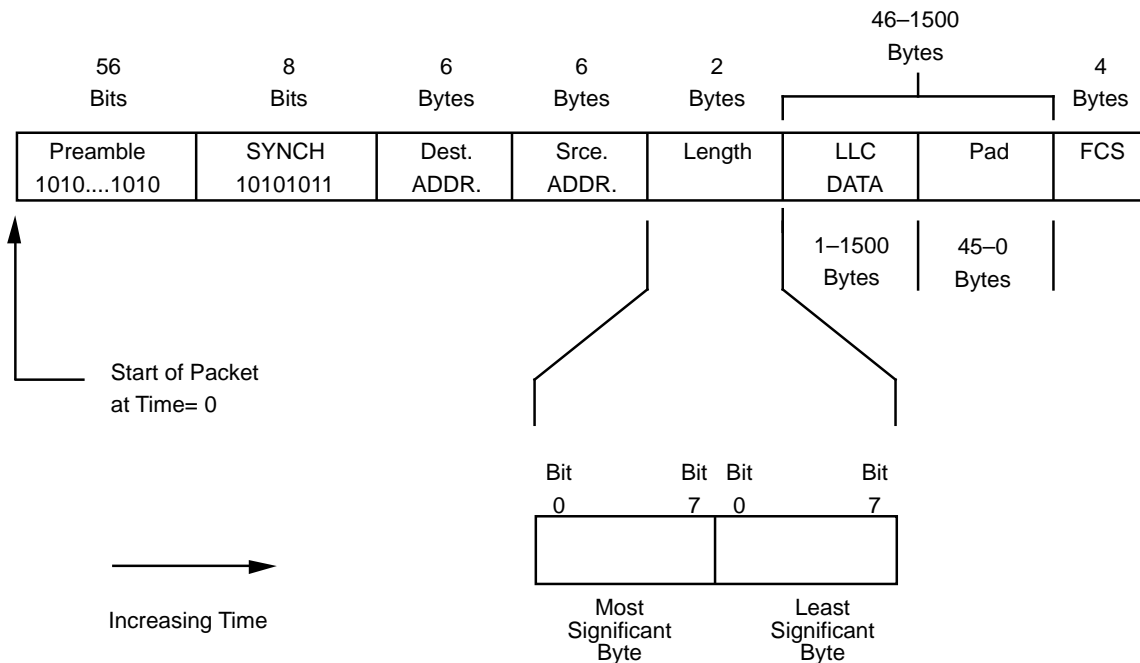
During reception of an 802.3 frame the pad field can be stripped automatically. ASTRP_RCV (bit 10 in CSR4) = 1 enables the automatic pad stripping feature. The pad field will be stripped before the frame is passed to the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has had the pad characters stripped.

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

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

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

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



18183B-20

IEEE/ANSI 802.3 Frame and Length Field Transmission Order

Receive FCS Checking

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

error is detected, this will be reported by the CRC bit in RMD1.

Receive Exception Conditions

Exception conditions for frame reception fall into two distinct categories; those which are the result of normal

network operation, and those which occur due to abnormal network and/or host related events.

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

Abnormal network conditions include:

- FCS errors
- Late collision

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

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

Loopback Operation

Loopback is a mode of operation intended for system diagnostics. In this mode, the transmitter and receiver are both operating at the same time so that the controller receives its own transmissions. The controller provides two types of internal loopback and one type of external loopback. In internal loopback mode, the transmitted data can be looped back to the receiver at one of two places inside the controller without actually transmitting any data to the external network. The receiver will move the received data to the next receive buffer, where it can be examined by software. Alternatively, in external loopback mode, data can be transmitted to and received from the external network.

There are restrictions on loopback operation. The PCnet-ISA⁺ controller has only one FCS generator circuit. The FCS generator can be used by the transmitter to generate the FCS to append to the frame, or it can be used by the receiver to verify the FCS of the received frame. It can not be used by the receiver and transmitter simultaneously.

If the FCS generator is connected to the receiver, the transmitter will not append an FCS to the frame, but the receiver will check for one. The user can, however, calculate the FCS value for a frame and include this four-byte number in the transmit buffer.

If the FCS generator is connected to the transmitter, the transmitter will append an FCS to the frame, but the

receiver will not check for the FCS. However, the user can verify the FCS by software.

During loopback, the FCS logic can be allocated to the receiver by setting DXMTFCS = 1 in CSR15.

If DXMTFCS=0, the MAC Engine will calculate and append the FCS to the transmitted message. The receive message passed to the host will therefore contain an additional 4 bytes of FCS. In this loopback configuration, the receive circuitry cannot detect FCS errors if they occur.

If DXMTFCS=1, the last four bytes of the transmit message must contain the (software generated) FCS computed for the transmit data preceding it. The MAC Engine will transmit the data without addition of an FCS field, and the FCS will be calculated and verified at the receiver.

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

The multicast address detection logic uses the FCS generator circuit. Therefore, in the loopback mode(s), the multicast address detection feature of the MAC Engine, programmed by the contents of the Logical Address Filter (LADRF [63:0] in CSRs 8–11) can only be tested when DXMTFCS=1, allocating the FCS generator to the receiver. All other features operate identically in loopback as in normal operation, such as automatic transmit padding and receive pad stripping.

When performing an internal loopback, no frame will be transmitted to the network. However, when the PCnet-ISA⁺ controller is configured for internal loopback the receiver will not be able to detect network traffic. External loopback tests will transmit frames onto the network if the AU1 port is selected, and the PCnet-PCI controller will receive network traffic while configured for external loopback when the AU1 port is selected. Runt Packet Accept is automatically enabled when any loopback mode is invoked.

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

When the 10BASE-T MAU is selected in external loopback mode, the collision detection is disabled. This is necessary, because a collision in a 10BASE-T system is defined as activity on the transmitter outputs and receiver inputs at the same time, which is exactly what occurs during external loopback.

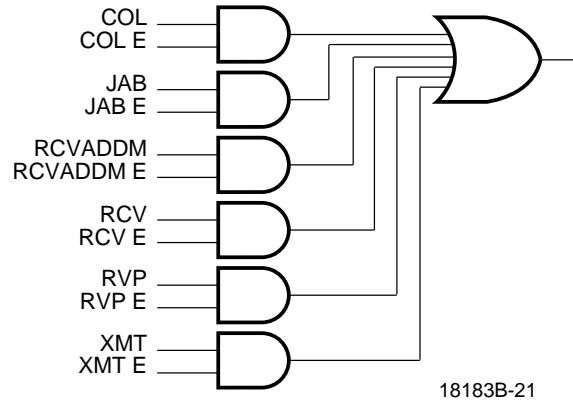
Since a 10BASE-T hub does not normally feed the station's transmitter outputs back into the station's receiver inputs, the use of external loopback in a 10BASE-T system usually requires some sort of external hardware that connects the outputs of the 10BASE-T MAU to its inputs.

LEDs

The PCnet-ISA+ controller's LED control logic allows programming of the status signals, which are displayed on 3 LED outputs. One LED (LED0) is dedicated to displaying 10BASE-T Link Status. The status signals available are Collision, Jabber, Receive, Receive Polarity (active when receive polarity is okay), and Transmit. If more than one status signal is enabled, they are ORed together. An optional pulse stretcher is available for each programmable output. This allows emulation of the TPEX (Am79C98) and TPEX+ (Am79C100) LED outputs.

| Signal | Behavior |
|---------|--|
| LNKST | Active during Link OK Not active during Link Down |
| RCV | Active while receiving data |
| RVPOL | Active during receive polarity is OK Not active during reverse receive polarity |
| RCVADDM | Active during Receive with Address Match |
| XMT | Active while transmitting data |

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



LED Control Logic

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

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

PCnet-ISA⁺ CONTROLLER REGISTERS

The PCnet-ISA⁺ controller implements all LANCE (Am7990) registers, plus a number of additional registers. The PCnet-ISA⁺ controller registers are compatible with the original LANCE, but there are some places where previously reserved LANCE bits are now used by the PCnet-ISA⁺ controller. If the reserved LANCE bits were used as recommended, there should be no compatibility problems.

Register Access

Internal registers are accessed in a two-step operation. First, the address of the register to be accessed is written into the register address port (RAP). Subsequent read or write operations will access the register pointed to by the contents of the RAP. The data will be read from (or written to) the selected register through the data port, either the register data port (RDP) for control and status registers (CSR) or the ISACSR register data port (IDP) for ISA control and status registers (ISACSR)

RAP: Register Address Port

| Bit | Name | Description |
|------|------|---|
| 15-7 | RES | Reserved locations. Read and written as zeroes. |
| 6-0 | RAP | Register Address Port select. Selects the CSR or ISACSR location to be accessed. RAP is cleared by RESET. |

Control and Status Registers

CSR0: PCnet-ISA⁺ Controller Status Register

| Bit | Name | Description |
|-----|------|--|
| 15 | ERR | Error is set by the ORing of BABL, CERR, MISS, and MERR. ERR remains set as long as any of the error flags are true. ERR is read only; write operations are ignored. |
| 14 | BABL | Babble is a transmitter time-out error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum length frame. BABL will be set if 1519 bytes or greater are transmitted. When BABL is set, IRQ is asserted if IENA = 1 and the mask bit BABLM (CSR3.14) is clear. BABL assertion will set the ERR bit. BABL is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. BABL is cleared |

13 CERR

by RESET or by setting the STOP bit.

Collision Error indicates that the collision inputs to the AUI port failed to activate within 20 network bit times after the chip terminated transmission (SQE Test). This feature is a transceiver test feature. CERR will be set in 10BASE-T mode during transmit if in Link Fail state.

CERR assertion will not result in an interrupt being generated. CERR assertion will set the ERR bit.

CERR is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. CERR is cleared by RESET or by setting the STOP bit.

12 MISS

Missed Frame is set when PCnet-ISA⁺ controller has lost an incoming receive frame because a Receive Descriptor was not available. This bit is the only indication that receive data has been lost since there is no receive descriptor available for status information.

When MISS is set, IRQ is asserted if IENA = 1 and the mask bit MISSM (CSR3.12) is clear. MISS assertion will set the ERR bit.

MISS is set by the Buffer Management Unit and cleared by writing a "1". Writing a "0" has no effect. MISS is cleared by RESET or by setting the STOP bit.

11 MERR

Memory Error is set when PCnet-ISA⁺ controller is a bus master and has not received $\overline{\text{DACK}}$ assertion after 50 μs after DRQ assertion. Memory Error indicates that PCnet-ISA⁺ controller is not receiving bus mastership in time to prevent overflow/underflow conditions in the receive and transmit FIFOs.

(MERR indicates a slightly different condition for the LANCE; for the LANCE MERR occurs when $\overline{\text{READY}}$ has not been asserted 25.6 μs after the address has been asserted.)

When MERR is set, IRQ is asserted if IENA = 1 and the mask bit MERRM (CSR3.11) is clear.

| | | | | | |
|----|------|---|---|------|---|
| | | MERR assertion will set the ERR bit. MERR is set by the Bus Interface Unit and cleared by writing a "1". Writing a "0" has no effect. MERR is cleared by RESET or by setting the STOP bit. | | | and INTR is set, IRQ will be active. INTR is cleared automatically when the condition that caused interrupt is cleared. INTR is read only. INTR is cleared by RESET or by setting the STOP bit. |
| 10 | RINT | Receive Interrupt is set after reception of a receive frame and toggling of the OWN bit in the last buffer in the Receive Descriptor Ring. When RINT is set, IRQ is asserted if IENA = 1 and the mask bit RINTM (CSR3.10) is clear. RINT is set by the Buffer Management Unit after the last receive buffer has been updated and cleared by writing a "1". Writing a "0" has no effect. RINT is cleared by RESET or by setting the STOP bit. | 6 | IENA | Interrupt Enable allows IRQ to be active if the Interrupt Flag is set. If IENA = "0" then IRQ will be disabled regardless of the state of INTR. IENA is set by writing a "1" and cleared by writing a "0". IENA is cleared by RESET or by setting the STOP bit. |
| 9 | TINT | Transmit Interrupt is set after transmission of a transmit frame and toggling of the OWN bit in the last buffer in the Transmit Descriptor Ring. When TINT is set, IRQ is asserted if IENA = 1 and the mask bit TINTM (CSR3.9) is clear. TINT is set by the Buffer Management Unit after the last transmit buffer has been updated and cleared by writing a "1". Writing a "0" has no effect. TINT is cleared by RESET or by setting the STOP bit. | 5 | RXON | Receive On indicates that the Receive function is enabled. RXON is set if DRX (CSR15.0) = "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. RXON is read only. RXON is cleared by RESET or by setting the STOP bit. |
| 8 | IDON | Initialization Done indicates that the initialization sequence has completed. When IDON is set, PCnet-ISA ⁺ controller has read the Initialization block from memory. When IDON is set, IRQ is asserted if IENA = 1 and the mask bit IDONM (CSR3.8) is clear. IDON is set by the Buffer Management Unit after the initialization block has been read from memory and cleared by writing a "1". Writing a "0" has no effect. IDON is cleared by RESET or by setting the STOP bit. | 4 | TXON | Transmit On indicates that the Transmit function is enabled. TXON is set if DTX (CSR15.1) = "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. TXON is read only. TXON is cleared by RESET or by setting the STOP bit. |
| 7 | INTR | Interrupt Flag indicates that one or more of the following interrupt causing conditions has occurred: BABL, MISS, MERR, MPCO, RCVCCO, RINT, TINT, IDON, JAB or TXSTRT; and its associated mask bit is clear. If IENA = 1 | 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 DPOLL bit in CSR4 is set; setting TDMD while DPOLL = 0 merely hastens the PCnet-ISA ⁺ controller's response to a Transmit Descriptor Ring Entry. 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 RESET or by setting the STOP bit. |

| | | | | | |
|---|------|--|-----|--------------|---|
| 2 | STOP | <p>STOP assertion disables the chip from all external 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.</p> <p>STOP is set by writing a "1" or by RESET. Writing a "0" has no effect. STOP is cleared by setting either STRT or INIT.</p> | 7-0 | IADR [23:16] | <p>Upper 8 bits of the address of the Initialization Block. Bit locations 15-8 must be written with zeros. Whenever this register is written, CSR17 is updated with CSR2's contents.</p> <p>Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.</p> |
| 1 | STRT | <p>STRT assertion enables PCnet-ISA+ controller to send and receive frames, and perform buffer management operations. Setting STRT clears the STOP bit. If STRT and INIT are set together, PCnet-ISA+ controller initialization will be performed first.</p> <p>STRT is set by writing a "1". Writing a "0" has no effect. STRT is cleared by RESET or by setting the STOP bit.</p> | | | |
| 0 | INIT | <p>INIT assertion enables PCnet-ISA+ 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, PCnet-ISA+ controller initialization will be performed first. INIT is not cleared when the initialization sequence has completed.</p> <p>INIT is set by writing a "1". Writing a "0" has no effect. INIT is cleared by RESET or by setting the STOP bit.</p> | | | |

CSR1: IADR[15:0]

| Bit | Name | Description |
|------|-------------|--|
| 15-0 | IADR [15:0] | <p>Lower address of the Initialization address register. Bit location 0 must be zero. Whenever this register is written, CSR16 is updated with CSR1's contents.</p> <p>Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.</p> |

CSR2: IADR[23:16]

| Bit | Name | Description |
|------|------|---|
| 15-8 | RES | Reserved locations. Read and written as zero. |

CSR3: Interrupt Masks and Deferral Control

| Bit | Name | Description |
|-----|-------|---|
| 15 | RES | Reserved location. Written as zero and read as undefined. |
| 14 | BABLM | <p>Babble Mask. If BABLM is set, the BABL bit in CSR0 will be masked and will not set INTR flag in CSR0.</p> <p>BABLM is cleared by RESET and is not affected by STOP.</p> |
| 13 | RES | Reserved location. Written as zero and read as undefined. |
| 12 | MISSM | <p>Missed Frame Mask. If MISSM is set, the MISS bit in CSR0 will be masked and will not set INTR flag in CSR0.</p> <p>MISSM is cleared by RESET and is not affected by STOP.</p> |
| 11 | MERRM | <p>Memory Error Mask. If MERRM is set, the MERR bit in CSR0 will be masked and will not set INTR flag in CSR0.</p> <p>MERRM is cleared by RESET and is not affected by STOP.</p> |
| 10 | RINTM | <p>Receive Interrupt Mask. If RINTM is set, the RINT bit in CSR0 will be masked and will not set INTR flag in CSR0.</p> <p>RINTM is cleared by RESET and is not affected by STOP.</p> |
| 9 | TINTM | <p>Transmit Interrupt Mask. If TINTM is set, the TINT bit in CSR0 will be masked and will not set INTR flag in CSR0.</p> <p>TINTM is cleared by RESET and is not affected by STOP.</p> |
| 8 | IDONM | <p>Initialization Done Mask. If IDONM is set, the IDON bit in CSR0 will be masked and will not set INTR flag in CSR0.</p> <p>IDONM is cleared by RESET and is not affected by STOP.</p> |
| 7-6 | RES | Reserved locations. Written as zero and read as undefined. |

5 LAPPEN

Look Ahead Packet Processing (LAPPEN) . When set to a one, the LAPPEN bit will cause the PCnet-ISA+ controller to generate an interrupt following the descriptor write operation to the first buffer of a receive packet. 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 one also enables the PCnet-ISA+ controller to read the STP bit of the receive descriptors. PCnet-ISA+ controller will use STP information to determine where it should begin writing a receive packet's data. Note that while in this mode, the PCnet-ISA+ controller can write intermediate packet data to buffers whose descriptors do not contain STP bits set to one. Following the write to the last descriptor used by a packet, the PCnet-ISA+ controller will scan through the next descriptor entries to locate the next STP bit that is set to a one. The PCnet-ISA+ controller will begin writing the next packet's data to the buffer pointed to by that descriptor.

Note that because several descriptors may be allocated by the host for each packet, and not all messages may need all of the descriptors that are allocated between descriptors that contain STP = one, then some descriptors/buffers may be skipped in the ring. While performing the search for the next STP bit that is set to one, the PCnet-ISA+ controller will advance through the receive descriptor ring regardless of the state of ownership bits. If any of the entries that are examined during this search indicate PCnet-ISA+ will RESET the OWN bit to zero in these entries. If a scanned entry indicates host ownership with STP="0", then the PCnet-ISA+ 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 PCnet-ISA+ controller, then the PCnet-ISA+ 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 PCnet-ISA+ controller, then the PCnet-ISA+ controller will stop advancing through the ring entries, store the descriptor information that is 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 zero by RESET and will unaffected by the STOP. See Appendix E for more information on LAPP.

4 DXMT2PD

Disable Transmit Two Part Deferral. (Described in the Media Access Management section). If DXMT2PD is set, Transmit Two Part Deferral will be disabled.

DXMT2PD is cleared by RESET and is not affected by STOP.

3 EMBA

Enable Modified Back-off Algorithm. If EMBA is set, a modified back-off algorithm is implemented as described in the Media Access Management section.

Read/Write accessible. EMBA is cleared by RESET and is not affected by STOP.

2-0 RES

Reserved locations. Written as zero and read as undefined.

CSR4: Test and Features Control

| Bit | Name | Description |
|-----|-------|---|
| 15 | ENTST | Enable Test Mode operation. When ENTST is set, writing to test mode registers CSR124 and CSR126 is allowed, and other |

| | | | | | |
|----|-----------|--|-----|---------|---|
| | | register test functions are enabled. In order to set ENTST, it must be written with a "1" during the first write access to CSR4 after RESET. Once a "0" is written to this bit location, ENTST cannot be set until after the PCnet-ISA+ controller is reset. | | | This bit indicates the MFC (CSR112) has overflowed. Can be cleared by writing a "1" to this bit. Also cleared by RESET or setting the STOP bit. Writing a "0" has no effect. |
| | | ENTST is cleared by RESET. | 8 | MFCOM | Missed Frame Counter Overflow Mask. If MFCOM is set, MFCO will not set INTR in CSR0. |
| 14 | DMAPLUS | When DMAPLUS = "1", the burst transaction counter in CSR80 is disabled. If DMAPLUS = "0", the burst transaction counter is enabled. | 7-6 | RES | MFCOM is set by Reset and is not affected by STOP. Reserved locations. Read and written as zero. |
| | | DMA-PLUS is cleared by RESET. | 5 | RCVCCO | Receive Collision Counter Overflow. |
| 13 | TIMER | Timer Enable Register. If TIMER is set, the Bus Timer Register, CSR82, is enabled. If TIMER is set, CSR82 must be written with a value. If TIMER is cleared, the Bus Timer Register is disabled. | | | This bit indicates the Receive Collision Counter (CSR114) has overflowed. It can be cleared by writing a 1 to this bit. Also cleared by RESET or setting the STOP bit. Writing a 0 has no effect. |
| | | TIMER is cleared by RESET. | 4 | RCVCCOM | Receive Collision Counter Overflow Mask. If RCVCCOM is set, RCVCCO will not set INTR in CSR0. |
| 12 | DPOLL | Disable Transmit Polling. If DPOLL is set, the Buffer Management Unit will disable transmit polling. Likewise, if DPOLL is cleared, automatic transmit polling is enabled. If DPOLL is set, TDMD bit in CSR0 must be periodically set in order to initiate a manual poll of a transmit descriptor. Transmit descriptor polling will not take place if TXON is reset. | 3 | TXSTRT | RCVCCOM is set by RESET and is not affected by STOP. Transmit Start status is set whenever PCnet-ISA+ controller begins transmission of a frame. When TXSTRT is set, IRQ is asserted if IENA = 1 and the mask bit TXSTRTM (CSR4.2) is clear. |
| | | DPOLL is cleared by RESET. | | | TXSTRT is set by the MAC Unit and cleared by writing a "1", setting RESET or setting the STOP bit. Writing a "0" has no effect. |
| 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 (CSR15.3). | 2 | TXSTRTM | Transmit Start Mask. If TXSTRTM is set, the TXSTRT bit in CSR4 will be masked and will not set INTR flag in CSR0. |
| | | APAD_XMT is reset by activation of the RESET pin. | | | TXS-TRTM is set by RESET and is not affected by STOP. |
| 10 | ASTRP_RCV | 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. | 1 | JAB | Jabber Error is set when the PCnet-ISA+ controller Twisted-pair MAU function exceeds an allowed transmission limit. Jabber is set by the TMAU cell and can only be asserted in 10BASE-T mode. |
| | | ASTRP_RCV is reset by activation of the RESET pin. | | | When JAB is set, IRQ is asserted if IENA = 1 and the mask bit JABM (CSR4.4) is clear. |
| 9 | MFCO | Missed Frame Counter Overflow Interrupt. | | | |

The JAB bit can be reset even if the jabber condition is still present.

JAB is set by the TMAU circuit and cleared by writing a "1". Writing a "0" has no effect. JAB is also cleared by RESET or setting the STOP bit.

| | | |
|---|------|--|
| 0 | JABM | Jabber Error Mask. If JABM is set, the JAB bit in CSR4 will be masked and will not set INTR flag in CSR0. JABM is set by RESET and is not affected by STOP. |
|---|------|--|

CSR6: RCV/XMT Descriptor Table Length

| Bit | Name | Description |
|-------|------|---|
| 15-12 | TLEN | Contains a copy of the transmit encoded ring length (TLEN) field read from the initialization block during PCnet-ISA+ controller initialization. This field is written during the PCnet-ISA+ controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. TLEN is only defined after initialization. |
| 11-8 | RLEN | Contains a copy of the receive encoded ring length (RLEN) read from the initialization block during PCnet-ISA+ controller initialization. This field is written during the PCnet-ISA+ controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. RLEN is only defined after initialization. |
| 7-0 | RES | Reserved locations. Read as zero. Write operations should not be performed. |

CSR8: Logical Address Filter, LADRF[15:0]

| Bit | Name | Description |
|------|-------------|--|
| 15-0 | LADRF[15:0] | Logical Address Filter, LADRF [15:0]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set. |

CSR9: Logical Address Filter, LADRF[31:16]

| Bit | Name | Description |
|------|--------------|--|
| 15-0 | LADRF[31:16] | Logical Address Filter, LADRF[31:16]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set. |

CSR10: Logical Address Filter, LADRF[47:32]

| Bit | Name | Description |
|------|--------------|--|
| 15-0 | LADRF[47:32] | Logical Address Filter, LADRF[47:32]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set. |

CSR11: Logical Address Filter, LADRF[63:48]

| Bit | Name | Description |
|------|--------------|--|
| 15-0 | LADRF[63:48] | Logical Address Filter, LADRF[63:48]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set. |

CSR12: Physical Address Register, PADR[15:0]

| Bit | Name | Description |
|------|------------|--|
| 15-0 | PADR[15:0] | Physical Address Register, PADR[15:0]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set. |

CSR13: Physical Address Register, PADR[31:16]

| Bit | Name | Description |
|------|-------------|---|
| 15-0 | PADR[31:16] | Physical Address Register, PADR[31:16]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set. |

CSR14: Physical Address Register, PADR[47:32]

| Bit | Name | Description |
|------|-------------|---|
| 15-0 | PADR[47:32] | Physical Address Register, PADR[47:32]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set. |

CSR15: Mode Register

| Bit | Name | Description |
|-----|--------|--|
| 15 | PROM | Promiscuous Mode. When PROM = "1", all incoming receive frames are accepted. Read/write accessible only when STOP bit is set. |
| 14 | DRCVBC | Disable Receive Broadcast. When set, disables the PCnet-ISA+ 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 the RESET pin (broadcast messages will be received). Read/write accessible only when STOP bit is set. |
| 13 | DRCVPA | Disable Receive Physical Address. When set, the physical address detection (Station or |

node ID) of the PCnet-ISA+ controller will be disabled. Frames addressed to the nodes individual physical address will not be recognized (although the frame may be accepted by the EADI mechanism).

Read/write accessible only when STOP bit is set.

12 DLNKTST

Disable Link Status. When DLNKTST = "1", monitoring of Link Pulses is disabled. When DLNKTST = "0", monitoring of Link Pulses is enabled. This bit only has meaning when the 10BASE-T network interface is selected.

Read/write accessible only when STOP bit is set.

11 DAPC

Disable Automatic Polarity Correction. When DAPC = "1", the 10BASE-T receive polarity reversal algorithm is disabled. Likewise, when DAPC = "0", the polarity reversal algorithm is enabled.

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

Read/write accessible only when STOP bit is set.

10 MENDECL

MENDEC Loopback Mode. See the description of the LOOP bit in CSR15.

Read/write accessible only when STOP bit is set.

9 LRT/TSEL

Low Receive Threshold (T-MAU Mode only)

Transmit Mode Select (AUI Mode only)

LRT

Low Receive Threshold. When LRT = "1", the internal twisted pair receive thresholds are reduced by 4.5 dB below the standard 10BASE-T value (approximately 3/5) and the unsquelch threshold for the RXD circuit will be 180-312 mV peak.

When LRT = "0", the unsquelch threshold for the RXD circuit will be the standard 10BASE-T value, 300-520 mV peak.

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

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

| | | | | | |
|-----|------------------|---|---|-------|---|
| | | Read/write accessible only when STOP bit is set. Cleared by RESET. | 4 | FCOLL | Force Collision. This bit allows the collision logic to be tested. PCnet-ISA+ controller must be in internal loopback for FCOLL to be valid. If FCOLL = "1", a collision will be forced during loopback transmission attempts; a Retry Error will ultimately result. If FCOLL = "0", the Force Collision logic will be disabled. Read/write accessible only when STOP bit is set. |
| | TSEL | Transmit Mode Select. TSEL controls the levels at which the AUI drivers rest when the AUI transmit port is idle. When TSEL = 0, DO+ and DO- yield "zero" differential to operate transformer coupled loads (Ethernet 2 and 802.3). When TSEL = 1, the DO+ idles at a higher value with respect to DO-, yielding a logical HIGH state (Ethernet 1). This bit only has meaning when the AUI network interface is selected. Not available under Auto-Select Mode. Read/write accessible only when STOP bit is set. Cleared by RESET. | | | |
| 8-7 | PORTSEL [1:0] | Port Select bits allow for software controlled selection of the network medium. PORTSEL active only when Media-Select Bit set to 0 in ISACSR2. Read/write accessible only when STOP bit is set. Cleared by RESET. The network port configuration are as follows: | | | |
| | | | | 3 | DXMTFCS |
| | | | | | Disable Transmit CRC (FCS). When DXMTFCS = 0, the transmitter will generate and append a FCS to the transmitted frame. When DXMTFCS = 1, the FCS logic is allocated to the receiver and no FCS is generated or sent with the transmitted frame. See also the ADD_FCS bit in TMD1. If DXMTFCS is set, no FCS will be generated. If both 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. In loopback mode, this bit determines if the transmitter appends FCS or if the receiver checks the FCS. This bit was called DTCR in the LANCE (Am7990). Read/write accessible only when STOP bit is set. |
| | | | | 2 | LOOP |
| 6 | INTL | Internal Loopback. See the description of LOOP, CSR15.2. Read/write accessible only when STOP bit is set. | | | Loopback Enable allows PCnet-ISA+ controller to operate in full duplex mode for test purposes. When LOOP = "1", loopback is enabled. In combination with INTL and MENDECL, various loopback modes are defined as follows: |
| 5 | DRTY | Disable Retry. When DRTY = "1", PCnet-ISA+ controller will attempt only one transmission. If DRTY = "0", PCnet-ISA+ controller will attempt to transmit 16 times before signaling a retry error. Read/write accessible only when STOP bit is set. | | | |

| PORTSEL[1:0] | Network Port |
|--------------|--------------|
| 0 0 | AUI |
| 0 1 | 10BASE-T |
| 1 0 | GPSI* |
| 1 1 | Reserved |

*Refer to the section on General Purpose Serial Interface for detailed information on accessing GPSI.

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

| | | |
|---|-----|---|
| 1 | DTX | <p>Read/write accessible only when STOP bit is set. LOOP is cleared by RESET.</p> <p>Disable Transmit. If this bit is set, the PCnet-ISA+ controller will not access the Transmit Descriptor Ring and, therefore, no transmissions will occur. DTX = "0" will set TXON bit (CSR0.4) after STRT (CSR0.1) is asserted. DTX is defined after the initialization block is read.</p> <p>Read/write accessible only when STOP bit is set.</p> |
| 0 | DRX | <p>Disable Receiver. If this bit is set, the PCnet-ISA+ controller will not access the Receive Descriptor Ring and, therefore, all receive frame data are ignored. DRX = "0" will set RXON bit (CSR0.5) after STRT (CSR0.1) is asserted. DRX is defined after the initialization block is read.</p> <p>Read/write accessible only when STOP bit is set.</p> |

CSR16: Initialization Block Address Lower

| Bit | Name | Description |
|------|------|---|
| 15-0 | IADR | <p>Lower 16 bits of the address of the Initialization Block. Bit location 0 must be zero. This register is an alias of CSR1. Whenever this register is written, CSR1 is updated with CSR16's contents.</p> <p>Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.</p> |

CSR17: Initialization Block Address Upper

| Bit | Name | Description |
|------|------|---|
| 15-8 | RES | Reserved locations. Written as zero and read as undefined. |
| 7-0 | IADR | Upper 8 bits of the address of the Initialization Block. Bit locations 15-8 must be written with zeros. |

This register is an alias of CSR2. Whenever this register is written, CSR2 is updated with CSR17's contents.

Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR18-19: Current Receive Buffer Address

| Bit | Name | Description |
|-------|------|---|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | CRBA | <p>Contains the current receive buffer address to which the PCnet-ISA+ controller will store incoming frame data.</p> <p>Read/write accessible only when STOP bit is set.</p> |

CSR20-21: Current Transmit Buffer Address

| Bit | Name | Description |
|-------|------|---|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | CXBA | <p>Contains the current transmit buffer address from which the PCnet-ISA+ controller is transmitting.</p> <p>Read/write accessible only when STOP bit is set.</p> |

CSR22-23: Next Receive Buffer Address

| Bit | Name | Description |
|-------|------|--|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | NRBA | <p>Contains the next receive buffer address to which the PCnet-ISA+ controller will store incoming frame data.</p> <p>Read/write accessible only when STOP bit is set.</p> |

CSR24-25: Base Address of Receive Ring

| Bit | Name | Description |
|-------|------|---|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | BADR | <p>Contains the base address of the Receive Ring.</p> <p>Read/write accessible only when STOP bit is set.</p> |

CSR26-27: Next Receive Descriptor Address

| Bit | Name | Description |
|-------|------|---|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | NRDA | Contains the next RDRE address pointer. Read/write accessible only when STOP bit is set. |

CSR28-29: Current Receive Descriptor Address

| Bit | Name | Description |
|-------|------|--|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | CRDA | Contains the current RDRE address pointer. Read/write accessible only when STOP bit is set. |

CSR30-31: Base Address of Transmit Ring

| Bit | Name | Description |
|-------|------|---|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | BADX | Contains the base address of the Transmit Ring. Read/write accessible only when STOP bit is set. |

CSR32-33: Next Transmit Descriptor Address

| Bit | Name | Description |
|-------|------|---|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | NXDA | Contains the next TDRE address pointer. Read/write accessible only when STOP bit is set. |

CSR34-35: Current Transmit Descriptor Address

| Bit | Name | Description |
|-------|------|--|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | CXDA | Contains the current TDRE address pointer. Read/write accessible only when STOP bit is set. |

CSR36-37: Next Next Receive Descriptor Address

| Bit | Name | Description |
|------|-------|--|
| 31-0 | NNRDA | Contains the next next RDRE address pointer. Read/write accessible only when STOP bit is set. |

CSR38-39: Next Next Transmit Descriptor Address

| Bit | Name | Description |
|------|-------|--|
| 31-0 | NNXDA | Contains the next next TDRE address pointer. Read/write accessible only when STOP bit is set. |

CSR40-41: Current Receive Status and Byte Count

| Bit | Name | Description |
|-------|------|---|
| 31-24 | CRST | Current Receive Status. This field is a copy of bits 15:8 of RMD1 of the current receive descriptor. Read/write accessible only when STOP bit is set. |
| 23-12 | RES | Reserved locations. Written as zero and read as undefined. |
| 11-0 | CRBC | Current Receive Byte Count. This field is a copy of the BCNT field of RMD2 of the current receive descriptor. Read/write accessible only when STOP bit is set. |

CSR42-43: Current Transmit Status and Byte Count

| Bit | Name | Description |
|-------|------|--|
| 31-24 | CXST | Current Transmit Status. This field is a copy of bits 15:8 of TMD1 of the current transmit descriptor. Read/write accessible only when STOP bit is set. |
| 23-12 | RES | Reserved locations. Written as zero and read as undefined. |
| 11-0 | CXBC | Current Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the current transmit descriptor. |

Read/write accessible only when STOP bit is set.

CSR44-45: Next Receive Status and Byte Count

| Bit | Name | Description |
|-------|------|---|
| 31-24 | NRST | Next Receive Status. This field is a copy of bits 15:8 of RMD1 of the next receive descriptor. Read/write accessible only when STOP bit is set. |
| 23-12 | RES | Reserved locations. Written as zero and read as undefined. |
| 11-0 | NRBC | Next Receive Byte Count. This field is a copy of the BCNT field of RMD2 of the next receive descriptor. Read/write accessible only when STOP bit is set. |

CSR46: Poll Time Counter

| Bit | Name | Description |
|------|------|--|
| 15-0 | POLL | Poll Time Counter. This counter is incremented by the PCnet-ISA+ controller microcode and is used to trigger the descriptor ring polling operation of the PCnet-ISA+ controller. Read/write accessible only when STOP bit is set. |

CSR47: Polling Interval

| Bit | Name | Description |
|-------|---------|--|
| 31-16 | RES | Reserved locations. Written as zero and read as undefined. |
| 15-0 | POLLINT | Polling Interval. This register contains the time that the PCnet-ISA+ controller will wait between successive polling operations. The POLLINT value is expressed as the two's complement of the desired interval, where each bit of POLLINT represents one-half of an XTAL1 period of time. POLLINT[3:0] are ignored. (POLLINT[16] is implied to be a one, so POLLINT[15] is significant, and does not represent the sign of the two's complement POLLINT value.) The default value of this register is 0000. This corresponds to a |

polling interval of 32,768 XTAL1 periods. The POLLINT value of 0000 is created during the microcode initialization routine, and therefore might not be seen when reading CSR47 after 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 in CSR0. Then the user may write to CSR47 and then set START in CSR0. In this way, the default value of 0000 in CSR47 will be overwritten with the desired user value.

Read/write accessible only when STOP bit is set.

CSR48-49: Temporary Storage

| Bit | Name | Description |
|------|------|---|
| 31-0 | TMP0 | Temporary Storage location. Read/write accessible only when STOP bit is set. |

CSR50-51: Temporary Storage

| Bit | Name | Description |
|------|------|---|
| 31-0 | TMP1 | Temporary Storage location. Read/write accessible only when STOP bit is set. |

CSR52-53: Temporary Storage

| Bit | Name | Description |
|------|------|---|
| 31-0 | TMP2 | Temporary Storage location. Read/write accessible only when STOP bit is set. |

CSR54-55: Temporary Storage

| Bit | Name | Description |
|------|------|---|
| 31-0 | TMP3 | Temporary Storage location. Read/write accessible only when STOP bit is set. |

CSR56-57: Temporary Storage

| Bit | Name | Description |
|------|------|---|
| 31-0 | TMP4 | Temporary Storage location. Read/write accessible only when STOP bit is set. |

CSR58-59: Temporary Storage

| Bit | Name | Description |
|------|------|---|
| 31-0 | TMP5 | Temporary Storage location. Read/write accessible only when STOP bit is set. |

CSR60-61: Previous Transmit Descriptor Address

| Bit | Name | Description |
|-------|------|---|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | PXDA | Contains the previous TDRE address pointer. The PCnet-ISA+ controller has the capability to stack multiple transmit frames. Read/write accessible only when STOP bit is set. |

CSR62-63: Previous Transmit Status and Byte Count

| Bit | Name | Description |
|-------|------|---|
| 31-24 | PXST | Previous Transmit Status. This field is a copy of bits 15:8 of TMD1 of the previous transmit descriptor. Read/write accessible only when STOP bit is set. |
| 23-12 | RES | Reserved locations. Written as zero and read as undefined. Accessible only when STOP bit is set. |
| 11-0 | PXBC | Previous Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the previous transmit descriptor. Read/write accessible only when STOP bit is set. |

CSR64-65: Next Transmit Buffer Address

| Bit | Name | Description |
|-------|------|---|
| 31-24 | RES | Reserved locations. Written as zero and read as undefined. |
| 23-0 | NXBA | Contains the next transmit buffer address from which the PCnet-ISA+ controller will transmit an outgoing frame. Read/write accessible only when STOP bit is set. |

CSR66-67: Next Transmit Status and Byte Count

| Bit | Name | Description |
|-------|------|---|
| 31-24 | NXST | Next Transmit Status. This field is a copy of bits 15:8 of TMD1 of the next transmit descriptor. Read/write accessible only when STOP bit is set. |
| 23-12 | RES | Reserved locations. Written as zero and read as undefined. Accessible only when STOP bit is set. |
| 11-0 | NXBC | Next Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the next transmit descriptor. Read/write accessible only when STOP bit is set. |

CSR68-69: Transmit Status Temporary Storage

| Bit | Name | Description |
|------|-------|---|
| 31-0 | XSTMP | Transmit Status Temporary Storage location. Read/write accessible only when STOP bit is set. |

CSR70-71: Temporary Storage

| Bit | Name | Description |
|------|------|---|
| 31-0 | TMP8 | Temporary Storage location. Read/write accessible only when STOP bit is set. |

CSR72: Receive Ring Counter

| Bit | Name | Description |
|------|-------|--|
| 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 two's complement value of -1 (FFFFh) corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set. |

CSR74: Transmit Ring Counter

| Bit | Name | Description |
|------|-------|--|
| 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 two's complement value of -1 (FFFFh) corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set. |

CSR76: Receive Ring Length

| Bit | Name | Description |
|------|-------|--|
| 15-0 | RCVRL | Receive Ring Length. Contains the Two's complement of the receive descriptor ring length. This register is initialized during the PCnet-ISA ⁺ controller initialization routine based on the value in the RLEN field of the initialization block. This register can be manually altered; the actual receive ring length is defined by the current value in this register. Read/write accessible only when STOP bit is set. |

CSR78: Transmit Ring Length

| Bit | Name | Description |
|------|-------|--|
| 15-0 | XMTRL | Transmit Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the PCnet-ISA ⁺ controller initialization routine based on the value in the TLEN field of the initialization block. This register |

can be manually altered; the actual transmit ring length is defined by the current value in this register.

Read/write accessible only when STOP bit is set.

CSR80: Burst and FIFO Threshold Control

| Bit | Name | Description |
|-------|------------|---|
| 15-14 | RES | Reserved locations. Read as ones. Written as zero. |
| 13-12 | RCVFW[1:0] | Receive FIFO Watermark. RCVFW controls the point at which ISA bus 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 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, 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). RCVFW is set to a value of 10b (64 bytes) after RESET. Read/write accessible only when STOP bit is set. |

| RCVFW[1:0] | Bytes Received |
|------------|----------------|
| 00 | 16 |
| 01 | 32 |
| 10 | 64 |
| 11 | Reserved |

11-10XMTSP[1:0] Transmit Start Point. XMTSP controls the point at which preamble transmission attempts commence in relation to the number of bytes written to the transmit FIFO for the current transmit frame. When the entire frame is in the FIFO, transmission will start regardless of the value in XMTSP. XMTSP is given a value of 10b (64 bytes) after RESET. Regardless of XMTSP, the FIFO will not internally over

write its data until at least 64 bytes (or the entire frame if <64 bytes) have been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be re-written to the transmit FIFO, and re-tries will be handled autonomously by the MAC. This bit is read/write accessible only when the STOP bit is set.

| XMTSP[1:0] | Bytes Written |
|------------|---------------|
| 00 | 4 |
| 01 | 16 |
| 10 | 64 |
| 11 | 112 |

9-8 XMTFW[1:0] Transmit FIFO Watermark. XMTFW specifies the point at which transmit DMA stops, based upon the number of write cycles that could be performed to the transmit FIFO without FIFO overflow. Transmit DMA is allowed at any time when the number of write cycles specified by XMTFW could be executed without causing transmit FIFO overflow. XMTFW is set to a value of 00b (8 cycles) after hardware RESET. Read/write accessible only when STOP bit is set.

| XMTFW[1:0] | Write Cycles |
|------------|--------------|
| 00 | 8 |
| 01 | 16 |
| 10 | 32 |
| 11 | Reserved |

7-0 DMABR DMA Burst Register. This register contains the maximum allowable number of transfers to system memory that the Bus Interface will perform during a single DMA cycle. The Burst Register is not used to limit the number of transfers during Descriptor transfers. A value of zero will be interpreted as one transfer. During RESET a value of 16 is loaded in the BURST register. If DMAPLUS (CSR4.14) is set, the DMA Burst Register is disabled.

When the Bus Activity Timer register (CSR82: DMABAT) is enabled, the PCnet-ISA+ controller will relinquish the bus when either the time specified in DMABAT has elapsed or the

number of transfers specified in DMABR have occurred.

Read/write accessible only when STOP bit is set.

CSR82: Bus Activity Timer

| Bit | Name | Description |
|------|--------|---|
| 15-0 | DMABAT | Bus Activity Timer. If the TIMER bit in CSR4 is set, this register contains the maximum allowable time that the PCnet-ISA+ controller will take up on the system bus during FIFO data transfers in each bus mastership period. The DMABAT starts counting upon receipt of \overline{DACK} from the host system. The DMABAT Register does not limit the number of transfers during Descriptor transfers. |

A value of zero will limit the PCnet-ISA+ controller to one bus cycle per mastership period. A non-zero value is interpreted as an unsigned number with a resolution of 100 ns. For instance, a value of 51 μ s would be programmed with a value of 510. When the TIMER bit in CSR4 is set, DMABAT is enabled and must be initialized by the user. The DMABAT register is undefined until written.

When the Bus Activity Timer register (CSR82: DMABAT) is enabled, the PCnet-ISA+ controller will relinquish the bus when either the time specified in DMABAT has elapsed or the number of transfers specified in DMABR have occurred. When ENTST (CSR4.15) is asserted, all writes to this register will automatically perform a decrement cycle.

Read/write accessible only when STOP bit is set.

CSR84-85: DMA Address

| Bit | Name | Description |
|------|-------|--|
| 31-0 | DMABA | DMA Address Register. This register contains 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 DMABA register is undefined until the first PCnet-ISA+ controller DMA operation.

This register has meaning only if the PCnet-ISA+ controller is in Bus Master Mode.

Read/write accessible only when STOP bit is set.

CSR86: Buffer Byte Counter

| Bit | Name | Description |
|-------|-------|---|
| 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 STOP bit is set. |

CSR88-89: Chip ID

| Bit | Name | Description |
|-------|------|--|
| 31-28 | | Version. This 4-bit pattern is silicon revision dependent. |
| 27-12 | | Part number. The 16-bit code for the PCnet-ISA+ controller is 0010001001100000b. |
| 11-1 | | Manufacturer ID. The 11-bit manufacturer code for AMD is 00000000001b. This code is per the JEDEC Publication 106-A. |
| 0 | | Always a logic 1. This register is exactly the same as the Chip ID register in the JTAG description. |

CSR92: Ring Length Conversion

| Bit | Name | Description |
|------|------|--|
| 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 STOP bit is set.

CSR94: Transmit Time Domain Reflectometry Count

| Bit | Name | Description |
|-------|--------|--|
| 15-10 | RES | Reserved locations. Read and written as zero. |
| 9-0 | XMTTDR | Time Domain Reflectometry reflects the state of an internal counter that counts from the start of transmission to the occurrence of loss of carrier. TDR is incremented at a rate of 10 MHz. Read accessible only when STOP bit is set. Write operations are ignored. XMTTDR is cleared by RESET. |

CSR96-97: Bus Interface Scratch Register 0

| Bit | Name | Description |
|------|------|---|
| 31-0 | SCR0 | This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. The SCR0 register is undefined until written. Read/write accessible only when STOP bit is set. |

CSR98-99: Bus Interface Scratch Register 1

| Bit | Name | Description |
|------|------|---|
| 31-0 | SCR1 | This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. Read/write accessible only when STOP bit is set. |

CSR104-105: SWAP

| Bit | Name | Description |
|------|------|---|
| 31-0 | SWAP | This register performs word and byte swapping depending upon if |

32-bit or 16-bit internal write operations are performed. This register is used internally by the BIU/BMU as a word or byte swapper. The swap register can perform 32-bit operations that the PC can not; the register is externally accessible for test reasons only. CSR104 holds the lower 16 bits and CSR105 holds the upper 16 bits.

The swap function is defined as follows:

| Internal Write Operation | SWAP Register Result |
|--------------------------|--|
| 32-Bit word | SRC[31:16] → SWAP[15:0] SRC[15:0] → SWAP[31:16] |
| Lower 16-Bit (CSR104) | SRC[15:8] → SWAP[7: 0] SRC[7:0] → SWAP[15:8] |

Read/write accessible only when STOP bit is set.

CSR108-109: Buffer Management Scratch

| Bit | Name | Description |
|------|-------|---|
| 31-0 | BMSCR | The Buffer Management Scratch register is used for assembling Receive and Transmit Status. This register is also used as the primary scan register for Buffer Management Test Modes. BMSCR register is undefined until written. Read/write accessible only when STOP bit is set. |

CSR112: Missed Frame Count

| Bit | Name | Description |
|------|------|--|
| 15-0 | MFC | Counts the number of missed frames. This register is always readable and is cleared by STOP. A write to this register performs an increment when the ENTST bit in CSR4 is set. When MFC is all 1's (65535) and a missed frame occurs, MFC increments to 0 and sets MFC0 bit (CSR4.9). |

CSR114: Receive Collision Count

| Bit | Name | Description |
|------|-------|---|
| 15-0 | RCVCC | Counts the number of Receive collisions seen, regular and late. This register is always readable and is cleared by STOP. A write to this register performs an increment when the ENTST bit in CSR4 is set. When RCVCC is all 1's (65535) and a receive collision occurs, RCVCC increments to 0 and sets RCVCC0 bit (CSR4.5) |

CSR124: Buffer Management Unit Test

| Bit | Name | Description |
|------|--------|---|
| 15-5 | RES | This register is used to place the BMU/BIU into various test modes to support Test/Debug. This register is writeable when the ENTST bit in CSR4 is set. Reserved locations. Written as zero and read as undefined. |
| 4 | GPSIEN | This mode places the PCnet-ISA+ controller in the GPSI Mode. This mode will reconfigure the External Address Pins so that the GPSI port is exposed. This allows bypassing the MENDEC- TMAU logic. This bit should only be set if the external logic supports GPSI operation. Damage to the device may occur in a non-GPSI configuration. Refer to the GPSI section. |
| 3 | RPA | Runt Packet Accept. This bit forces the CORE receive logic to accept Runt Packets. This bit allows for faster testing. |
| 2-0 | RES | For test purposes only. Reserved locations. Written as zero and read as undefined. |

ISA Bus Configuration Registers

The ISA Bus Data Port (IDP) allows access to registers which are associated with the ISA bus. These registers are called ISA Bus Configuration Registers (ISACSRs), and are indexed by the value in the Register Address Port (RAP). The table below defines the ISACSRs which

can be accessed. All registers are 16 bits. The “Default” value is the value in the register after reset and is hexadecimal.

Refer to the section “LEDs” for information on LED control logic.

| ISACSR | MNEMONIC | Default | Name |
|--------|----------|---------|---|
| 0 | MSRDA | 0005H | Master Mode Read Active |
| 1 | MSWRA | 0005H | Master Mode Write Active |
| 2 | MC | 0002H | Miscellaneous Configuration |
| 3 | EC | 8000H* | EEPROM Configuration |
| 4 | LED0 | 0000H | Link Integrity |
| 5 | LED1 | 0084H | Default: RCV |
| 6 | LED2 | 0008H | Default: RCVPOL |
| 7 | LED3 | 0090H | Default: XMT |
| 8 | SC | 0000H | Software Configuration (Read-Only register) |

*This value can be 0000H for systems that do not support EEPROM option.

ISACSR0: Master Mode Read Active

| Bit | Name | Description |
|------|-------|---|
| 3-0 | MSRDA | This register is used to tune the MEMR command signal active time. The value stored in MSRDA defines the number of 50 ns periods that the command signal is active. The default value of 5h indicates 250 ns pulse widths. A value of 0 or 1 will generate 50 ns wide commands. |
| 15-4 | RES | Reserved locations. Written as zero and read as undefined. |

ISACSR1: Master Mode Write Active

| Bit | Name | Description |
|------|-------|---|
| 3-0 | MSWRA | This register is used to tune the MEMW command signal active time. The value stored in MSWRA defines the number of 50 ns periods that the command signal is active. The default value of 5h indicates 250 ns pulse widths. A value of 0 or 1 will generate 50 ns wide commands. |
| 15-4 | RES | Reserved locations. Written as zero and read as undefined. |

ISACSR2: Miscellaneous Configuration

| Bit | Name | Description |
|-----|-------------|---|
| 15 | MODE_STATUS | Mode Status. This is a read-only register which indicates whether the PCnet-ISA+ is configured in shared memory mode. A set condition indicates shared-memory while a clear condition indicates bus-master condition. |
| 14 | TMAU_LOOPE | 10BASE-T External Loop back Enable. This bit is usable only when 10BASE-T is selected AND PCnet-ISA+ is in external loop back. External loop back is set during initialization via the MODE register. When TMAU_LOOPE is set, a board level test is enabled via a loop back clip which ties the 10BASE-T RJ45 transmit pair to the receiver pair. This will test all external components (i.e. transformers, resistors, etc.) of the 10BASE-T path. TMAU_LOOPE assertion is not suitable for live network tests. When TMAU_LOOPE is deasserted, default condition, external loop back in 10BASE-T is allowed. |
| 13 | Reserved | Written with zero and read as undefined. |
| 12 | SLOT_ID | Slot Identification. This is a read-only register bit which indicates if PCnet-ISA+ is either in an 16 or 8 bit slot. Reading a one indicates an 8 bit slot. Zero indicates a 16-bit slot. (SLOT_ID bit is not valid after the INIT bit is set in CSR0.) |
| 11 | ISA_PROTECT | ISA Protect. When set, the ISACSR's 0-2 and 4-7 are protected from being written over by software drivers. When ISA_PROTECT is cleared, ISACSR's 0-7 are allowed to be written over by software and reset by reading the Software reset I/O location. (Default is zero) |
| 10 | EISA_DECODE | EISA Decode. This control bit allows EISA product identifier registers 12-bit decode xC80 - xC83 (4 Bytes). Default is zero. |
| 9 | P&P_ACT | Plug and Play Active. When this bit is set, PCnet-ISA+ will become active after serially reading the EEPROM. If check sum failure exist, PCnet-ISA+ will not become active and alternate |

- 8 APWEN Address PROM Write Enable. It is reset to zero by RESET. When asserted, this pin allows write access to the internal Address PROM RAM. APWEN is used also to protect the Flash device from write cycles. When programming of the Flash device is required, the APWEN bit needs to be set. When reset, this pin protects the internal Address PROM RAM, and external Flash device from being overwritten.
- 7 EISA_LVL EISA Level. This bit is a read-only register. It indicates if the level or edge sensitive interrupts have been selected. A set condition indicates level sensitive interrupts. A clear condition indicates ISA edge.
- 6 DSDBUS Disable Staggered Data Bus. When this bit is a zero, the data bus driver timing is staggered from the address bus driver timing in Bus Master mode. When this bit is a one, the data bus is not staggered. It is similar to the PCnet-ISA (Am79C960) timing. This bit is reset to zero. For most applications, this bit should not have to be set.
- 5 10BASE5_SEL 10BASE5 Select. When this bit is a one, the DC to DC converter will be deselected via the DXCVR pin. When 10BASE5_SEL is a zero, the DC to DC converter will be selected via the DXCVR bit when the AUI port is selected to support a DC-DC converter for 10BASE2 MUAs. When 10BASE-T port is selected by whatever means, DXCR pin will high independent of the bit selected by the driver software mode register, MEDSEL bits, and Auto Selection process. 10BASE5_SEL is reset to zero.
- 4 ISAINACT ISAINACT allows for reduced inactive timing appropriate for modern ISA machines. ISAINACT is cleared when RESET is asserted. When ISAINACT is a zero, tMMR3 and tMMW3 parameters are nominally 200 ns, which is compatible with EISA system. When ISAINACT is set by writing a one, tMMR3 and tMMW3 are nominally set to 100 ns.

- 3 EADISEL EADI Select. Enables EADI match mode.
When EADI mode is selected, the pins named LED1, LED2, and LED3 change in function while LED0 continues to indicate 10BASE-T Link Status.

| LED | EADI Function |
|-----|---------------|
| 1 | SF/BD |
| 2 | SRD |
| 3 | SRDCLK |

- 2 AWAKE Auto-Wake. If AWAKE = "1", the 10BASE-T receive circuitry is active during sleep and listens for Link Pulses. LED0 indicates Link Status and goes active if the 10BASE-T port comes out of "link fail" state. This LED0 pin can be used by external circuitry to re-enable the PCnet-ISA+ controller and/or other devices.

When AWAKE = "0", the Auto-Wake circuitry is disabled. This bit only has meaning when the 10BASE-T network interface is selected.

- 1,0 MEDSEL Media Select. It was previously defined as ASEL (Auto Select) and XMAUSEL (External MAU Select) in the PCnet-ISA. They are now combined together and defined to be software compatible with ASEL and XMAUSEL in the PCnet-ISA (Am79C960).

| MEDSEL (1:0) | Function |
|--------------|-----------------------------------|
| 0 0 | Software Select (Mode Reg, CSR15) |
| 0 1 | 10BASE-T Port |
| 1 0 | Auto Selection (Default) |
| 1 1 | AUI Port |

ISACSR3: EEPROM Configuration

| Bit | Name | Description |
|-----|----------|---|
| 15 | EE_VALID | EEPROM Valid. This bit is a read-only register. When a one is read, EE_PROM has a valid checksum. The sum of the total bytes reads should equals FF hex. When a zero is read, checksum failed, or SHFTBUSY pin was sampled with a zero which indicates no EEPROM present. |
| 14 | EE_LOAD | EEPROM Load. When written with a one, the device will load the EE_PROM into the |

| | | |
|------|---------|--|
| | | PCnet-ISA+, performing self configuration. This command must be last write to ISACSR3 Register. PCnet-ISA+ will not respond to any slave commands while loading the EE_PROM register. EE_LOAD will be reset with a zero after EE_PROM is read. It takes approximately, 1.4 ms for serial EEPROM load process to complete. |
| 13-5 | N/A | Reserved. Read and written as zeros. |
| 4 | EE_EN | EEPROM Enable. When EE_EN is written with a one, the lower three bits of PRDB becomes SK, DI and DO, respectively. EECS and SHFBUSY are controlled by the software select bits. This bit must be written with a one to write to or read from the EEPROM. PCnet-ISA+ should be in the STOP state when EE_EN is written. When EN_EN is cleared, DI/DO, SK, EECS and SHFBUSY have no control. |
| 3 | SHFBUSY | Shift Busy. SHFBUSY allows for the control of the SHFBUSY pin. When a one is written, SHFBUSY goes high provided EE_EN is a 1. When a zero is written, SHFBUSY is held to a zero. When EE_EN is cleared, SHFBUSY will maintain the last value programmed. (Refer to Bit 4 above, EE_EN, for detailed use of this bit.) |
| 2 | EECS | EEPROM Chip Select. EECS asserts the chip select to the Serial EEPROM. (Refer to Bit 4 above, EE_EN, for detailed use of this bit.) |
| 1 | SK | Serial Shift Clock. SK controls the SK input to the Serial EEPROM and the optional External Shift Logic. (Refer to Bit 4 above, EE_EN, for detailed use of this bit.) |
| 0 | DI/DO | Serial Shift Data In and Serial Shift Data Out. When written, this bit controls the DI input of the serial EEPROM. When read, this bit represents the DO value of the serial EEPROM. (Refer to Bit 4 above, EE_EN, for detailed use of this bit.) |

ISACSR4: LED0 Status (Link Integrity)

| Bit | Name | Description |
|------|-------|--|
| 15 | LNKST | ISACSR4 is a non-programmable register that uses one bit to reflect the status of the LED0 pin. This pin defaults to twisted pair MAU Link Status (LNKST) and is not programmable. LNKST is a read-only register bit that indicates whether the Link Status LED is asserted. When LNKST is read as zero, the Link Status LED is not asserted. When LNKST is read as one, the Link Status LED is asserted, indicating good 10BASE-T integrity. |
| 14-0 | RES | Reserved locations. Written as zero, read as undefined. |

ISACSR5: LED1 Status

| Bit | Name | Description |
|------|---------|---|
| 15 | LEDOUT | ISACSR5 controls the LED1 function(s) that the 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. ISACSR5 defaults to Receive Status (RCV) with pulse stretcher enabled (PSE = 1) and is fully programmable. Indicates the current (non-stretched) state of the function(s) generated. Read only. |
| 14-8 | RES | Reserved locations. Read and written as zero. |
| 7 | PSE | Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence. 0 is disabled, 1 is enabled. |
| 6 | RES | Reserved locations. Read and written as zero. |
| 5 | RCVADDM | Receive Address Match. This bit when set allows for LED control of only receive packets which match internal address match. |
| 4 | XMT E | Enable Transmit Status Signal. Indicates PCnet-ISA+ controller transmit activity . |

| | | |
|---|---------|---|
| 3 | RVPOL E | 0 disables the signal, 1 enables the signal. Enable Receive Polarity Signal. Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored. |
| 2 | RCV E | Enable Receive Status Signal. Indicates receive activity on the network. 0 disables the signal, 1 enables the signal. |
| 1 | JAB E | Enable Jabber Signal. Indicates the PCnet-ISA+ controller is jabbering on the network. 0 disables the signal, 1 enables the signal. |
| 0 | COL E | Enable Collision Signal. Indicates collision activity on the network. 0 disables the signal, 1 enables the signal. |

| RVPOLE | LEDXOR | Result |
|--------|--------|---|
| 0 | X | 10BASE-T polarity function ignored |
| 1 | 0 | LED1 pin low with "Good" 10BASE-T polarity (LED on) |
| 1 | 1 | LED1 pin high with "Good" 10BASE-T polarity (LED off) |

ISACSR6: LED2 Status

| Bit | Name | Description |
|-----|------|-------------|
|-----|------|-------------|

| | | |
|----|--------|--|
| | | ISACSR6 controls the function(s) that the 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. ISACSR6 defaults to twisted pair MAU Receive Polarity (RCVPOL) with pulse stretcher enabled (PSE = 1) and is fully programmable. |
| 15 | LEDOUT | Indicates the current (non-stretched) state of the function(s) generated. Read only. |
| 14 | LEDXOR | This bit when set causes LED2 to be an active high signal when asserted. When this bit is cleared, LED2 will be active low when asserted. <i>(Note: This bit when used in conjunction with the RVPOLE bit (Bit 3) of ISACSR5, ISACSR6, and ISACSR7 can be used to create a "Polarity Bad" LED.)</i> |

| | | |
|------|---------|---|
| 13-8 | RES | Reserved locations. Read and written as zero. |
| 7 | PSE | Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence. 0 is disabled, 1 is enabled. |
| 6 | RES | Reserved locations. Read and written as zero. |
| 5 | RCVADDM | Receive Address Match. This bit when set allows for LED control of only receive packets that match internal address match. |
| 4 | XMT E | Enable Transmit Status Signal. Indicates PCnet-ISA+ controller transmit activity . 0 disables the signal, 1 enables the signal. |
| 3 | RVPOL E | Enable Receive Polarity Signal. Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored. |
| 2 | RCV E | Enable Receive Status Signal. Indicates receive activity on the network. 0 disables the signal, 1 enables the signal. |
| 1 | JAB E | Enable Jabber Signal. Indicates the PCnet-ISA+ controller is jabbering on the network. 0 disables the signal, 1 enables the signal. |
| 0 | COL E | Enable Collision Signal. Indicates collision activity on the network. 0 disables the signal, 1 enables the signal. |

ISACSR7: LED3 Status

| Bit | Name | Description |
|------|---------|---|
| | | ISACSR7 controls the function(s) that the 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. ISACSR7 defaults to Transmit Status (XMT) with pulse stretcher enabled (PSE = 1) and is fully programmable. |
| 15 | LEDOUT | Indicates the current (non-stretched) state of the function(s) generated. Read only. |
| 14-8 | RES | Reserved locations. Read and written as zero. |
| 7 | PSE | Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence. 0 is disabled, 1 is enabled. |
| 6 | RES | Reserved locations. Read and written as zero. |
| 5 | RCVADDM | Receive Address Match. This bit when set allows for LED control of only receive packets that match internal address match. |
| 4 | XMT E | Enable Transmit Status Signal. Indicates PCnet-ISA+ controller transmit activity. 0 disables the signal, 1 enables the signal. |
| 3 | RVPOL E | Enable Receive Polarity Signal. Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored. |
| 2 | RCV E | Enable Receive Status Signal. Indicates receive activity on the network. 0 disables the signal, 1 enables the signal. |
| 1 | JAB E | Enable Jabber Signal. Indicates the PCnet-ISA+ controller is jabbering on the network. 0 disables the signal, 1 enables the signal. |
| 0 | COL E | Enable Collision Signal. Indicates collision activity on the network. 0 disables the signal, 1 enables the signal. |

ISACSR8: Software Configuration Register (Read-Only Register)

| Bit | Description |
|-------|--|
| 15–12 | Read-only image of SR_AM(3:0) of P&P register 0x48 - 0x49. |
| 11–8 | Read-only image of BP_AM(3:0) of P&P register 0x40 - 0x41. |
| 7–4 | Read-only image of IRQSEL(3:0) of P&P register 0x70. |
| 3 | Reserved, written with zero, read as undefined. |
| 2–0 | Read-only image of DMASEL(2:0) of P&P register 0x74. |

Initialization Block

The figure below shows the Initialization Block memory configuration. Note that the Initialization Block must be based on a word (16-bit) boundary.

| Address | Bits 15–12 | Bits 11–8 | Bits 7–4 | Bits 3–0 |
|---------|-------------|-----------|------------|----------|
| IADR+00 | MODE 15-00 | | | |
| IADR+02 | PADR 15-00 | | | |
| IADR+04 | PADR 31-16 | | | |
| IADR+06 | PADR 47-32 | | | |
| IADR+08 | LADRF 15-00 | | | |
| IADR+10 | LADRF 31-16 | | | |
| IADR+12 | LADRF 47-32 | | | |
| IADR+14 | LADRF 63-48 | | | |
| IADR+16 | RDRA 15-00 | | | |
| IADR+18 | RLEN | RES | RDRA 23-16 | |
| IADR+20 | TDRA 15-00 | | | |
| IADR+22 | TLEN | RES | TDRA 23-16 | |

RLEN and TLEN

The TLEN and RLEN fields in the initialization block are 3 bits wide, occupying bits 15, 14, and 13, and the value in these fields determines the number of Transmit and Receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is as follows:

| R/TLEN | # of DREs |
|--------|-----------|
| 000 | 1 |
| 001 | 2 |
| 010 | 4 |
| 011 | 8 |
| 100 | 16 |
| 101 | 32 |
| 110 | 64 |
| 111 | 128 |

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete. See the description of the appropriate CSRs.

RDRA and TDRA

TDRA and RDRA indicate where the transmit and receive descriptor rings, respectively, begin. Each DRE must be located on an 8-byte boundary.

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", the address is deemed logical. 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 are 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 zeroes and promiscuous mode is disabled, all incoming logical addresses except broadcast will be rejected.

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is handled as follows:

- 1) If the Disable Broadcast Bit is cleared, the broadcast address is accepted.
- 2) If the Disable Broadcast Bit is set and promiscuous mode is enabled, the broadcast address is accepted.
- 3) If the Disable Broadcast Bit is set and promiscuous mode is disabled, the broadcast address is rejected.

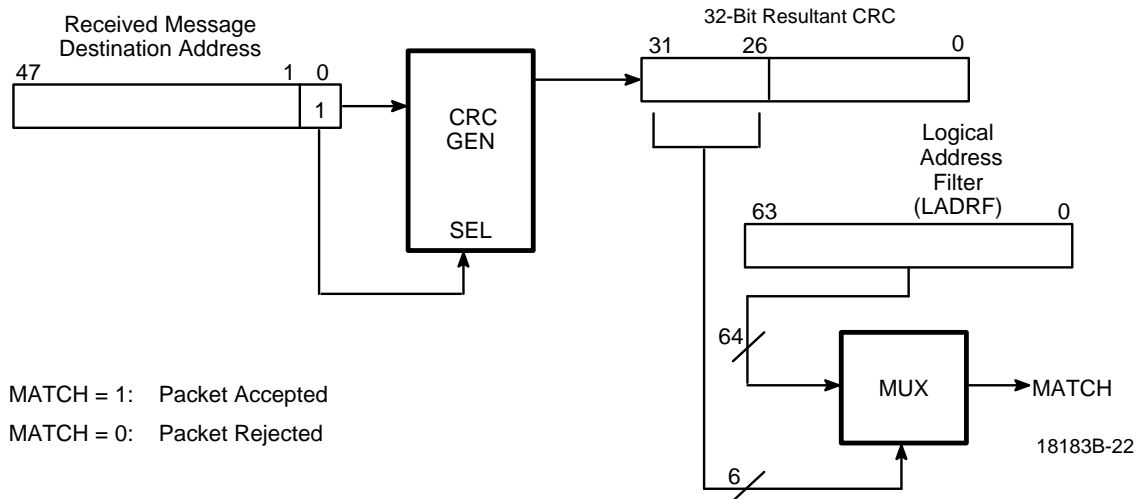
If external loopback is used, the FCS logic must be allocated to the receiver (by setting the DXMTFCS bit in CSR15, and clearing the ADD_FCS bit in TMD1) when using multicast addressing.

PADR

This 48-bit value represents the unique node address assigned by the IEEE and used for internal address comparison. PADR[0] is the first address bit transmitted on the wire, and must be zero. The six-hex-byte nomenclature used by the IEEE maps to the PCnet-ISA+ controller PADR register as follows: the first byte comprises PADR[7:0], with PADR[0] being the least significant bit of the byte. The second IEEE byte maps to PADR[15:8], again from LSbit to MSbit, and so on. The sixth byte maps to PADR[47:40], the LSbit being PADR[40].

MODE

The mode register in the initialization block is copied into CSR15 and interpreted according to the description of CSR15.



Address Match Logic

Receive Descriptors

The Receive Descriptor Ring Entries (RDREs) are composed of four receive message fields (RMD0-3). Together they contain the following information:

- The address of the actual message data buffer in user (host) memory
- The length of that message buffer
- Status information indicating the condition of the buffer. The eight most significant bits of RMD1 (RMD1[15:0]) are collectively termed the STATUS of the receive descriptor.

RMD0

Holds LADRF [15:0]. This is combined with HADR [7:0] in RMD1 to form the 24-bit address of the buffer pointed to by this descriptor table entry. There are no restrictions on buffer byte alignment or length.

RMD1

| Bit | Name | Description |
|-----|------|--|
| 15 | OWN | This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-ISA+ controller (OWN=1). The PCnet-ISA+ controller clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the PCnet-ISA+ controller or host has relinquished ownership of a buffer, it must not change any field in the descriptor entry. |
| 14 | ERR | ERR is the OR of FRAM, OFLO, CRC, or BUFF. ERR is written by the PCnet-ISA+ controller. |

| | | |
|----|------|---|
| 13 | FRAM | FRAMING ERROR indicates that the incoming frame contained 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 written by the PCnet-ISA+ controller. |
| 12 | OFLO | OVERFLOW error indicates that the receiver has lost all or part of the incoming frame, due to an inability to store the frame in a memory buffer before the internal FIFO overflowed. OFLO is valid only when ENP is not set. OFLO is written by the PCnet-ISA+ controller. |
| 11 | 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 written by the PCnet-ISA+ controller. |
| 10 | BUFF | BUFFER ERROR is set any time the PCnet-ISA+ controller does not own the next buffer while data chaining a received frame. This can occur in either of two ways: <ol style="list-style-type: none"> 1) The OWN bit of the next buffer is zero 2) FIFO overflow occurred before the PCnet-ISA+ controller polled 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 written by the PCnet-ISA+ controller.

| | | |
|-----|------|--|
| 9 | STP | START OF PACKET indicates that this is the first buffer used by the PCnet-ISA+ controller for this frame. It is used for data chaining buffers. STP is written by the PCnet-ISA+ controller in normal operation. In SRPINT Mode (CSR3.5 set to 1) this bit is written by the driver. |
| 8 | ENP | END OF PACKET indicates that this is the last buffer used by the PCnet-ISA+ 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 written by the PCnet-ISA+ controller. |
| 7-0 | HADR | The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and is not changed by the PCnet-ISA+ controller. |

RMD2

| Bit | Name | Description |
|-------|------|--|
| 15-12 | ONES | MUST BE ONES. This field is written by the host and unchanged by the PCnet-ISA+ 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 is not changed by the PCnet-ISA+ controller. |

RMD3

| Bit | Name | Description |
|-------|------|--|
| 15-12 | RES | RESERVED and read as zeros. |
| 11-0 | MCNT | MESSAGE BYTE COUNT is the length in bytes of the received message, expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the PCnet-ISA+ controller and cleared by the host. |

Transmit Descriptors

The Transmit Descriptor Ring Entries (TDREs) are composed of four transmit message fields (TMD0-3). Together they contain 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. The eight most significant bits of TMD1 (TMD1[15:8]) are collectively termed the STATUS of the transmit descriptor.

Note that bit 13 of TMD1, which was formerly a reserved bit in the LANCE (Am7990), is assigned a new meaning, ADD_FCS.

TMD0

Holds LADR [15:0]. This is combined with HADR [7:0] in TMD1 to form a 24-bit address of the buffer pointed to by this descriptor table entry. There are no restrictions on buffer byte alignment or length.

TMD1

| Bit | Name | Description |
|-----|---------|--|
| 15 | OWN | This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-ISA+ controller (OWN=1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The PCnet-ISA+ controller clears the OWN bit after transmitting the contents of the buffer. Both the PCnet-ISA+ controller and the host must not alter a descriptor entry after it has relinquished ownership. |
| 14 | ERR | ERR is the OR of UFLO, LCOL, LCAR, or RTRY. ERR is written by the PCnet-ISA+ controller. 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. |
| 13 | ADD_FCS | ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the STP bit is set. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS = 0, FCS generation is controlled by DXMTFCS. ADD_FCS is written |

| | | | TMD2 | | |
|-----|------|---|-------------|-------------|---|
| | | | Bit | Name | Description |
| 12 | MORE | by the host, and unchanged by the PCnet-ISA+ controller. This was a reserved bit in the LANCE (Am7990). MORE indicates that more than one re-try was needed to transmit a frame. MORE is written by the PCnet-ISA+ controller. This bit has meaning only if the ENP or the ERR bit is set. | 15-12 | ONES | MUST BE ONES. This field is written by the host and unchanged by the PCnet-ISA+ controller. |
| 11 | ONE | ONE indicates that exactly one re-try was needed to transmit a frame. ONE flag is not valid when LCOL is set. ONE is written by the PCnet-ISA+ controller. This bit has meaning only if the ENP or the ERR bit is set. | 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 is the number of bytes from this buffer that will be transmitted by the PCnet-ISA+ controller. This field is written by the host and is not changed by the PCnet-ISA+ controller. There are no minimum buffer size restrictions. Zero length buffers are allowed for protocols which require it. |
| 10 | DEF | DEFERRED indicates that the PCnet-ISA+ controller had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the PCnet-ISA+ controller is ready to transmit. DEF is written by the PCnet-ISA+ controller. This bit has meaning only if the ENP or ERR bits are set. | | | |
| | | | TMD3 | | |
| | | | Bit | Name | Description |
| 9 | STP | START OF PACKET indicates that this is the first buffer to be used by the PCnet-ISA+ controller for this frame. It is used for data chaining buffers. The STP bit must be set in the first buffer of the frame, or the PCnet-ISA+ controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is written by the host and is not changed by the PCnet-ISA+ controller. | 15 | BUFF | BUFFER ERROR is set by the PCnet-ISA+ controller during transmission when the PCnet-ISA+ controller does not find the ENP flag in the current buffer and does not own the next buffer. This can occur in either of two ways: 1) The OWN bit of the next buffer is zero. 2) FIFO underflow occurred before the PCnet-ISA+ controller obtained the next STATUS byte (TMD1[15:8]). |
| 8 | ENP | END OF PACKET indicates that this is the last buffer to be used by the PCnet-ISA+ 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 written by the host and is not changed by the PCnet-ISA+ controller. | | | |
| 7-0 | HADR | The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and is not changed by the PCnet-ISA+ controller. | 14 | UFLO | BUFF error will turn off the transmitter (CSR0, TXON = 0). If a Buffer Error occurs, an Underflow Error will also occur. BUFF is not valid when LCOL or RTRY error is set during transmit data chaining. BUFF is written by the PCnet-ISA+ controller. UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the FIFO has emptied before the end of the frame was |

| | | | | | |
|----|------|---|-------|------|---|
| | | reached. Upon UFLO error, the transmitter is turned off (CSR0, TXON = 0). UFLO is written by the PCnet-ISA+ controller. | 10 | 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 = 1 in the MODE register, RTRY will set after one failed transmission attempt. RTRY is written by the PCnet-ISA+ controller. |
| 13 | RES | RESERVED bit. The PCnet-ISA+ controller will write this bit with a "0". | | | |
| 12 | LCOL | LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The PCnet-ISA+ controller does not re-try on late collisions. LCOL is written by the PCnet-ISA+ controller. | 09-00 | TDR | TIME DOMAIN REFLECTOMETRY reflects the state of an internal PCnet-ISA+ controller counter that counts at a 10 MHz rate from the start of a transmission to the occurrence of a collision or loss of carrier. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the PCnet-ISA+ controller and is valid only if RTRY is set. |
| 11 | LCAR | LOSS OF CARRIER is set in AUI mode when the carrier is lost during an PCnet-ISA+ controller-initiated transmission. The PCnet-ISA+ controller does not stop transmission upon loss of carrier. It will continue to transmit the whole frame until done. LCAR is written by the PCnet-ISA+ controller. In 10BASE-T mode, LCAR will be set when the T-MAU is in link fail state. | | | Note that 10 MHz gives very low resolution and in general has not been found to be particularly useful. This feature is here primarily to maintain full compatibility with the LANCE. |

Register Summary**Ethernet Controller Registers
(Accessed via RDP Port)**

| RAP Addr | Symbol | Width | User Register | Comments |
|----------|--------|--------|---------------|--|
| 00 | CSR0 | 16-bit | Y | PCnet-ISA+ controller status |
| 01 | CSR1 | 16-bit | Y | Lower IADR: maps to location 16 |
| 02 | CSR2 | 16-bit | Y | Upper IADR: maps to location 17 |
| 03 | CSR3 | 16-bit | Y | Mask Register |
| 04 | CSR4 | 16-bit | Y | Miscellaneous Register |
| 05 | CSR5 | 16-bit | | Reserved |
| 06 | CSR6 | 16-bit | | RXTX: RX/TX Encoded Ring Lengths |
| 07 | CSR7 | 16-bit | | Reserved |
| 08 | CSR8 | 16-bit | Y | LADR0: LADRF[15:0] |
| 09 | CSR9 | 16-bit | Y | LADR1: LADRF[31:16] |
| 10 | CSR10 | 16-bit | Y | LADR2: LADRF[47:32] |
| 11 | CSR11 | 16-bit | Y | LADR3: LADRF[63:48] |
| 12 | CSR12 | 16-bit | Y | PADR0: PADR[15:0] |
| 13 | CSR13 | 16-bit | Y | PADR1: PADR[31:16] |
| 14 | CSR14 | 16-bit | Y | PADR2: PADR[47:32] |
| 15 | CSR15 | 16-bit | Y | MODE: Mode Register |
| 16-17 | CSR16 | 32-bit | | IADR: Base Address of INIT Block |
| 18-19 | CSR18 | 32-bit | | CRBA: Current RCV Buffer Address |
| 20-21 | CSR20 | 32-bit | | CXBA: Current XMT Buffer Address |
| 22-23 | CSR22 | 32-bit | | NRBA: Next RCV Buffer Address |
| 24-25 | CSR24 | 32-bit | Y | BADR: Base Address of RCV Ring |
| 26-27 | CSR26 | 32-bit | | NRDA: Next RCV Descriptor Address |
| 28-29 | CSR28 | 32-bit | | CRDA: Current RCV Descriptor Address |
| 30-31 | CSR30 | 32-bit | Y | BADX: Base Address of XMT Ring |
| 32-33 | CSR32 | 32-bit | | NXDA: Next XMT Descriptor Address |
| 34-35 | CSR34 | 32-bit | | CXDA: Current XMT Descriptor Address |
| 36-37 | CSR36 | 32-bit | | Next Next Receive Descriptor Address |
| 38-39 | CSR38 | 32-bit | | Next Next Transmit Descriptor Address |
| 40-41 | CSR40 | 32-bit | | CRBC: Current RCV Stat and Byte Count |
| 42-43 | CSR42 | 32-bit | | CXBC: Current XMT Status and Byte Count |
| 44-45 | CSR44 | 32-bit | | NRBC: Next RCV Stat and Byte Count |
| 46 | CSR46 | 16-bit | | POLL: Poll Time Counter |
| 47 | CSR47 | 32-bit | Y | Polling Interval |
| 48-49 | CSR48 | 32-bit | | TMP0: Temporary Storage |
| 50-51 | CSR50 | 32-bit | | TMP1: Temporary Storage |
| 52-53 | CSR52 | 32-bit | | TMP2: Temporary Storage |
| 54-55 | CSR54 | 32-bit | | TMP3: Temporary Storage |
| 56-57 | CSR56 | 32-bit | | TMP4: Temporary Storage |
| 58-59 | CSR58 | 32-bit | | TMP5: Temporary Storage |
| 60-61 | CSR60 | 32-bit | | PXDA: Previous XMT Descriptor Address |
| 62-63 | CSR62 | 32-bit | | PXBC: Previous XMT Status and Byte Count |

Register Summary
**Ethernet Controller Registers
(Accessed via RDP Port) (continued)**

| RAP Addr | Symbol | Width | User Register | Comments |
|----------|--------|--------|---------------|--|
| 64-65 | CSR64 | 32-bit | | NXBA: Next XMT Buffer Address |
| 66-67 | CSR66 | 32-bit | | NXBC: Next XMT Status and Byte Count |
| 68-69 | CSR68 | 32-bit | | XSTMP: XMT Status Temporary |
| 70-71 | CSR70 | 32-bit | | RSTMP: RCV Status Temporary |
| 72 | CSR72 | 16-bit | | RCVRC: RCV Ring Counter |
| 74 | CSR74 | 16-bit | | XMTRC: XMT Ring Counter |
| 76 | CSR76 | 16-bit | Y | RCVRL: RCV Ring Length |
| 78 | CSR78 | 16-bit | Y | XMTRL: XMT Ring Length |
| 80 | CSR80 | 16-bit | Y | DMABR: Burst Register |
| 82 | CSR82 | 16-bit | Y | DMABAT: Bus Activity Timer |
| 84-85 | CSR84 | 32-bit | | DMABA: Address Register |
| 86 | CSR86 | 16-bit | | DMABC: Byte Counter/Register |
| 88-89 | CSR88 | 32-bit | Y | Chip ID Register |
| 92 | CSR92 | 16-bit | | RCON: Ring Length Conversion Register |
| 94 | CSR94 | 16-bit | | XMTTDR: Transmit Time Domain Reflectometry |
| 96-97 | CSR96 | 32-bit | | SCR0: BIU Scratch Register 0 |
| 98-99 | CSR98 | 32-bit | | SCR1: BIU Scratch Register 1 |
| 104-105 | CSR104 | 32-bit | | SWAP: 16-bit word/byte Swap Register |
| 108-109 | CSR108 | 32-bit | | BMSCR: BMU Scratch Register |
| 112 | CSR112 | 16-bit | Y | Missed Frame Count |
| 114 | CSR114 | 16-bit | Y | Receive Collision Count |
| 124 | CSR124 | 16-bit | Y | BMU Test Register |
| 126 | CSR126 | 16-bit | | Reserved |

Note: Although the PCnet-ISA* controller has many registers that can be accessed by software, most of these registers are intended for debugging and production testing purposes only. The registers with a "Y" are the only registers that should be accessed by network software.

Register Summary

ISACSR—ISA Bus Configuration Registers (Accessed via IDP Port)

| RAP Addr | Mnemonic | Default | Name |
|----------|----------|---------|---|
| 0 | MSRDA | 0005H | Master Mode Read Active |
| 1 | MSWRA | 0005H | Master Mode Write Active |
| 2 | MC | 0002H | Miscellaneous Configuration |
| 3 | EC | 8000*H | EEPROM Configuration |
| 4 | LED0 | 0000H | $\overline{\text{LED0}}$ Status (Link Integrity) |
| 5 | LED1 | 0084H | $\overline{\text{LED1}}$ Status (Default: RCV) |
| 6 | LED2 | 0008H | $\overline{\text{LED2}}$ Status (Default: RCVPOL) |
| 7 | LED3 | 0090H | $\overline{\text{LED3}}$ Status (Default: XMT) |
| 8 | SC | 0000H | Software Configuration (Read-Only Register) |

**This value can be 0000H for systems that do not support EEPROM option*

I/O Address Offset

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

SYSTEM APPLICATION

ISA Bus Interface

Compatibility Considerations

Although 8 MHz is now widely accepted as the standard speed at which to run the ISA bus, many machines have been built which operate at higher speeds with non-standard timing. Some machines do not correctly support 16-bit I/O operations with wait states. Although the PCnet-ISA+ controller is quite fast, some operations still require an occasional wait state. The PCnet-ISA+ controller moves data through memory accesses, therefore, I/O operations do not affect performance. By configuring the PCnet-ISA+ controller as an 8-bit I/O device, compatibility with PC/AT-class machines is obtained at virtually no cost in performance. To treat the PCnet-ISA+ controller as an 8-bit software resource (for non-ISA applications), the even-byte must be accessed first, followed by an odd-byte access.

Memory cycle timing is an area where some tradeoffs may be necessary. Any slow down in a memory cycle translates directly into lower bandwidth. The PCnet-ISA+ controller starts out with much higher bandwidth than most slave type controllers and should continue to be superior even if an extra 50 or 100 ns are added to memory cycles.

The memory cycle active time is tunable in 50 ns increments with a default of 250 ns. The memory cycle idle time defaults to 200 ns and can be reprogrammed to 100 ns. See register description for ISACS42. Most machines should not need tuning.

The PCnet-ISA+ controller is compatible with NE2100 and NE1500T software drivers. All the resources such

as address PROM, boot PROM, RAP, and RDP are in the same location with the same semantics. An additional set of registers (ISA CSR) is available to configure on board resources such as ISA bus timing and LED operation. However, loopback frames for the PCnet-ISA+ controller must contain more than 64 bytes of data if the Runt Packet Accept feature is not enabled; this size limitation does not apply to LANCE (Am7990) based boards such as the NE2100 and NE1500T.

Bus Master

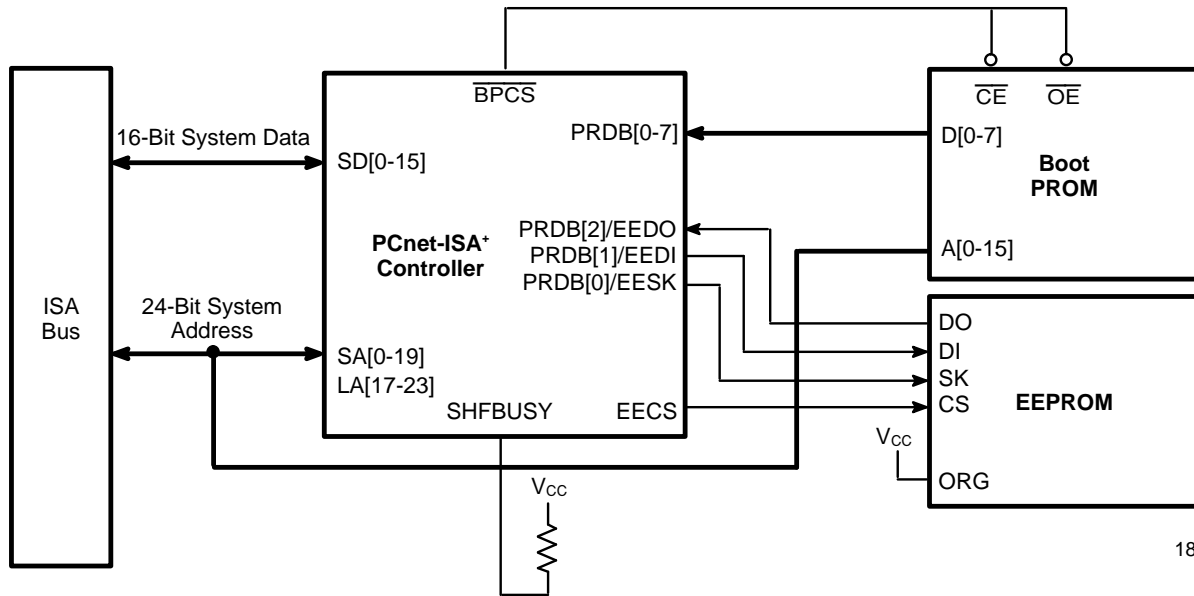
Bus Master mode is the preferred mode for client applications on PC/AT or similar machines supporting 16-bit DMA with its unsurpassed combination of high performance and low cost.

Shared Memory

The shared memory mode is recommended for file servers or other applications where there is very high, average or peak latency.

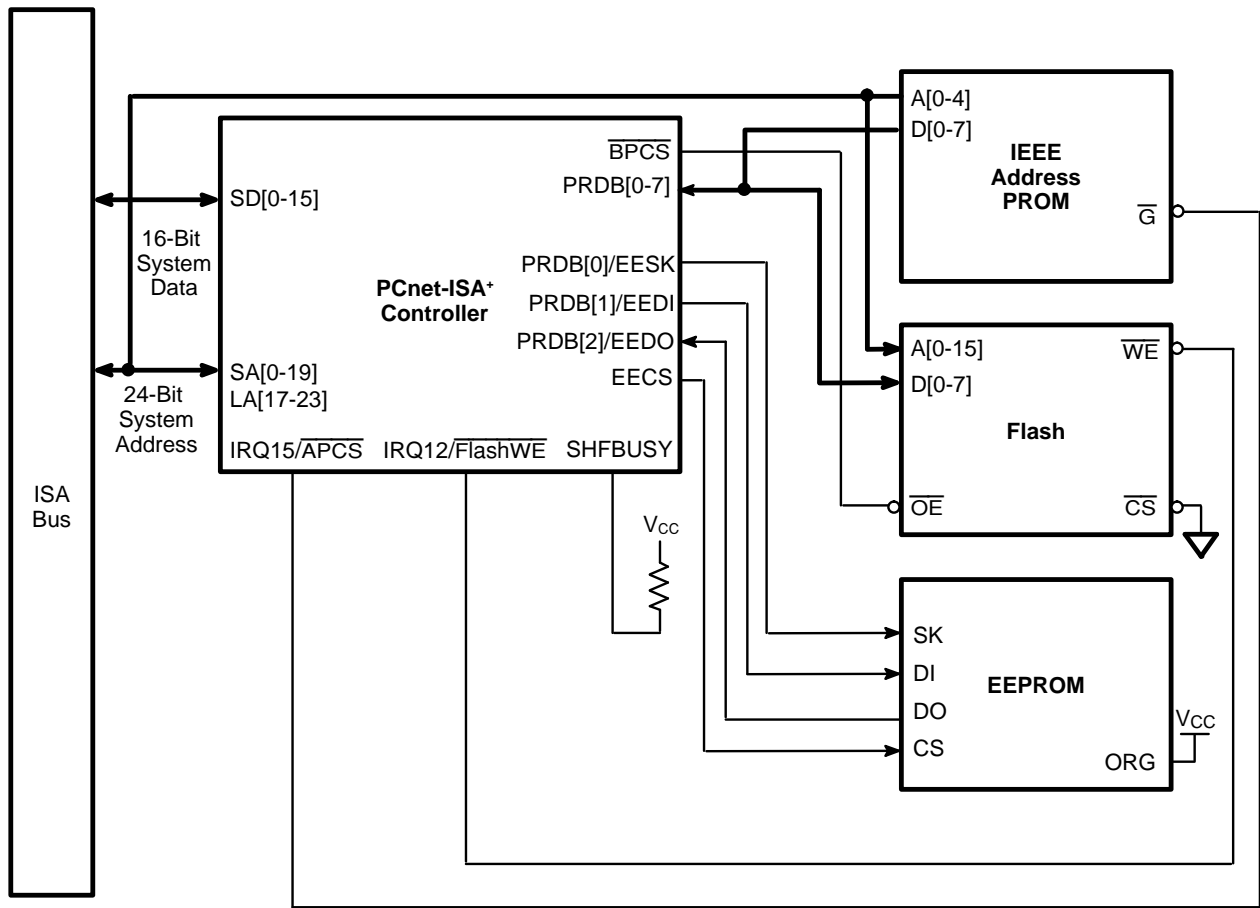
The address compare circuit has the following functions. It receives the 7 LA signals, generates MEMCS16, and compares them to the desired shared memory and boot PROM addresses. The logic latches the address compare result when BALE goes inactive and uses the appropriate SA signals to generate SMAM and BPAM.

All these functions can be performed in one PAL device. To operate in an 8-bit PC/XT environment, the LA signals should have weak pull-down resistors connected to them to present a logic 0 level when not driven.



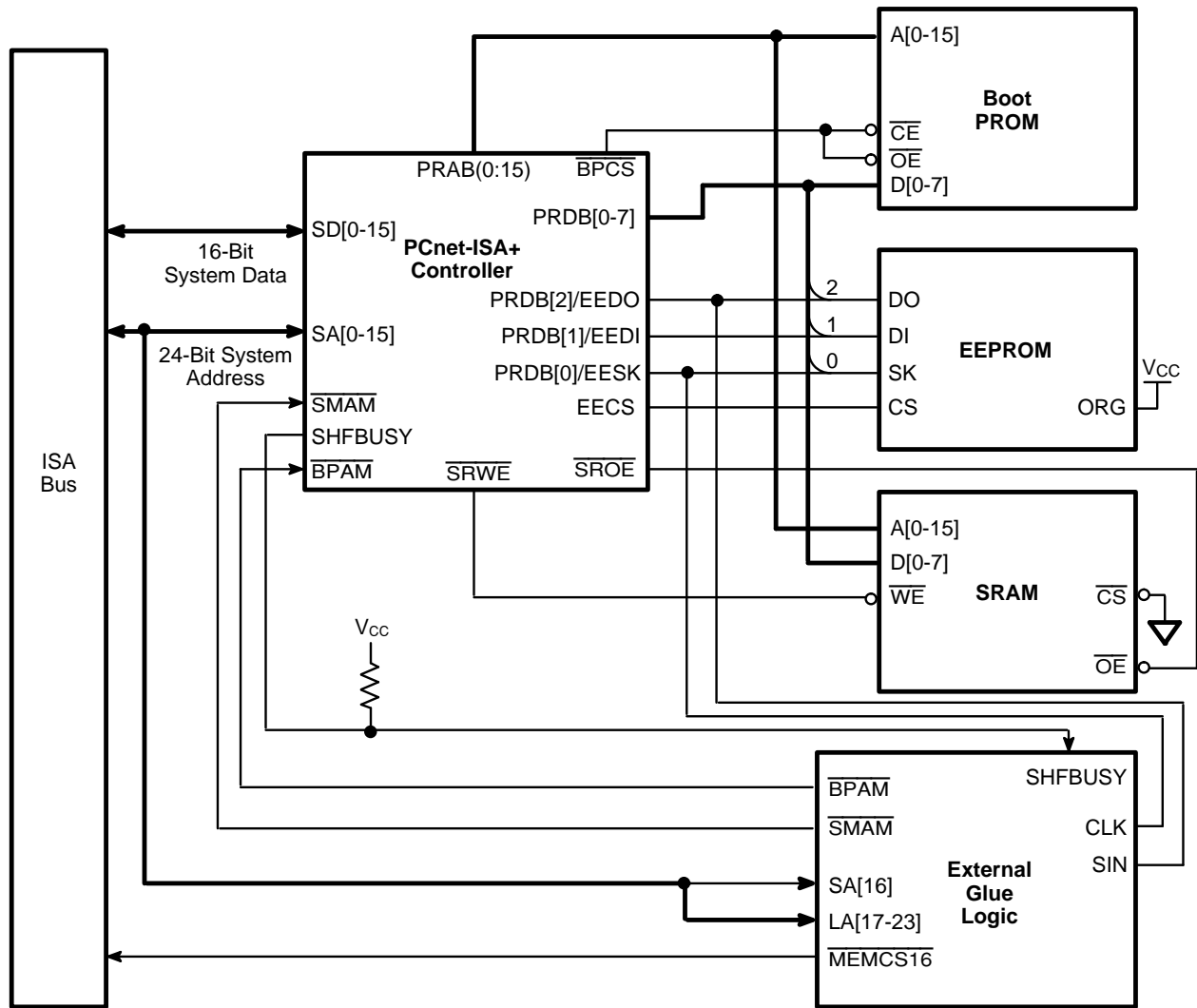
18183B-6

**Bus Master Block Diagram
Plug and Play Compatible**



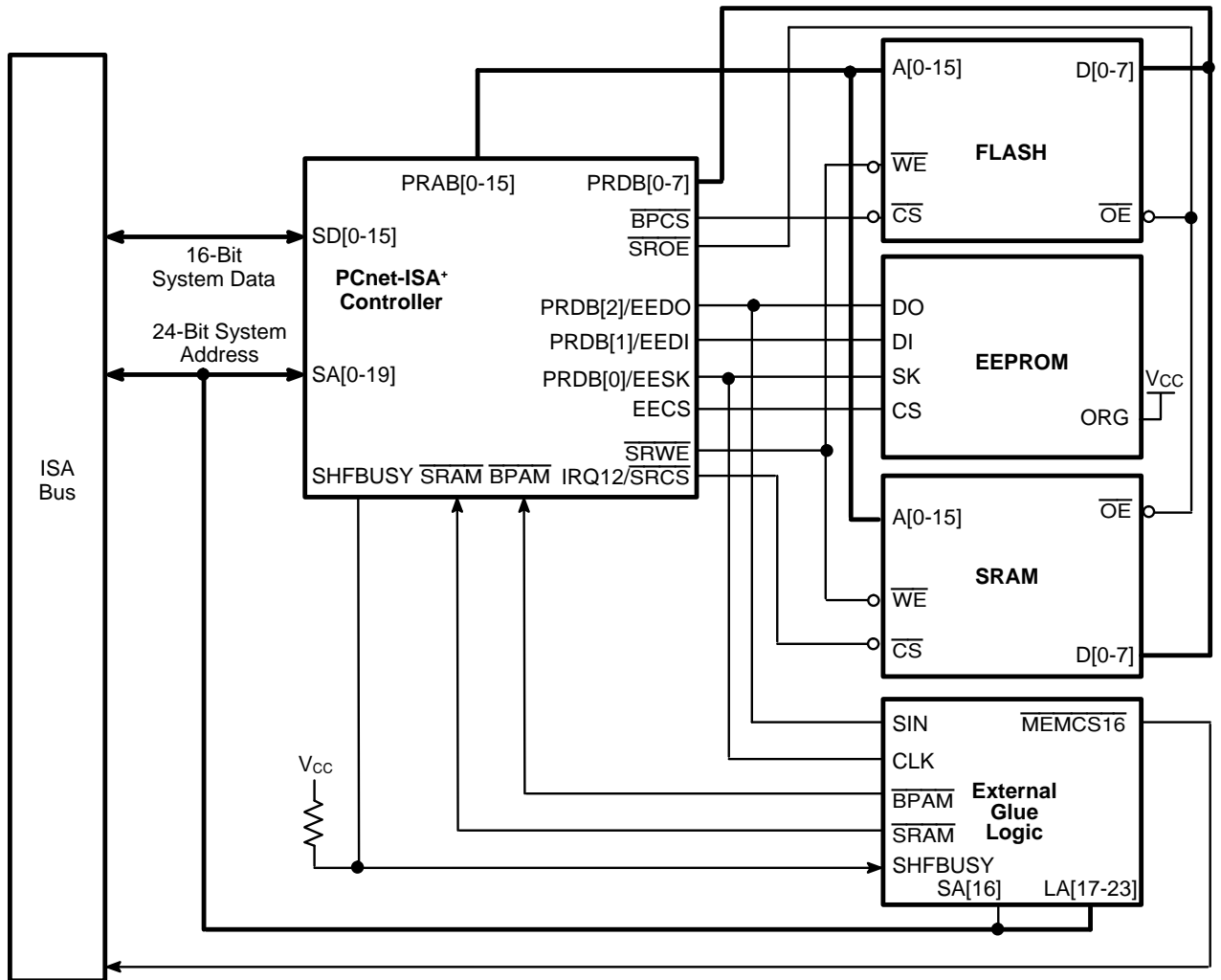
18183B-7

Bus Master Block Diagram
Plug and Play Compatible with Flash Support



18183B-9

Shared Memory Block Diagram
Plug and Play Compatible

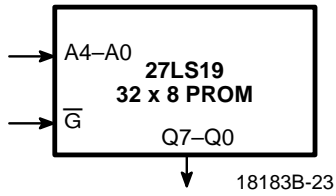


18183B-10

**Shared Memory Block Diagram
Plug and Play Compatible with Flash Memory Support**

Optional Address PROM Interface

The suggested address PROM is the Am27LS19, a 32x8 device. \overline{APCS} should be connected directly to the device's \overline{G} input.

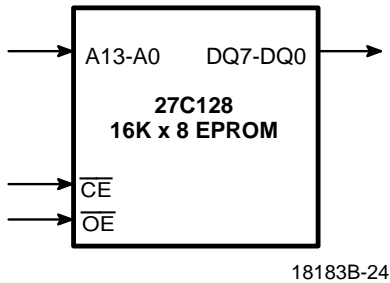


Address PROM Example

Boot PROM Interface

The boot PROM is a 8K–64K EPROM. Its \overline{OE} pin should be tied to ground, and chip enable \overline{CE} to \overline{BPCS} to minimize power consumption at the expense of speed. Shown below is a 27C128.

Higher density EPROMs place an address line on the pin that is defined for lower density EPROMs as the V_{PP} (programming voltage) pin. For READ only operation on an EPROM, the V_{PP} pin can assume any logic level, as long as the voltage on the V_{PP} pin does not exceed the programming voltage threshold (typically 7 V to 12 V). Therefore, a socket with a 27512 pinout will also support 2764 and 27128 EPROM devices.



Boot PROM Example

Static RAM Interface (for Shared Memory Only)

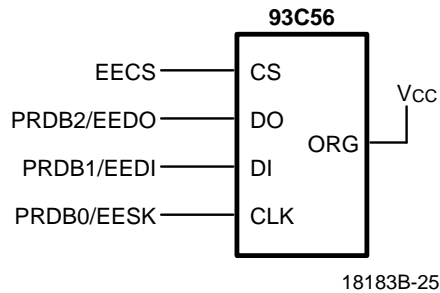
The SRAM is an 8Kx8 or 32Kx8 device. The PCnet-ISA+ controller can support 64 Kbytes of SRAM address space. The PCnet-ISA+ controller provides \overline{SROE} and \overline{SRWE} outputs which can go directly to the \overline{OE} and \overline{WE} pins of the SRAM, respectively. The address lines are connected as described in the shared memory section and the data lines go to the Private Data Bus.

AUI

The PCnet-ISA+ controller drives the AUI through a set of transformers. The DI and CI inputs should each be terminated with a pair of matched 39 Ω or 40.2 Ω resistors connected in series with the middle node bypassed to ground with a .01 μ F to 0.1 μ F capacitor. Refer to the PCnet-ISA Technical Manual (PID #16850B) for network interface design and refer to Appendix A for a list of compatible AUI isolation transformers.

EEPROM Interface

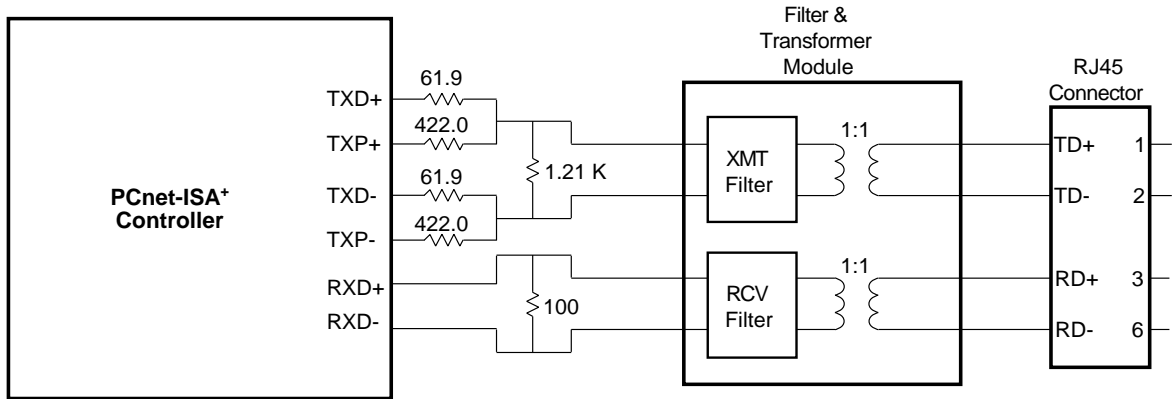
The suggested EEPROM is the industry standard 93C56 2 Kbit serial EEPROM. This is used in the 16-bit mode to provide 128 x 16-bit EEPROM locations to store configuration information as well as the Plug and Play information.



10BASE-T Interface

The diagram below shows the proper 10BASE-T network interface design. Refer to the *PCnet Family*

Technical Manual (PID #18216A) for more design details, and refer to Appendix A for a list of compatible 10BASE-T filter/transformer modules.



Note: All resistors are $\pm 1\%$

18183B-26

10BASE-T External Components and Hookup

ABSOLUTE MAXIMUM RATINGS

Storage Temperature -65°C to +150°C
 Ambient Temperature
 Under Bias 0°C to +70°C
 Supply Voltage to AV_{SS}
 or DV_{SS} (AV_{DD}, DV_{DD}) -0.3 V to +6.0 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A) 0°C to +70°C
 Supply Voltages
 (AV_{DD}, DV_{DD}) 5 V ±5%
 All inputs within the range: .. AV_{SS} - 0.5 V ≤ V_{in} ≤ AV_{DD} + 0.5 V, or DV_{SS} - 0.5 V ≤ V_{in} ≤ DV_{DD} + 0.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (refer to page 19 for driver types)

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit | |
|---------------------------------------|---|---|--------|------------------------|------|----|
| Digital Input Voltage | | | | | | |
| V _{IL} | Input LOW Voltage | | | 0.8 | V | |
| V _{IH} | Input HIGH Voltage | | 2.0 | DV _{DD} + 0.5 | V | |
| Digital Output Voltage | | | | | | |
| V _{OL} | Output LOW Voltage | | | 0.5 | V | |
| V _{OH} | Output HIGH Voltage | (Note 1) | 2.4 | | V | |
| Digital Input Leakage Current | | | | | | |
| I _{IX} | Input Leakage Current | V _{DD} = 5 V, V _{IN} = 0 V (Note 2) | -10 | 10 | μA | |
| Digital Output Leakage Current | | | | | | |
| I _{OZL} | Output Low Leakage Current (Note 3) | V _{OUT} = 0 V | -10 | | μA | |
| I _{OZH} | Output High Leakage Current (Note 3) | V _{OUT} = V _{DD} | | 10 | μA | |
| Crystal Input Current | | | | | | |
| V _{ILX} | XTAL1 Input LOW Threshold Voltage | V _{IN} = External Clock | -0.5 | 0.8 | V | |
| V _{ILHX} | XTAL1 Input HIGH Threshold Voltage | V _{IN} = External Clock | 3.5 | V _{DD} + 0.5 | V | |
| I _{ILX} | XTAL1 Input LOW Current | V _{IN} = DV _{SS} | Active | -120 | 0 | μA |
| | | | Sleep | -10 | +10 | μA |
| I _{IHX} | XTAL1 Input HIGH Current | V _{IN} = V _{DD} | Active | 0 | 120 | μA |
| | | | Sleep | | 400 | μA |
| Attachment Unit Interface | | | | | | |
| I _{IAXD} | Input Current at DI+ and DI- | AV _{SS} < V _{IN} < AV _{DD} | -500 | +500 | μA | |
| I _{IAXC} | Input current at CI+ and CI- | AV _{SS} < V _{IN} < AV _{DD} | -500 | +500 | μA | |
| V _{AOD} | Differential Output Voltage (DO+)-(DO-) | R _L = 78 Ω | 630 | 1200 | mV | |
| V _{AODOFF} | Transmit Differential Output Idle Voltage | R _L = 78 Ω (Note 5) | -40 | +40 | mV | |

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|--|---|--|-----------------|-----------------|-----------|
| Attachment Unit Interface (continued) | | | | | |
| IAODOFF | Transmit Differential Output Idle Current | $R_L = 78 \Omega$ (Note 4) | -1 | +1 | mA |
| VCMT | Transmit Output Common Mode Voltage | $R_L = 78 \Omega$ | 2.5 | AV_{DD} | V |
| VODI | DO_{\pm} Transmit Differential Output Voltage Imbalance | $R_L = 78 \Omega$ (Note 5) | | 25 | mV |
| VATH | Receive Data Differential Input Threshold | (Note 5) | -35 | 35 | mV |
| VASQ | DI_{\pm} and CI_{\pm} Differential Input Threshold (Squelch) | | -275 | -160 | mV |
| VIRDVD | DI_{\pm} and CI_{\pm} Differential Mode Input Voltage Range | | -1.5 | +1.5 | V |
| VICM | DI_{\pm} and CI_{\pm} Input Bias Voltage | $I_{IN} = 0$ mA | $AV_{DD}-3.0$ | $AV_{DD}-1.0$ | V |
| VOPD | DO_{\pm} Undershoot Voltage at Zero Differential on Transmit Return to Zero (ETD) | (Note 5) | | -100 | mV |
| Twisted Pair Interface | | | | | |
| IIRXD | Input Current at RXD_{\pm} | $AV_{SS} < V_{IN} < AV_{DD}$ | -500 | 500 | μA |
| RRXD | RXD_{\pm} Differential Input Resistance | (Note 5) | 10 | | $K\Omega$ |
| VTIVB | $RXD+$, $RXD-$ Open Circuit Input Voltage (Bias) | $I_{IN} = 0$ mA | $AV_{DD} - 3.0$ | $AV_{DD} - 1.5$ | V |
| VTIDV | Differential Mode Input Voltage Range (RXD_{\pm}) | $AV_{DD} = +5$ V | -3.1 | +3.1 | V |
| VTSQ+ | RXD Positive Squelch Threshold (Peak) | Sinusoid $5 \text{ MHz} \leq f \leq 10 \text{ MHz}$ | 300 | 520 | mV |
| VTSQ- | RXD Negative Squelch Threshold (Peak) | Sinusoid $5 \text{ MHz} \leq f \leq 10 \text{ MHz}$ | -520 | -300 | mV |
| VTHS+ | RXD Post-Squelch Positive Threshold (Peak) | Sinusoid $5 \text{ MHz} \leq f \leq 10 \text{ MHz}$ | 150 | 293 | mV |
| VTHS- | RXD Post-Squelch Negative Threshold (Peak) | Sinusoid $5 \text{ MHz} \leq f \leq 10 \text{ MHz}$ | -293 | -150 | mV |
| VLTSQ+ | RXD Positive Squelch Threshold (Peak) | $LRT = 1$ (Note 6) | 180 | 312 | mV |
| VLTSQ- | RXD Negative Squelch Threshold (Peak) | $LRT = 1$ (Note 6) | -312 | -180 | mV |
| VLTHS+ | RXD Post-Squelch Positive Threshold (Peak) | $LRT = 1$ (Note 6) | 90 | 156 | mV |
| VLTHS- | RXD Post-Squelch Negative Threshold (Peak) | $LRT = 1$ (Note 6) | -156 | -90 | mV |

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|---|---|---|------------------------|------------------------|------|
| Twisted Pair Interface (continued) | | | | | |
| VRXDTH | RXD Switching Threshold | (Note 5) | -35 | 35 | mV |
| V _{TXH} | TXD _± and TXP _± Output HIGH Voltage | DV _{SS} = 0 V | DV _{DD} - 0.6 | DV _{DD} | V |
| V _{TXL} | TXD _± and TXP _± Output LOW Voltage | DV _{DD} = +5 V | DV _{SS} | DV _{SS} + 0.6 | V |
| V _{TXI} | TXD _± and TXP _± Differential Output Voltage Imbalance | | -40 | +40 | mV |
| V _{TXOFF} | TXD _± and TXP _± Idle Output Voltage | DV _{DD} = +5 V | -40 | +40 | mV |
| R _{TX} | TXD _± Differential Driver Output Impedance | (Note 5) | | 40 | Ω |
| | TXP _± Differential Driver Output Impedance | (Note 5) | | 80 | Ω |
| IEEE 1149.1 (JTAG) Test Port | | | | | |
| V _{IL} | TCK, TMS, TDI | | | 0.8 | V |
| V _{IH} | TCK, TMS, TDI | | 2.0 | | V |
| V _{OL} | TDO | I _{OL} = 2.0 mA | | 0.4 | V |
| V _{OH} | TDO | I _{OH} = -0.4 mA | 2.4 | | V |
| I _{IL} | TCK, TMS, TDI | V _{DD} = 5.5 V, V _I = 0.5 V | | -200 | μA |
| I _{IH} | TCK, TMS, TDI | V _{DD} = 5.5 V, V _I = 2.7 V | | -100 | μA |
| I _{OZ} | TDO | 0.4 V < V _{OUT} < V _{DD} | -10 | +10 | μA |
| Power Supply Current | | | | | |
| I _{DD} | Active Power Supply Current | XTAL1 = 20 MHz | | 75 | mA |
| I _{DDCOMA} | Coma Mode Power Supply Current | $\overline{\text{SLEEP}}$ active | | 200 | μA |
| I _{DDSNOOZE} | Snooze Mode Mail Power Supply Current | Awake bit set active | | 10 | mA |

Notes:

1. V_{OH} does not apply to open-drain output pins.
2. I_{Ix} applies to all input only pins except DI_±, CI_±, XTAL1 and PRDB[7:0].
3. I_{OZL} applies to all three-state output pins and bi-directional pins, except PRDB[7:0]. I_{OZH} applies to pins PRDB[7:0].
4. Correlated to other tested parameters—not tested directly.
5. Parameter not tested.
6. LRT is bit 9 of Mode register (CSR15)

SWITCHING CHARACTERISTICS: BUS MASTER MODE

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|---------------------------------------|---|-----------------|------|-----|------|
| Input/Output Write Timing | | | | | |
| t _{IOW1} | AEN, $\overline{\text{SBHE}}$, SA0–9 Setup to $\downarrow \overline{\text{IOW}}$ | | 10 | | ns |
| t _{IOW2} | AEN, $\overline{\text{SBHE}}$, SA0–9 Hold After $\uparrow \overline{\text{IOW}}$ | | 5 | | ns |
| t _{IOW3} | $\overline{\text{IOW}}$ Assertion | | 100 | | ns |
| t _{IOW4} | $\overline{\text{IOW}}$ Inactive | | 55 | | ns |
| t _{IOW5} | SD Setup to $\uparrow \overline{\text{IOW}}$ | | 10 | | ns |
| t _{IOW6} | SD Hold After $\uparrow \overline{\text{IOW}}$ | | 10 | | ns |
| t _{IOW7} | $\downarrow \text{IOCHRDY}$ Delay From $\downarrow \overline{\text{IOW}}$ | | 0 | 35 | ns |
| t _{IOW8} | IOCHRDY Inactive | | 125 | | ns |
| t _{IOW9} | $\uparrow \text{IOCHRDY}$ to $\uparrow \overline{\text{IOW}}$ | | 0 | | ns |
| Input/Output Read Timing | | | | | |
| t _{IOR1} | AEN, $\overline{\text{SBHE}}$, SA0–9 Setup to $\downarrow \overline{\text{IOR}}$ | | 15 | | ns |
| t _{IOR2} | AEN, $\overline{\text{SBHE}}$, SA0–9 Hold After $\uparrow \overline{\text{IOR}}$ | | 5 | | ns |
| t _{IOR3} | $\overline{\text{IOR}}$ Inactive | | 55 | | ns |
| t _{IOR4} | SD Hold After $\uparrow \overline{\text{IOR}}$ | | 0 | 20 | ns |
| t _{IOR5} | SD Valid From $\downarrow \overline{\text{IOR}}$ | | 0 | 110 | ns |
| t _{IOR6} | $\downarrow \text{IOCHRDY}$ Delay From $\downarrow \overline{\text{IOR}}$ | | 0 | 35 | ns |
| t _{IOR7} | IOCHRDY Inactive | | 125 | | ns |
| t _{IOR8} | SD Valid From $\uparrow \text{IOCHRDY}$ | | –130 | 10 | ns |
| I/O To Memory Command Inactive | | | | | |
| t _{IOM1} | $\uparrow \overline{\text{IOW/MEMW}}$ to $\downarrow (\overline{\text{S}})\text{MEMR/IOR}$ | | 55 | | ns |
| t _{IOM2} | $\uparrow (\overline{\text{S}})\text{MEMR/IOR}$ to $\downarrow \overline{\text{IOW/MEMW}}$ | | 55 | | ns |
| IOCS16 Timing | | | | | |
| t _{IOCS1} | AEN, $\overline{\text{SBHE}}$, SA0–9 to $\downarrow \overline{\text{IOCS16}}$ | | 0 | 35 | ns |
| t _{IOCS2} | AEN, $\overline{\text{SBHE}}$, SA0–9 to $\overline{\text{IOCS16}}$ Tristated | | 0 | 25 | ns |
| Master Mode Bus Acquisition | | | | | |
| t _{MMA1} | $\overline{\text{REF}}$ Inactive to $\downarrow \overline{\text{DACK}}$ | | 5 | | ns |
| t _{MMA2} | $\uparrow \text{DRQ}$ to $\downarrow \overline{\text{DACK}}$ | | 0 | | ns |
| t _{MMA3} | $\overline{\text{DACK}}$ Inactive | | 55 | | ns |
| t _{MMA4} | $\downarrow \overline{\text{DACK}}$ to $\downarrow \overline{\text{MASTER}}$ | | | 35 | ns |
| t _{MMA5} | $\downarrow \overline{\text{MASTER}}$ to Active Command, $\overline{\text{SBHE}}$, SA0–19, LA17–23 | | 125 | 185 | ns |

SWITCHING CHARACTERISTICS: BUS MASTER MODE (continued)

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|--------------------------------|---|-----------------|-------------|--------------|------|
| Master Mode Bus Release | | | | | |
| tMMBR1 | Command Deassert to ↓ DRQ | | 45 | 65 | ns |
| tMMBR2 | ↓ DRQ to ↑ $\overline{\text{DACK}}$ | | 0 | | ns |
| tMMBR3 | ↓ DRQ to ↑ $\overline{\text{MASTER}}$ | | 40 | 60 | ns |
| tMMBR4 | ↓ DRQ to Command, $\overline{\text{SBHE}}$, SA0–19, LA17–23 Tristated | | –15 | 0 | ns |
| Master Write Cycles | | | | | |
| tMMW1 | $\overline{\text{SBHE}}$, SA0–19, LA17–23, Active to ↓ $\overline{\text{MEMW}}$ | (Note 1) | EXTIME + 45 | EXTIME + 65 | ns |
| tMMW2 | $\overline{\text{MEMW}}$ Active | (Note 2) | MSWRA – 10 | MSWRA + 5 | ns |
| tMMW3 | $\overline{\text{MEMW}}$ Inactive | (Note 1) | EXTIME + 97 | EXTIME + 105 | ns |
| tMMW4 | ↑ $\overline{\text{MEMW}}$ to $\overline{\text{SBHE}}$, SA0–19, LA17–23, SD Inactive | | 45 | 55 | ns |
| tMMW5 | $\overline{\text{SBHE}}$, SA0–19, LA17–23, SD Hold After ↑ $\overline{\text{MEMW}}$ | | 45 | 60 | ns |
| tMMW6 | $\overline{\text{SBHE}}$, SA0–19, LA17–23, SD Setup to ↓ $\overline{\text{MEMW}}$ | (Note 1) | EXTIME + 45 | EXTIME + 55 | ns |
| tMMW7 | ↓ IOCHRDY Delay From ↓ $\overline{\text{MEMW}}$ | | tMMW2 – 175 | | ns |
| tMMW8 | IOCHRDY Inactive | | 55 | | ns |
| tMMW9 | ↑ IOCHRDY to ↑ $\overline{\text{MEMW}}$ | | 130 | | ns |
| tMMW10 | SD Active to ↓ $\overline{\text{MEMW}}$ | (Note 1) | EXTIME + 20 | EXTIME + 60 | ns |
| tMMW11 | SD Setup to ↓ $\overline{\text{MEMW}}$ | (Note 1) | EXTIME + 20 | EXTIME + 60 | ns |
| Master Read Cycles | | | | | |
| tMMR1 | $\overline{\text{SBHE}}$, SA0–19, LA17–23, Active to ↓ $\overline{\text{MEMR}}$ | (Note 1) | EXTIME + 45 | EXTIME + 60 | ns |
| tMMR2 | $\overline{\text{MEMR}}$ Active | (Note 2) | MSRDA – 10 | MSRDA + 5 | ns |
| tMMR3 | $\overline{\text{MEMR}}$ Inactive | (Note 1) | EXTIME + 97 | EXTIME + 105 | ns |
| tMMR4 | ↑ $\overline{\text{MEMR}}$ to $\overline{\text{SBHE}}$, SA0–19, LA17–23 Inactive | | 45 | 55 | ns |
| tMMR5 | $\overline{\text{SBHE}}$, SA0–19, LA17–23 Hold After ↑ $\overline{\text{MEMW}}$ | | 45 | 55 | ns |
| tMMR6 | $\overline{\text{SBHE}}$, SA0–19, LA17–23 Setup to ↓ $\overline{\text{MEMR}}$ | (Note 1) | EXTIME + 45 | EXTIME + 55 | ns |
| tMMR7 | ↓ IOCHRDY Delay From ↓ $\overline{\text{MEMR}}$ | | tMMR2 – 175 | | ns |
| tMMR8 | IOCHRDY Inactive | | 55 | | ns |
| tMMR9 | ↑ IOCHRDY to ↑ $\overline{\text{MEMR}}$ | | 130 | | ns |
| tMMR10 | SD Setup to ↑ $\overline{\text{MEMR}}$ | | 30 | | ns |
| tMMR11 | SD Hold After ↑ $\overline{\text{MEMR}}$ | | 0 | | ns |

SWITCHING CHARACTERISTICS: BUS MASTER MODE (continued)

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|--------------------------------------|---|-----------------|-----|-----|------|
| Master Mode Address PROM Read | | | | | |
| tMA1 | $\downarrow \overline{IO\overline{R}}$ to $\downarrow \overline{APCS}$ | | 125 | 260 | ns |
| tMA2 | \overline{APCS} Active | | 140 | 155 | ns |
| tMA3 | PRDB Setup to $\uparrow \overline{APCS}$ | | 20 | | ns |
| tMA4 | PRDB Hold After $\uparrow \overline{APCS}$ | | 0 | | ns |
| tMA5 | $\uparrow \overline{APCS}$ to $\uparrow \overline{IOCHR\overline{DY}}$ | | 45 | 65 | ns |
| tMA6 | SD Valid From $\uparrow \overline{IOCHR\overline{DY}}$ | | 0 | 10 | ns |
| Master Mode Boot PROM Read | | | | | |
| tMB1 | \overline{REF} , \overline{SBHE} , SA0–19 Setup to $\downarrow \overline{SMEM\overline{R}}$ | | 10 | | ns |
| tMB2 | \overline{REF} , \overline{SBHE} , SA0–19 Hold $\uparrow \overline{SMEM\overline{R}}$ | | 5 | | ns |
| tMB3 | $\downarrow \overline{IOCHR\overline{DY}}$ Delay From $\downarrow \overline{SMEM\overline{R}}$ | | 0 | 35 | ns |
| tMB4 | $\overline{SMEM\overline{R}}$ Inactive | | 55 | | ns |
| tMB5 | $\downarrow \overline{SMEM\overline{R}}$ to $\downarrow \overline{BPCS}$ | | 125 | 260 | ns |
| tMB6 | \overline{BPCS} Active | | 290 | 305 | ns |
| tMB7 | $\uparrow \overline{BPCS}$ to $\uparrow \overline{IOCHR\overline{DY}}$ | | 45 | 65 | ns |
| tMB8 | PRDB Setup to $\uparrow \overline{BPCS}$ | | 20 | | ns |
| tMB9 | PRDB Hold After $\uparrow \overline{BPCS}$ | | 0 | | ns |
| tMB10 | SD Valid From $\uparrow \overline{IOCHR\overline{DY}}$ | | 0 | 10 | ns |
| tMB11 | SD Hold After $\uparrow \overline{SMEM\overline{R}}$ | | 0 | 20 | ns |
| tMB12 | LA20–23 Hold From $\downarrow \overline{BALE}$ | | 10 | | ns |
| tMB13 | LA20–23 Setup to $\downarrow \overline{MEM\overline{R}}$ | | 10 | | ns |
| tMB14 | $\uparrow \overline{BALE}$ Setup to $\downarrow \overline{MEM\overline{R}}$ | | 10 | | ns |

Notes:

1. *EXTIME* is 100 ns when *ISACSR2*, bit 4, is cleared (default). *EXTIME* is 0 ns when *ISACSR2*, bit 4, is set.
2. *MSRDA* and *MSWDA* are parameters which are defined in registers *ISACSR0* and *ISACSR1*, respectively.

SWITCHING CHARACTERISTICS: BUS MASTER MODE—FLASH READ CYCLE

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------|--|-----------------|-----|-----|------|
| tMFR1 | REF, SBHE, SA0–19 Setup to \downarrow MEMR | | 10 | | ns |
| tMFR2 | REF, SBHE, SA0–19 Hold From \uparrow MEMR | | 5 | | ns |
| tMFR3 | \downarrow IOCHRDY to $\overline{\text{MEMR}}$ | | 0 | 35 | ns |
| tMFR4 | \downarrow $\overline{\text{MEMR}}$ Inactive | | 55 | | ns |
| tMFR5 | \downarrow $\overline{\text{MEMR}}$ to \downarrow $\overline{\text{BPCS}}$ | | 125 | 260 | ns |
| tMFR6 | $\overline{\text{BPCS}}$ Active | | 190 | 205 | ns |
| tMFR7 | \uparrow $\overline{\text{BPCS}}$ to \uparrow IOCHRDY | | 45 | 65 | ns |
| tMFR8 | PRDB Setup to \uparrow of $\overline{\text{BPCS}}$ | | 20 | | ns |
| tMFR9 | PRDB Hold to \uparrow of $\overline{\text{BPCS}}$ | | 0 | | ns |
| tMFR10 | SD Valid From \uparrow IOCHRDY | | 0 | 10 | ns |
| tMFR11 | SD Tristate to \uparrow $\overline{\text{MEMR}}$ | | 0 | 20 | ns |
| tMFR12 | LA20–23 Hold From \downarrow BALE | | 10 | | ns |
| tMFR13 | LA20–23 Setup to \downarrow $\overline{\text{MEMR}}$ | | 10 | | ns |
| tMFR14 | \uparrow BALE Setup to \downarrow $\overline{\text{MEMR}}$ | | 15 | | ns |

SWITCHING CHARACTERISTICS: BUS MASTER MODE—FLASH WRITE CYCLE

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------|--|-----------------|-----|-----|------|
| tMFW1 | SBHE, SA0–19 Setup to \downarrow MEMW | | 10 | | ns |
| tMFW2 | SBHE, SA0–19 Hold From \uparrow MEMW | | 5 | | ns |
| tMFW3 | \downarrow IOCHRDY to \downarrow $\overline{\text{MEMW}}$ | | 0 | 35 | ns |
| tMFW4 | $\overline{\text{MEMW}}$ Inactive | | 50 | | ns |
| tMFW5 | \uparrow FL_WE to \uparrow IOCHRDY | | 20 | 90 | ns |
| tMFW6 | \uparrow $\overline{\text{MEMW}}$ Hold From \uparrow IOCHRDY | | 0 | | ns |
| tMFW7 | SD Valid From \downarrow $\overline{\text{MEMW}}$ | | | 175 | ns |
| tMFW8 | SD Hold From \uparrow $\overline{\text{MEMW}}$ | | 0 | | ns |
| tMFW9 | PRDB Valid From \downarrow $\overline{\text{MEMW}}$ | | | 175 | ns |
| tMFW10 | PRDB Setup to \downarrow FL_WE | | 15 | | ns |
| tMFW11 | FL_WE Active | | 140 | 155 | ns |
| tMFW12 | PRDB Hold From \uparrow FL_WE | | 15 | | ns |
| tMFW13 | LA20–23 Hold From \downarrow BALE | | 10 | | ns |
| tMFW14 | LA20–23 Setup to \downarrow $\overline{\text{MEMW}}$ | | 10 | | ns |
| tMFW15 | \uparrow BALE Setup to \downarrow $\overline{\text{MEMW}}$ | | 15 | | ns |

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|----------------------------------|--|-----------------|------|-----|------|
| Input/Output Write Timing | | | | | |
| t _{IOW1} | AEN, \overline{SBHE} , SA0–9 Setup to $\downarrow \overline{IOW}$ | | 10 | | ns |
| t _{IOW2} | AEN, \overline{SBHE} , SA0–9 Hold From $\uparrow \overline{IOW}$ | | 5 | | ns |
| t _{IOW3} | \overline{IOW} Assertion | | 150 | | ns |
| t _{IOW4} | \overline{IOW} Inactive | | 55 | | ns |
| t _{IOW5} | SD Setup to $\uparrow \overline{IOW}$ | | 10 | | ns |
| t _{IOW6} | SD Hold After $\uparrow \overline{IOW}$ | | 10 | | ns |
| t _{IOW7} | \downarrow IOCHRDY Delay From $\downarrow \overline{IOW}$ | | 0 | 35 | ns |
| t _{IOW8} | IOCHRDY Inactive | | 125 | | ns |
| t _{IOW9} | \uparrow IOCHRDY to $\uparrow \overline{IOW}$ | | 0 | | ns |
| Input/Output Read Timing | | | | | |
| t _{IOR1} | AEN, \overline{SBHE} , SA0–9 Setup to $\downarrow \overline{IOR}$ | | 15 | | ns |
| t _{IOR2} | AEN, \overline{SBHE} , SA0–9 Hold After $\uparrow \overline{IOR}$ | | 5 | | ns |
| t _{IOR3} | \overline{IOR} Inactive | | 55 | | ns |
| t _{IOR4} | SD Hold From $\uparrow \overline{IOR}$ | | 0 | 20 | ns |
| t _{IOR5} | SD Valid From $\downarrow \overline{IOR}$ | | 0 | 110 | ns |
| t _{IOR6} | \downarrow IOCHRDY Delay From $\downarrow \overline{IOR}$ | | 0 | 35 | ns |
| t _{IOR7} | IOCHRDY Inactive | | 125 | | ns |
| t _{IOR8} | SD Valid From \uparrow IOCHRDY | | –130 | 10 | ns |
| Memory Write Timing | | | | | |
| t _{MW1} | SA0–15, \overline{SBHE} , $\downarrow \overline{SMAM}$ Setup to $\downarrow \overline{MEMW}$ | | 10 | | ns |
| t _{MW2} | SA0–15, \overline{SBHE} , $\uparrow \overline{SMAM}$ Hold From $\uparrow \overline{MEMW}$ | | 5 | | ns |
| t _{MW3} | \overline{MEMW} Assertion | | 150 | | ns |
| t _{MW4} | \overline{MEMW} Inactive | | 55 | | ns |
| t _{MW5} | SD Setup to $\uparrow \overline{MEMW}$ | | 10 | | ns |
| t _{MW6} | SD Hold From $\uparrow \overline{MEMW}$ | | 10 | | ns |
| t _{MW7} | \downarrow IOCHRDY Delay From $\downarrow \overline{MEMW}$ | | 0 | 35 | ns |
| t _{MW8} | IOCHRDY Inactive | | 125 | | ns |
| t _{MW9} | $\uparrow \overline{MEMW}$ to \uparrow IOCHRDY | | 0 | | ns |

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE (continued)

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|--|---|-----------------|------|-----|------|
| Memory Read Timing | | | | | |
| t _{MR1} | SA0–15, $\overline{\text{SBHE}}$, $\downarrow \overline{\text{SMAM/BPAM}}$ Setup to $\downarrow \overline{\text{MEMR}}$ | | 10 | | ns |
| t _{MR2} | SA0–15, $\overline{\text{SBHE}}$, $\uparrow \overline{\text{SMAM/BPAM}}$ Hold From $\uparrow \overline{\text{MEMR}}$ | | 5 | | ns |
| t _{MR3} | $\overline{\text{MEMR}}$ Inactive | | 55 | | ns |
| t _{MR4} | SD Hold From $\uparrow \overline{\text{MEMR}}$ | | 0 | 20 | ns |
| t _{MR5} | SD Valid From $\downarrow \overline{\text{MEMR}}$ | | 0 | 110 | ns |
| t _{MR6} | $\downarrow \text{IOCHRDY}$ Delay From $\downarrow \overline{\text{MEMR}}$ | | 0 | 35 | ns |
| t _{MR7} | IOCHRDY Inactive | | 125 | | ns |
| t _{MR8} | SD Valid From $\uparrow \text{IOCHRDY}$ | | –130 | 10 | ns |
| I/O To Memory Command Inactive | | | | | |
| t _{IOM1} | $\downarrow \overline{\text{IOW/MEMW}}$ to $\downarrow (\overline{\text{S}})\overline{\text{MEMR/IOR}}$ | | 55 | | ns |
| t _{IOM2} | $\downarrow (\overline{\text{S}})\overline{\text{MEMR/IOR}}$ to $\downarrow \overline{\text{IOW/MEMW}}$ | | 55 | | ns |
| IOCS16 Timing | | | | | |
| t _{IOCS1} | AEN, $\overline{\text{SBHE}}$, SA0–9 to $\downarrow \overline{\text{IOCS16}}$ | | 0 | 35 | ns |
| t _{IOCS2} | AEN, $\overline{\text{SBHE}}$, SA0–9 to $\overline{\text{IOCS16}}$ Tristated | | 0 | 25 | ns |
| SRAM Read/Write, Boot PROM Read, Address PROM Read on Private Bus | | | | | |
| t _{PR4} | PRAB Change to PRAB Change, SRAM Access | | 95 | 105 | ns |
| t _{PR5} | PRDB Setup to PRAB Change, SRAM Access | | 20 | | ns |
| t _{PR6} | PRDB Hold From PRAB Change, SRAM Access | | 0 | | ns |
| t _{PR7} | PRAB Change to PRAB Change, APROM Access | | 145 | 155 | ns |

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE (continued)

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|--|---|-----------------|-----|-----|------|
| SRAM Read/Write, Boot PROM Read, Address PROM Read on Private Bus (continued) | | | | | |
| t _{PR8} | PRDB Setup to PRAB Change, APROM Access | | 20 | | ns |
| t _{PR9} | PRDB Hold After PRAB Change, APROM Access | | 0 | | ns |
| t _{PR10} | PRAB Change to PRAB Change, BPROM Access | | 290 | 305 | ns |
| t _{PR11} | PRDB Setup to PRAB Change, BPROM Access | | 20 | | ns |
| t _{PR12} | PRDB Hold After PRAB Change, BPROM Access | | 0 | | ns |
| t _{PR13} | PRAB Change to PRAB Change, SRAM Write | | 145 | 155 | ns |
| t _{PR14} | PRAB Change to $\downarrow \overline{SRWE}$ | | 20 | 30 | ns |
| t _{PR15} | PRAB Change to $\uparrow \overline{SRWE}$ | | 120 | 130 | ns |
| t _{PR16} | PRAB Change to PRAB Change, Flash Access | | 190 | 205 | ns |
| t _{PR17} | PRAB Change to PRAB Change, Flash Write | | 190 | 205 | ns |
| t _{PR18} | PRAB Change to $\uparrow \overline{SRWE}$ | | 170 | 180 | ns |

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE—FLASH READ CYCLE

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------|---|-----------------|-----|-----|------|
| tMFR1 | BPAM, REF, SBHE, SA0–19 Setup to $\downarrow \overline{\text{MEMR}}$ | | 10 | | ns |
| tMFR2 | BPAM, REF, SBHE, SA0–19 Hold From $\uparrow \overline{\text{MEMR}}$ | | 5 | | ns |
| tMFR3 | $\downarrow \text{IOCHRDY}$ to $\downarrow \overline{\text{MEMR}}$ | | 0 | 35 | ns |
| tMFR4 | $\overline{\text{MEMR}}$ Inactive | | 55 | | ns |
| tMFR5 | $\downarrow \overline{\text{MEMR}}$ to $\downarrow \overline{\text{BPCS/SROE}}$ | | 125 | 260 | ns |
| tMFR6 | $\overline{\text{BPCS/SROE}}$ Active | | 190 | 205 | ns |
| tMFR7 | $\uparrow \overline{\text{BPCS/SROE}}$ to $\uparrow \text{IOCHRDY}$ | | 45 | 65 | ns |
| tMFR8 | PRDB Setup to \uparrow of $\overline{\text{BPCS/SROE}}$ | | 20 | | ns |
| tMFR9 | PRDB Hold to \uparrow of $\overline{\text{BPCS/SROE}}$ | | 0 | | ns |
| tMFR10 | SD Valid From $\uparrow \text{IOCHRDY}$ | | 0 | 10 | ns |
| tMFR11 | SD Tristate to $\uparrow \overline{\text{MEMR}}$ | | 0 | 20 | ns |

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE—FLASH WRITE CYCLE

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------|--|-----------------|-----|-----|------|
| tMFW1 | BPAM, SBHE, SA0–19 Setup to $\downarrow \overline{\text{MEMW}}$ | | 10 | | ns |
| tMFW2 | BPAM, SBHE, SA0–19 Hold After $\uparrow \overline{\text{MEMW}}$ | | 5 | | ns |
| tMFW3 | $\downarrow \text{IOCHRDY}$ to $\downarrow \overline{\text{MEMW}}$ | | 0 | 35 | ns |
| tMFW4 | $\overline{\text{MEMW}}$ Inactive | | 50 | | ns |
| tMFW5 | $\uparrow \overline{\text{SRWE}}$ to $\uparrow \text{IOCHRDY}$ | | 20 | 90 | ns |
| tMFW6 | $\uparrow \overline{\text{MEMW}}$ Hold From $\uparrow \text{IOCHRDY}$ | | 0 | | ns |
| tMFW7 | SD Valid From $\downarrow \overline{\text{MEMW}}$ | | | 175 | ns |
| tMFW8 | SD Hold From $\uparrow \overline{\text{MEMW}}$ | | 0 | | ns |
| tMFW9 | $\overline{\text{BPCS/PRDB}}$ Valid From $\downarrow \overline{\text{MEMW}}$ | | | 175 | ns |
| tMFW10 | $\overline{\text{BPCS/PRDB}}$ Setup to $\downarrow \overline{\text{SRWE}}$ | | 15 | | ns |
| tMFW11 | $\overline{\text{SRWE}}$ Active | | 140 | 155 | ns |
| tMFW12 | $\overline{\text{BPCS/PRDB}}$ Hold From $\uparrow \overline{\text{SRWE}}$ | | 15 | | ns |

SWITCHING CHARACTERISTICS: EADI

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------|--|-----------------|-----|--------|------|
| tEAD1 | SRD Setup to \uparrow SRDCLK | | 40 | | ns |
| tEAD2 | SRD Hold to \uparrow SRDCLK | | 40 | | ns |
| tEAD3 | $\overline{\text{SF}}/\overline{\text{BD}}$ Change to \downarrow SRDCLK | | -15 | +15 | ns |
| tEAD4 | $\overline{\text{EAR}}$ Deassertion to \uparrow SRDCLK (First Rising Edge) | | 50 | | ns |
| tEAD5 | $\overline{\text{EAR}}$ Assertion From SFD Event (Packet Rejection) | | 0 | 51,090 | ns |
| tEAD6 | $\overline{\text{EAR}}$ Assertion | | 110 | | ns |

Note: External Address Detection interface is invoked by setting bit 3 in ISACSR2 and resetting bit 0 in ISACSR2. External MAU select is not available when EADISEL bit is set.

SWITCHING CHARACTERISTICS: JTAG (IEEE 1149.1) INTERFACE

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------|------------------------------------|-----------------|-----|-----|------|
| tJTG1 | TCK HIGH Assertion | | 20 | | ns |
| tJTG2 | TCK Period | | 50 | | ns |
| tJTG3 | TDI Setup to \uparrow TCK | | 5 | | ns |
| tJTG4 | TDI, TMS Hold From \uparrow TCK | | 5 | | ns |
| tJTG5 | TMS Setup to \uparrow TCK | | 8 | | ns |
| tJTG6 | TDO Active From \downarrow TCK | | 0 | 30 | ns |
| tJTG7 | TDO Change From \downarrow TCK | | 0 | 30 | ns |
| tJTG8 | TDO Tristate From \downarrow TCK | | 0 | 25 | ns |

Note: JTAG logic is reset with an internal Power-On Reset circuit independent of Sleep Modes.

SWITCHING CHARACTERISTICS: GPSI

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------------|---|-----------------|-------|--------|---------|
| Transmit Timing | | | | | |
| tGPT1 | STDCLK Period (802.3 Compliant) | | 99.99 | 100.01 | ns |
| tGPT2 | STDCLK HIGH Time | | 40 | 60 | ns |
| tGPT3 | TXDAT and TXEN Delay from \uparrow TCLK | | 0 | 70 | ns |
| tGPT4 | RXCRC Setup to \uparrow STDCLK(Last Bit) | | 210 | | ns |
| tGPT5 | RXCRC Hold From \downarrow TENA | | 0 | | ns |
| tGPT6 | CLSN Active Time to Trigger Collision | (Note 1) | 110 | | ns |
| tGPT7 | CLSN Active to \downarrow RXCRC to Prevent LCAR Assertion | | 0 | | ns |
| tGPT8 | CLSN Active to \downarrow RXCRC for SQE Hearbeat Window | | 0 | 4.0 | μ s |
| tGPT9 | CLSN Active to \uparrow RXCRC for Normal Collision | | 0 | 51.2 | μ s |
| Receive Timing | | | | | |
| tGPR1 | SRDCLK Period | (Note 2) | 80 | 120 | ns |
| tGPR2 | SRDCLK High Time | (Note 2) | 30 | 80 | ns |
| tGPR3 | SRDCLK Low Time | (Note 2) | 30 | 80 | ns |
| tGPR4 | RXDAT and RXCRC Setup to \uparrow SRDCLK | | 15 | | ns |
| tGPR5 | RXDAT Hold From \uparrow RCLK | | 15 | | ns |
| tGPR6 | RXCRC Hold From \downarrow SRDCLK | | 0 | | ns |
| tGPR7 | CLSN Active to First \uparrow SRDCLK (Collision Recognition) | | 0 | | ns |
| tGPR8 | CLSN Active to \uparrow SRDCLK for Address Type Designation Bit | (Note 3) | 51.2 | | μ s |
| tGPR9 | CLSN Setup to last \uparrow SRDCLK for Collision Recognition | | 210 | | ns |
| tGPR10 | CLSN Active | | 110 | | ns |
| tGPR11 | CLSN Inactive Setup to First \uparrow RCLK | | 300 | | ns |
| tGPR12 | CLSN Inactive Hold to Last \uparrow RCLK | | 300 | | ns |

Notes:

1. CLSN must be asserted for a continuous period of 110 ns or more. Assertion for less than 110 ns period may or may not result in CLSN recognition.
2. RCLK should meet jitter requirements of IEEE 802.3 specification.
3. CLSN assertion before 51.2 μ s will be indicated as a normal collision. CLSN assertion after 51.2 μ s will be considered as a Late Receive Collision.

SWITCHING CHARACTERISTICS: AUI

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|-------------------------------------|--|---|--------|--------|------|
| AUI Port | | | | | |
| t _{DOTR} | DO+,DO- Rise Time (10% to 90%) | | 2.5 | 5.0 | ns |
| t _{DOTF} | DO+,DO- Fall Time (90% to 10%) | | 2.5 | 5.0 | ns |
| t _{DORM} | DO+,DO- Rise and fall Time Mismatch | | – | 1.0 | ns |
| t _{DOETD} | DO+/- End of Transmission | | 200 | 375 | ns |
| t _{PWODI} | DI Pulse Width Accept/Reject Threshold | V _{IN} > V _{ASQ} (Note 1) | 15 | 45 | ns |
| t _{PWKDI} | DI Pulse Width Maintain/Turn-Off Threshold | V _{IN} > V _{ASQ} (Note 2) | 136 | 200 | ns |
| t _{PWOCI} | CI Pulse Width Accept/Reject Threshold | V _{IN} > V _{ASQ} (Note 3) | 10 | 26 | ns |
| t _{PWKCI} | CI Pulse Width Maintain/Turn-Off Threshold | V _{IN} > V _{ASQ} (Note 4) | 90 | 160 | ns |
| Internal MENDEC Clock Timing | | | | | |
| t _{X1} | XTAL1 Period | V _{IN} = External Clock | 49.995 | 50.005 | ns |
| t _{X1H} | XTAL1 HIGH Pulse Width | V _{IN} = External Clock | 20 | | ns |
| t _{X1L} | XTAL1 LOW Pulse width | V _{IN} = External Clock | 20 | | ns |
| t _{X1R} | XTAL1 Rise Time | V _{IN} = External Clock | | 5 | ns |
| t _{X1F} | XTAL1 Fall Time | V _{IN} = External Clock | | 5 | ns |

Notes:

1. DI pulses narrower than t_{PWODI} (min) will be rejected; pulses wider than t_{PWODI} (max) will turn internal DI carrier sense on.
2. DI pulses narrower than t_{PWKDI} (min) will maintain internal DI carrier sense on; pulses wider than t_{PWKDI} (max) will turn internal DI carrier sense off.
3. CI pulses narrower than t_{PWOCI} (min) will be rejected; pulses wider than t_{PWOCI} (max) will turn internal CI carrier sense on.
4. CI pulses narrower than t_{PWKCI} (min) will maintain internal CI carrier sense on; pulses wider than t_{PWKCI} (max) will turn internal CI carrier sense off.

SWITCHING CHARACTERISTICS: 10BASE-T INTERFACE

| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------------|--|------------------|-----|-----|------|
| Transmit Timing | | | | | |
| tTETD | Transmit Start of Idle | | 250 | 350 | ns |
| tTR | Transmitter Rise Time | (10% to 90%) | | 5.5 | ns |
| tTF | Transmitter Fall Time | (90% to 10%) | | 5.5 | ns |
| tTM | Transmitter Rise and Fall Time Mismatch | | | 2 | ns |
| tPERLP | Idle Signal Period | | 8 | 24 | ms |
| tpWLP | Idle Link Pulse Width | (Note 1) | 75 | 120 | ns |
| tpWPLP | Predistortion Idle Link Pulse Width | (Note 1) | 45 | 55 | ns |
| tJA | Transmit Jabber Activation Time | | 20 | 150 | ms |
| tJR | Transmit Jabber Reset Time | | 250 | 750 | ms |
| Receive Timing | | | | | |
| tpWNRD | RXD Pulse Width Not to Turn Off Internal Carrier Sense | VIN > VTHS (min) | 136 | – | ns |
| tpWROFF | RXD Pulse Width to Turn Off | VIN > VTHS (min) | | 200 | ns |

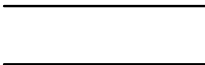
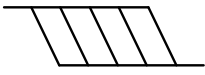
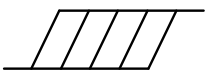
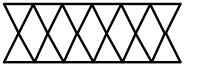
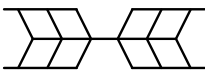
Note:

1. Not tested; parameter guaranteed by characterization.

SWITCHING CHARACTERISTICS: SERIAL EEPROM

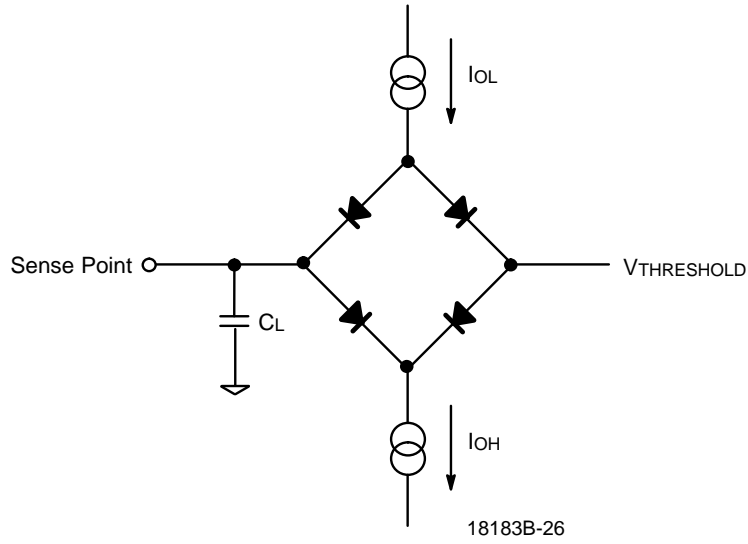
| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------|---|-----------------|------|-----|------|
| tsR1 | EESK High Time | | 790 | | ns |
| tsR2 | EESK Low Time | | 790 | | ns |
| tsR3 | ↑ EECS EEDI From ↓ EESK | | -15 | 15 | ns |
| tsR4 | ↓ EECS, EEDI and SHFBUSY From ↓ EESK | | -15 | 15 | ns |
| tsR5 | EECS Low Time | | 1590 | | ns |
| tsR6 | EEDO Setup to ↑ EESK | | 35 | | ns |
| tsR7 | EEDO Hold From ↑ EESK | | 0 | | ns |
| tsL1 | EEDO Setup to ↓ $\overline{IO\overline{R}}$ | | 95 | | ns |
| tsL2 | EEDO Setup to ↑ IOCHRDY | | 140 | | ns |
| tsL3 | EESK, EEDI, EECS and SHFBUSY Delay From ↑ $\overline{IO\overline{W}}$ | | 160 | 235 | ns |

KEY TO SWITCHING WAVEFORMS

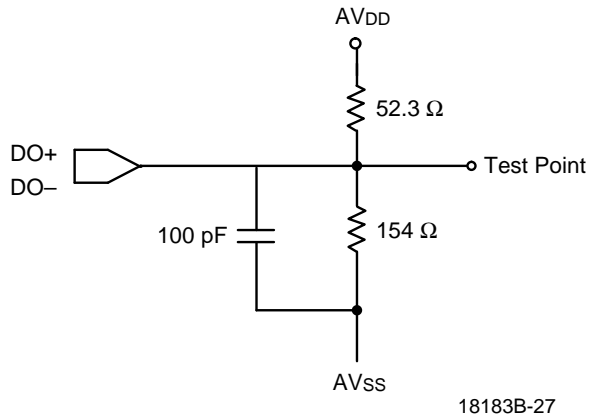
| WAVEFORM | INPUTS | OUTPUTS |
|---|----------------------------------|---|
|  | Must be Steady | Will be Steady |
|  | May Change from H to L | Will be Changing from H to L |
|  | May Change from L to H | Will be Changing from L to H |
|  | Don't Care, Any Change Permitted | Changing, State Unknown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

KS000010

SWITCHING TEST CIRCUITS

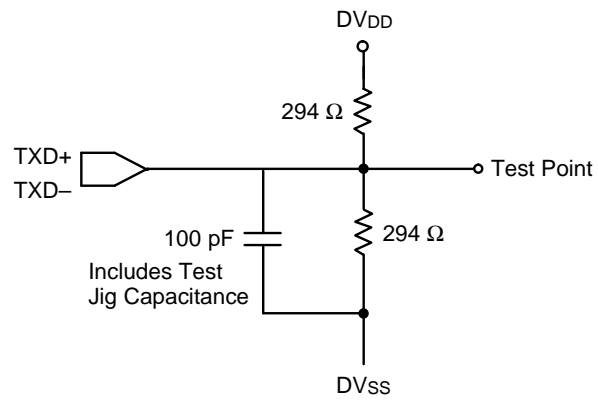


Normal and Three-State Outputs



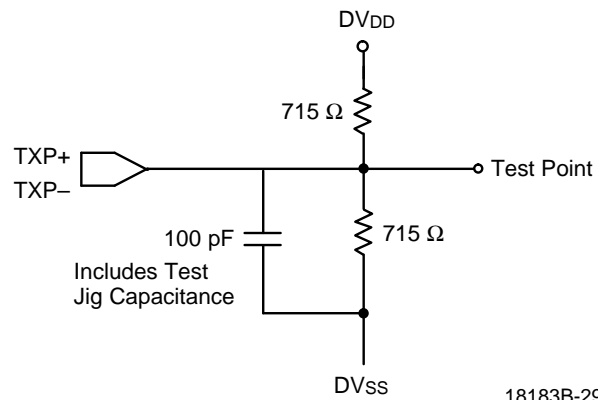
AUI DO Switching Test Circuit

SWITCHING TEST CIRCUITS



18183B-28

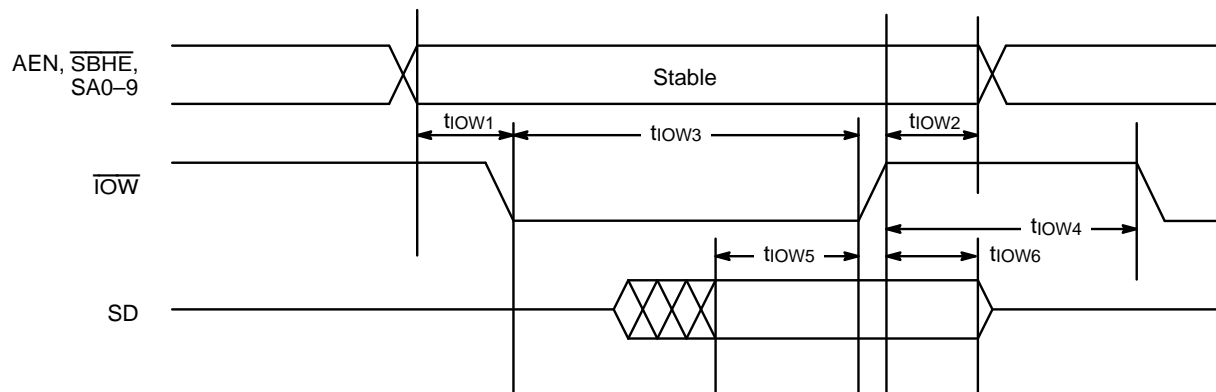
TXD Switching Test Circuit



18183B-29

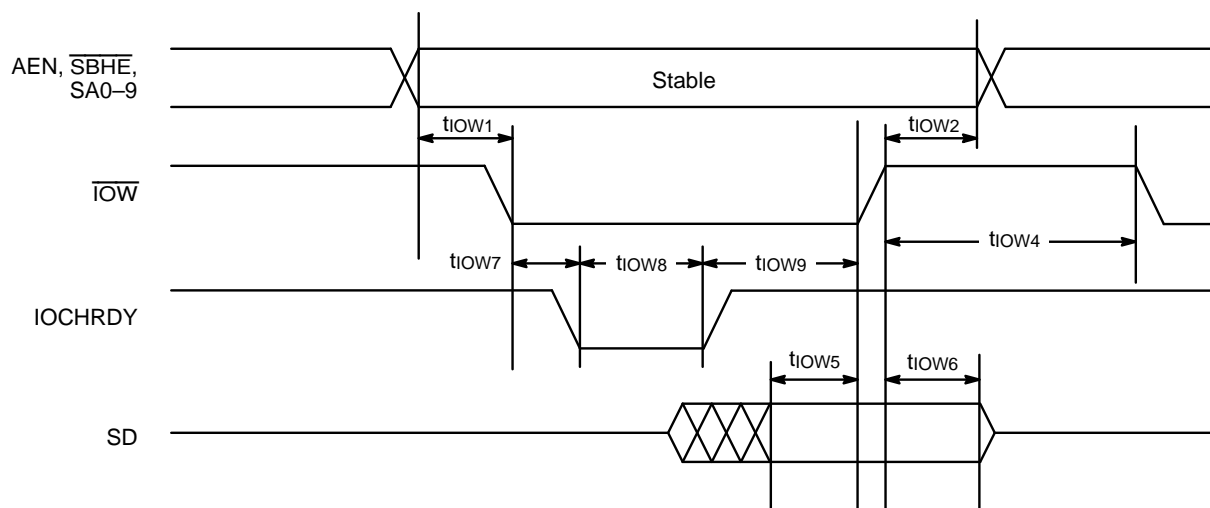
TXP Outputs Test Circuit

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-30

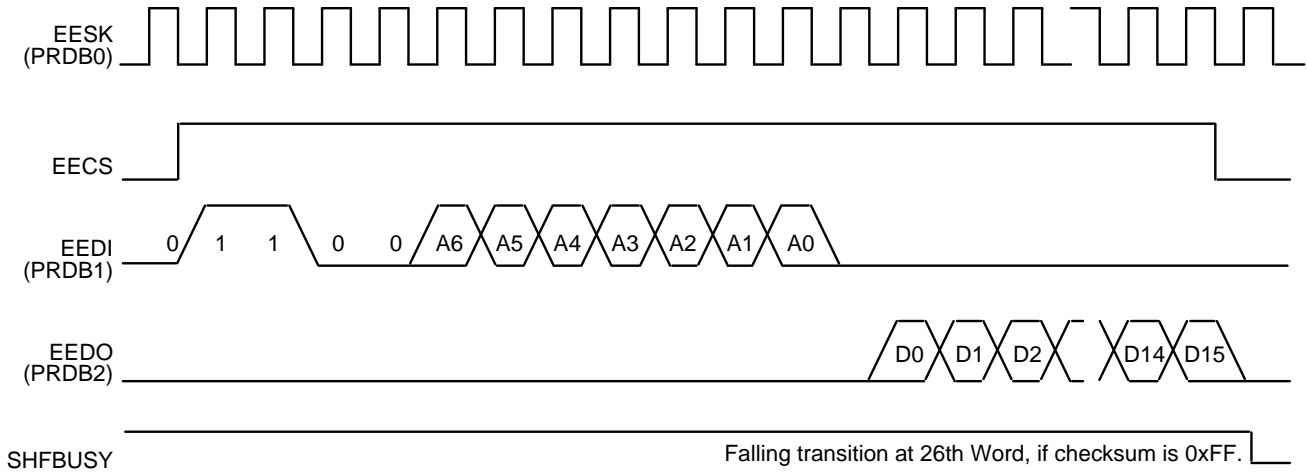
I/O Write without Wait States



18183B-31

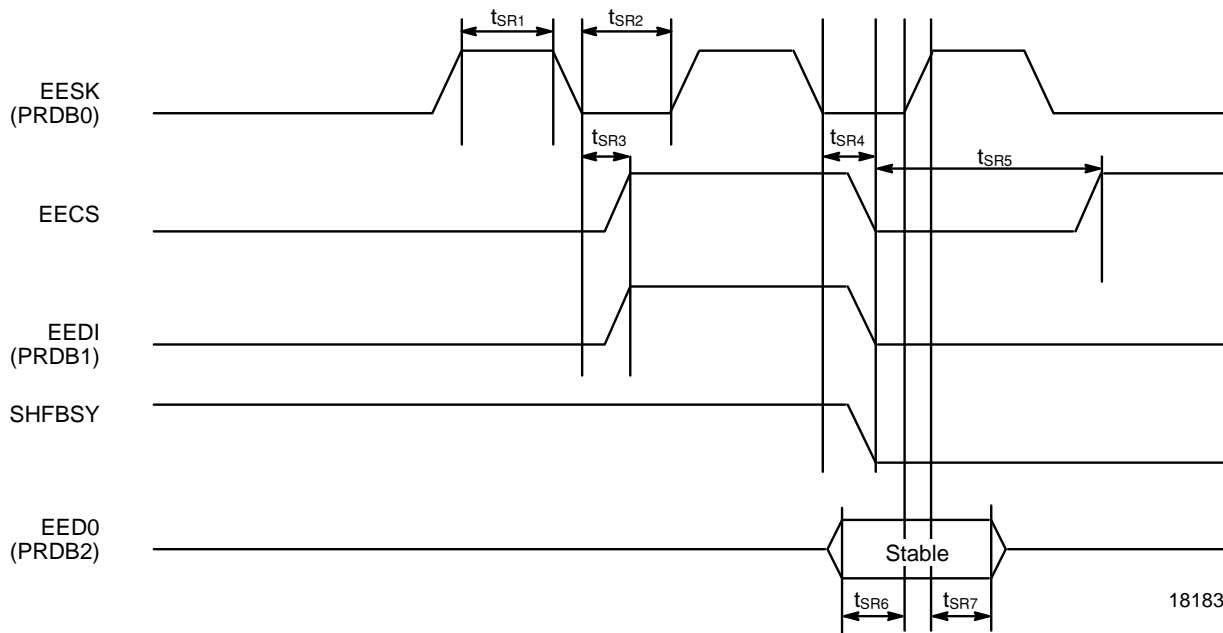
I/O Write with Wait States

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-32

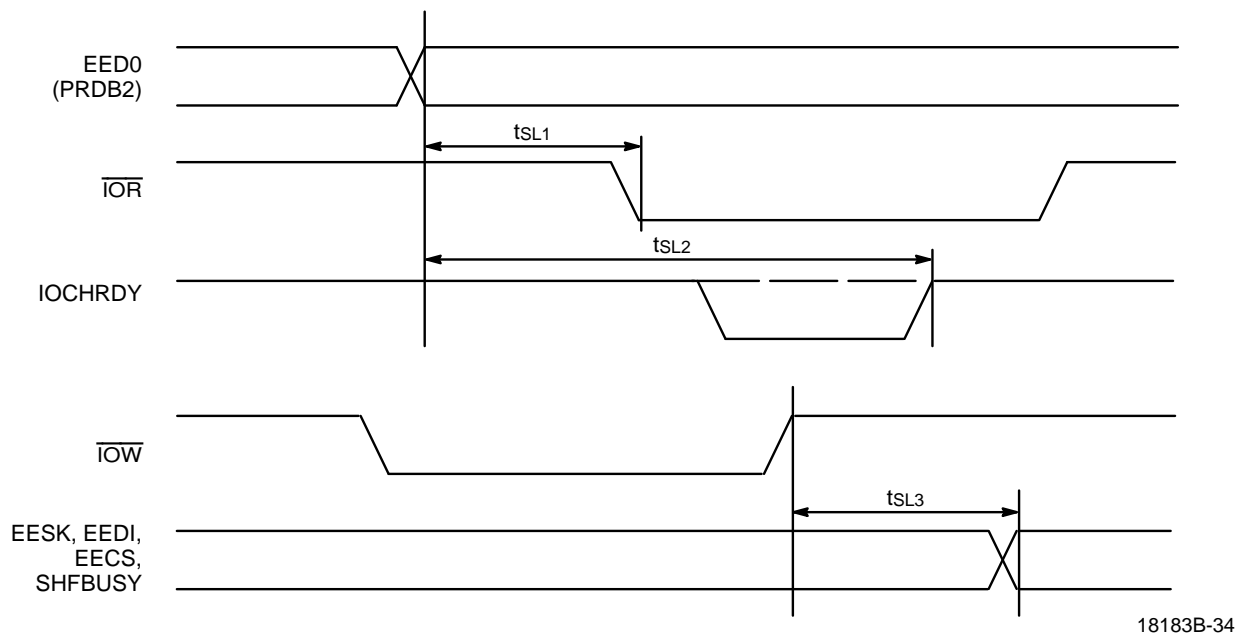
Serial Shift EEPROM Interface Read Timing



18183A-33

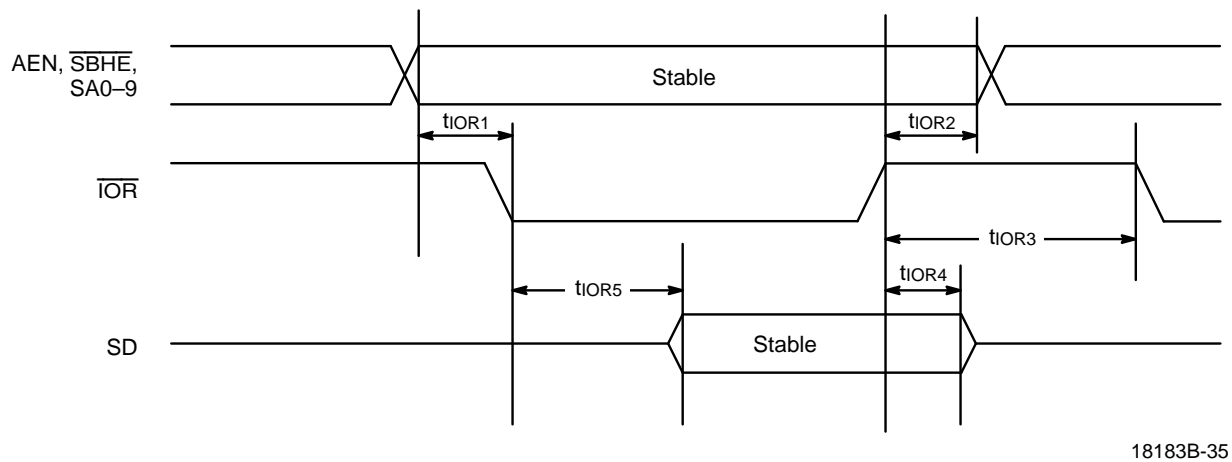
Serial EEPROM Control Timing

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-34

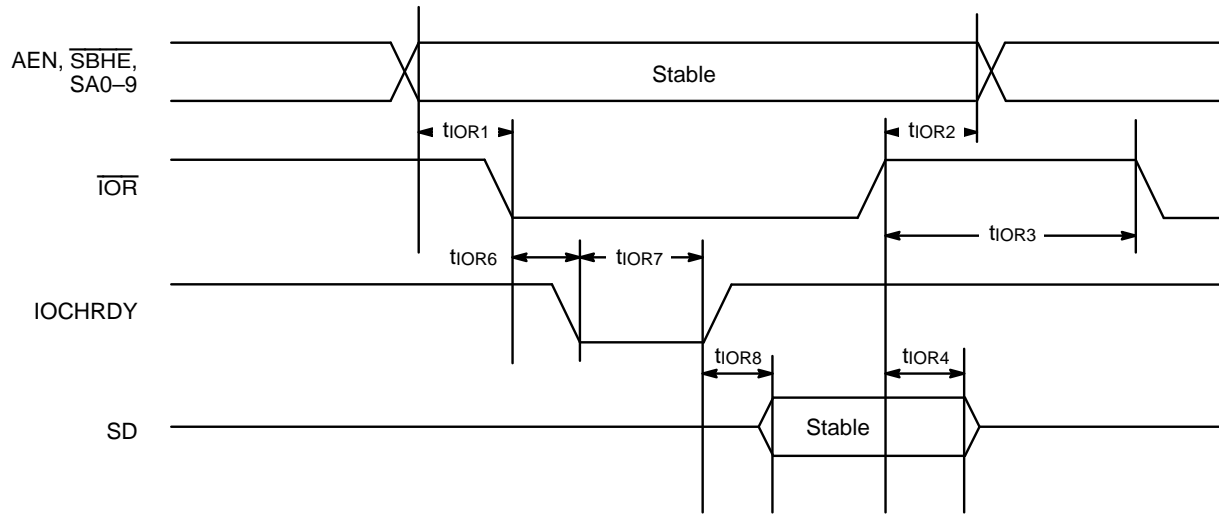
Slave Serial EEPROM Latency Timing



18183B-35

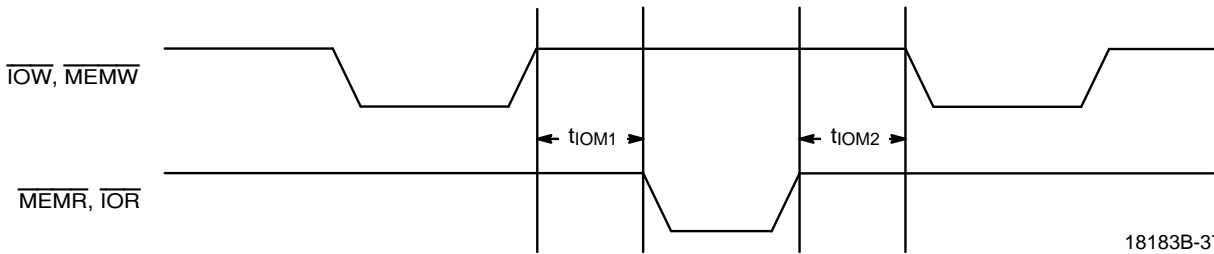
I/O Read without Wait States

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-36

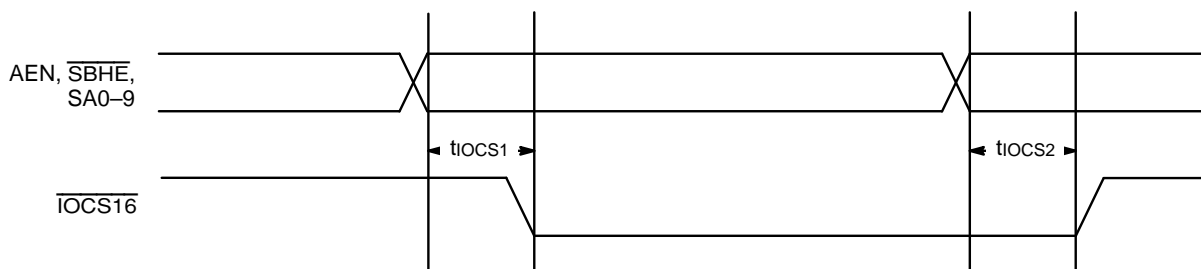
I/O Read with Wait States



18183B-37

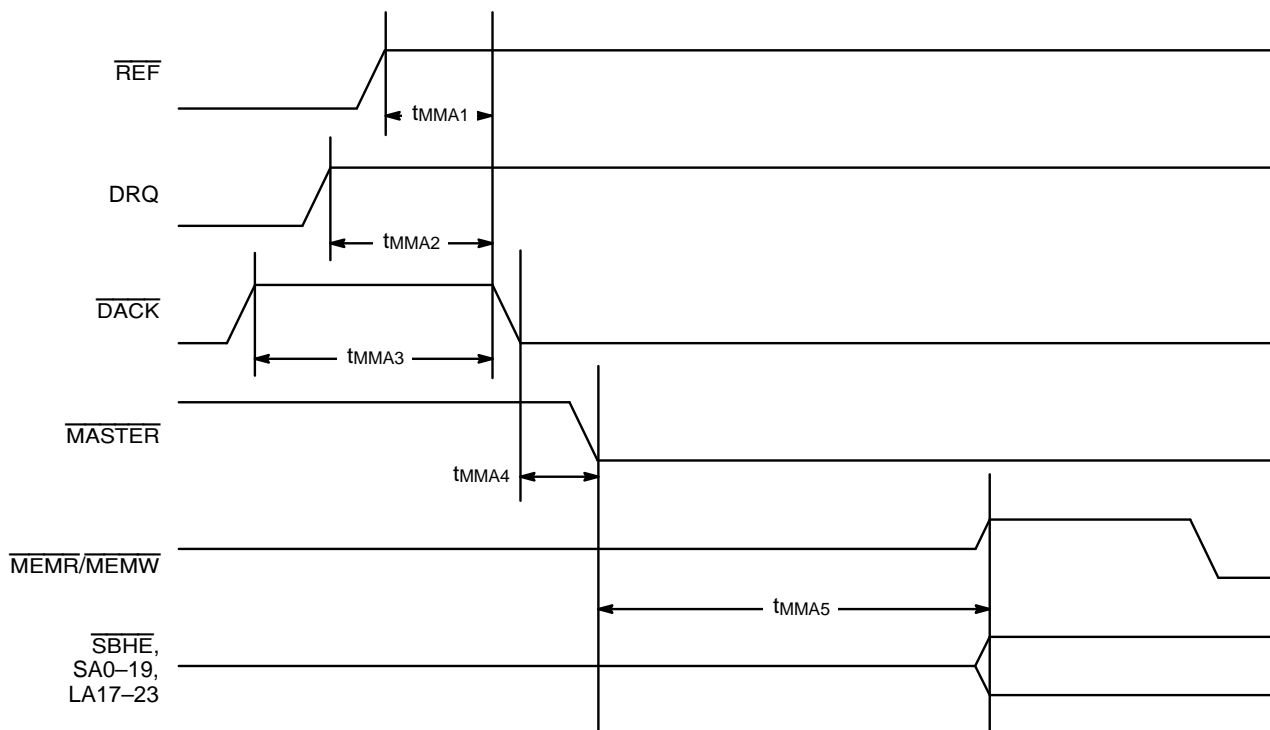
I/O to Memory Command Inactive Time

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-38

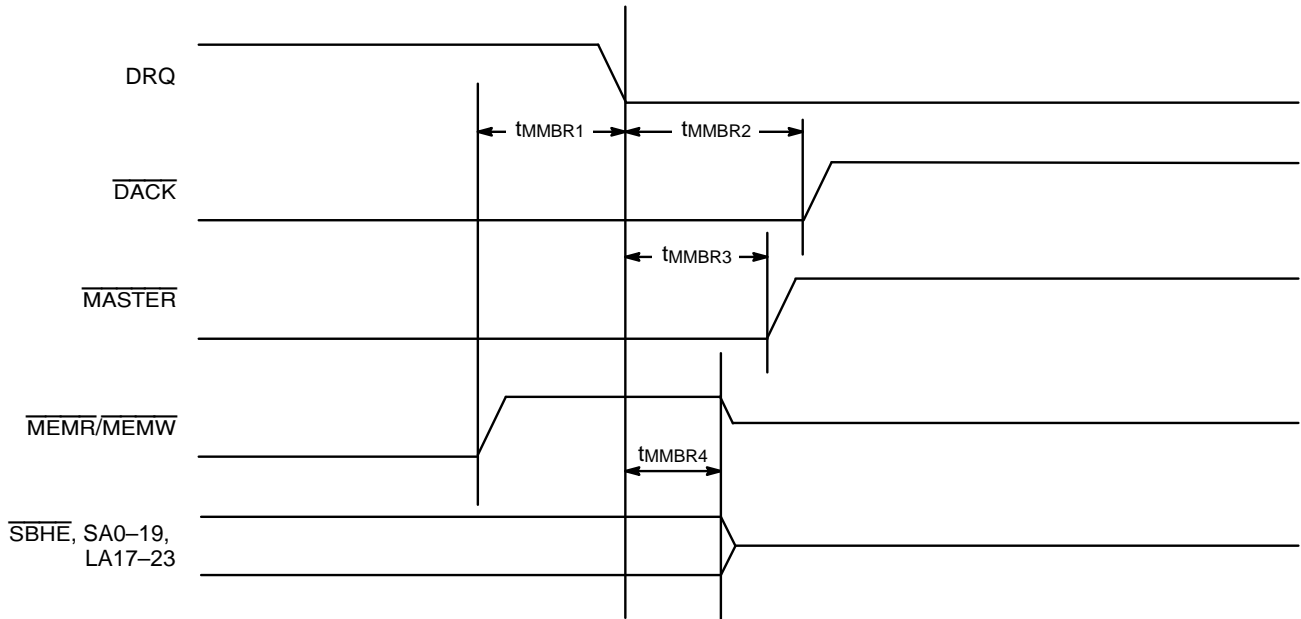
IOCS16 Timings



18183B-39

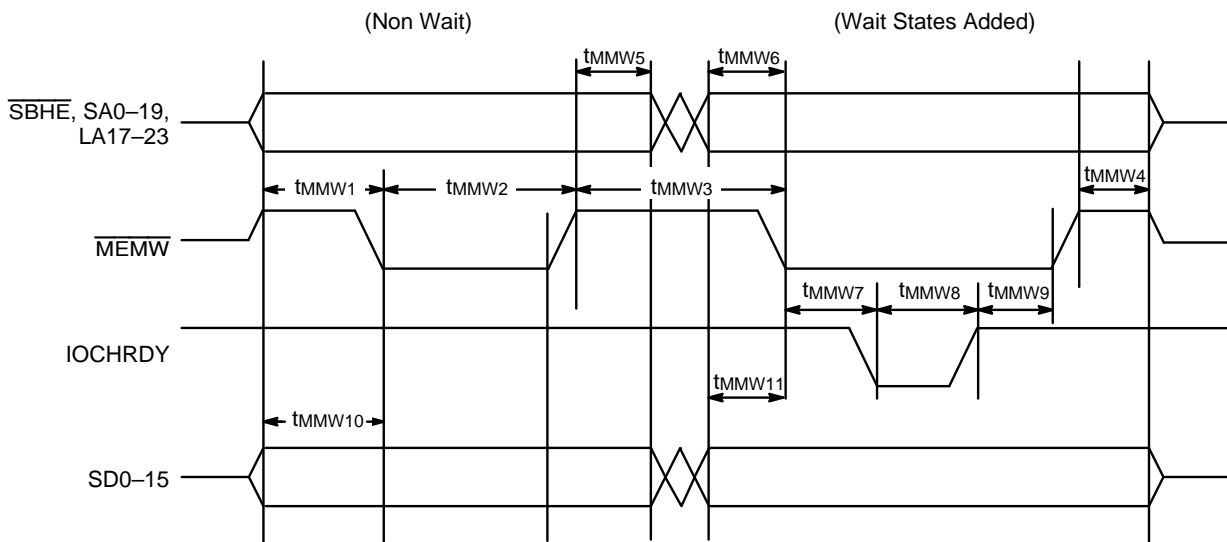
Bus Acquisition

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-40

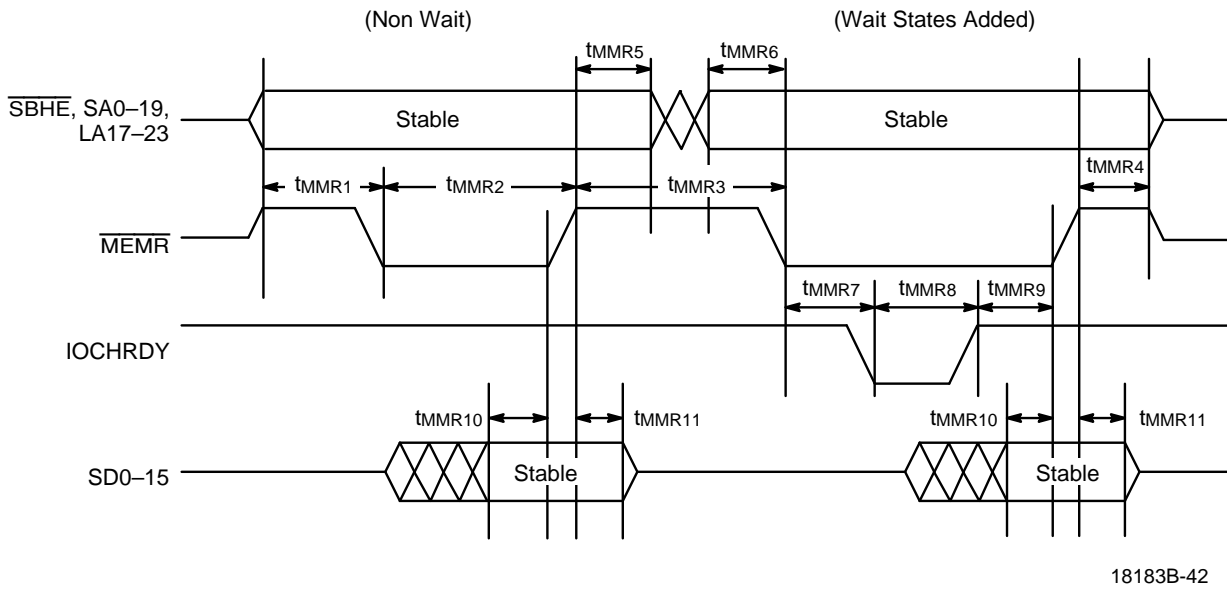
Bus Release



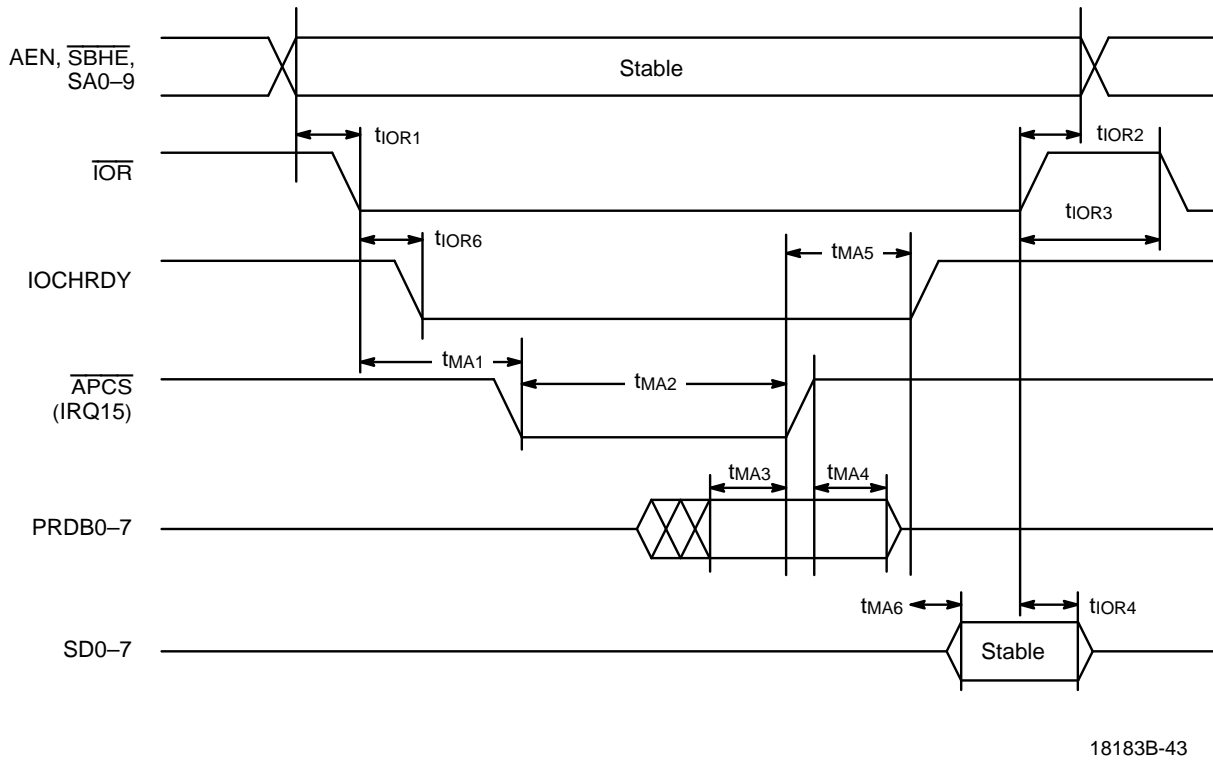
18183B-41

Write Cycles

SWITCHING WAVEFORMS: BUS MASTER MODE

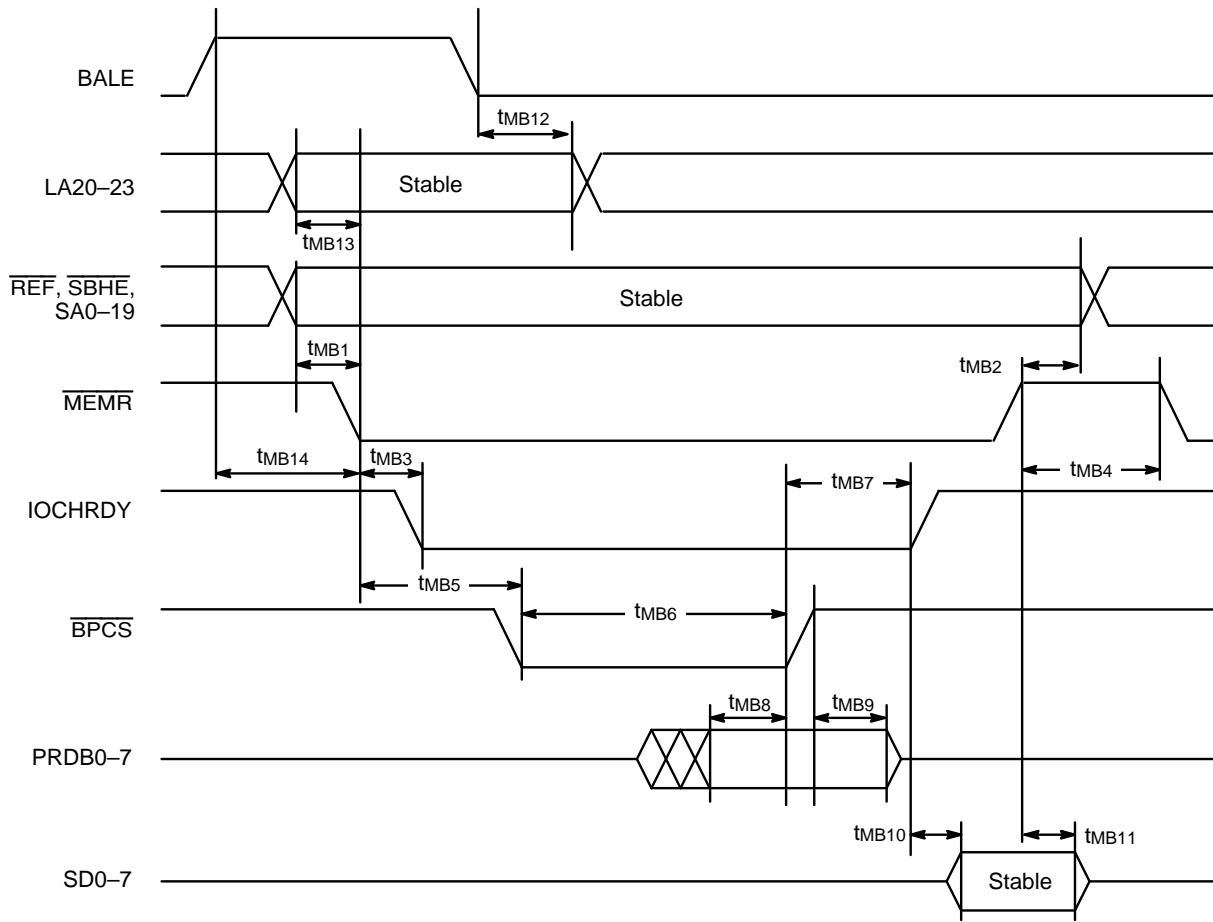


Read Cycles



External Address PROM Read Cycle

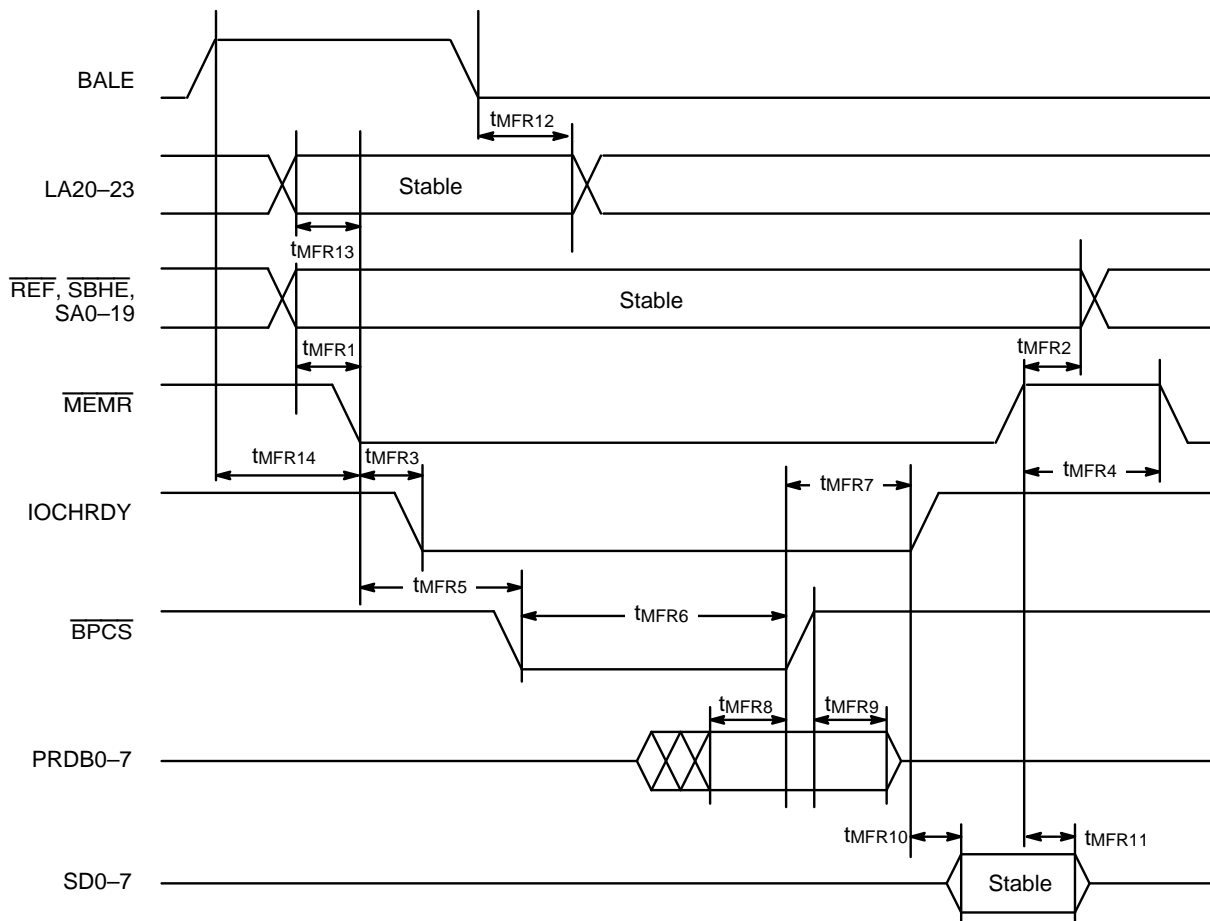
SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-44

Boot PROM Read Cycle

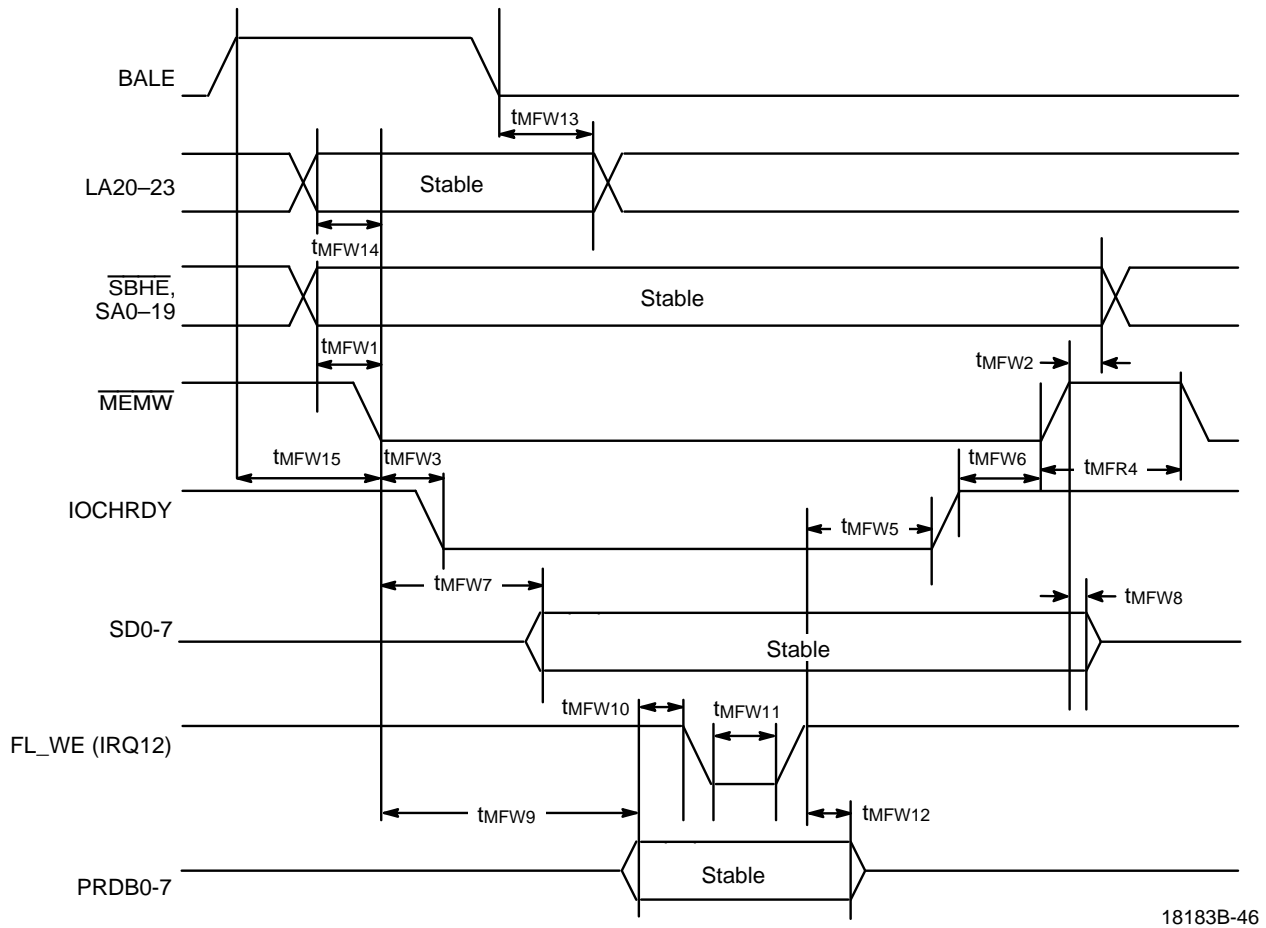
SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-45

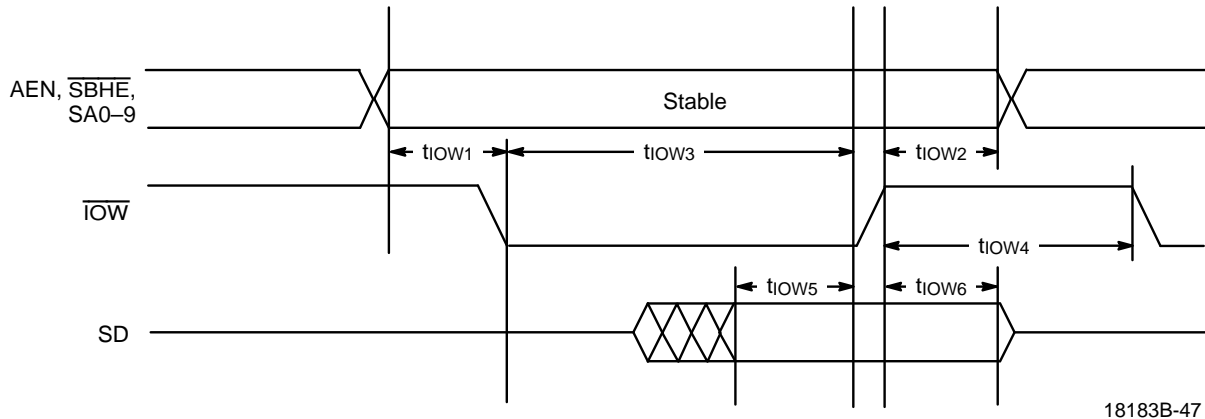
Flash Read Cycle

SWITCHING WAVEFORMS: BUS MASTER MODE



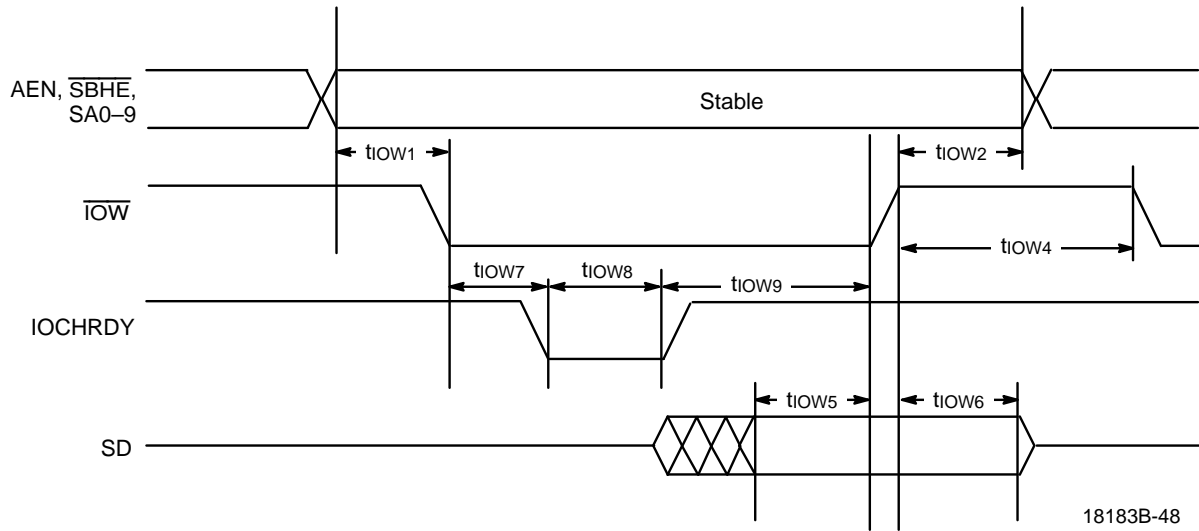
Flash Write Cycle

SWITCHING WAVEFORMS: SHARED MEMORY MODE



I/O Write without Wait States

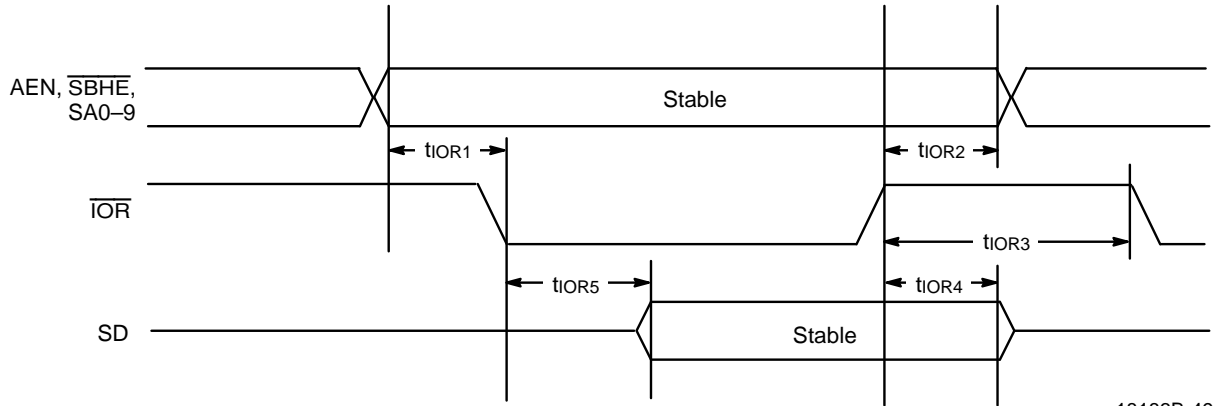
18183B-47



I/O Write with Wait States

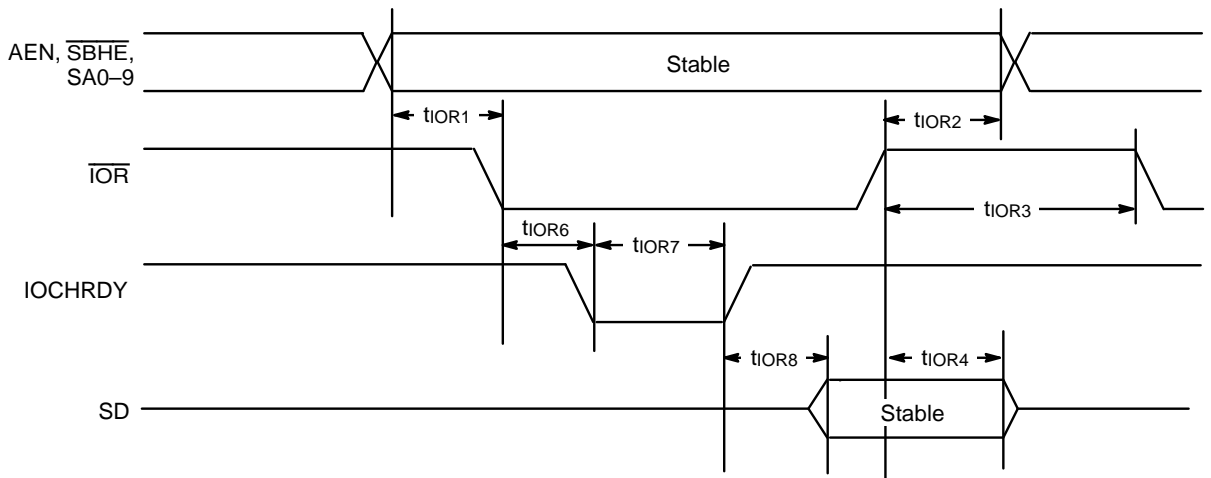
18183B-48

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-49

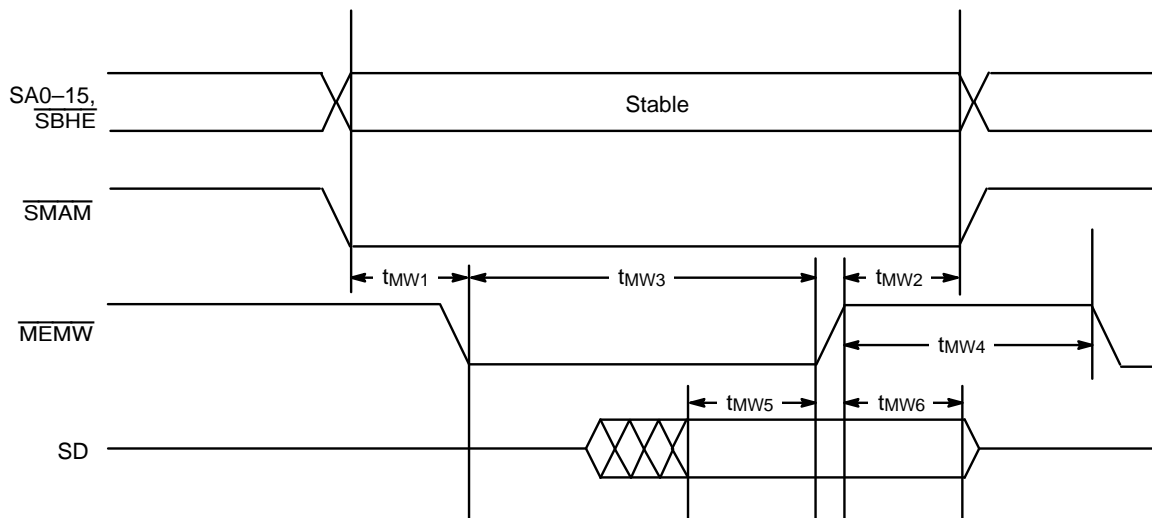
I/O Read without Wait States



18183B-50

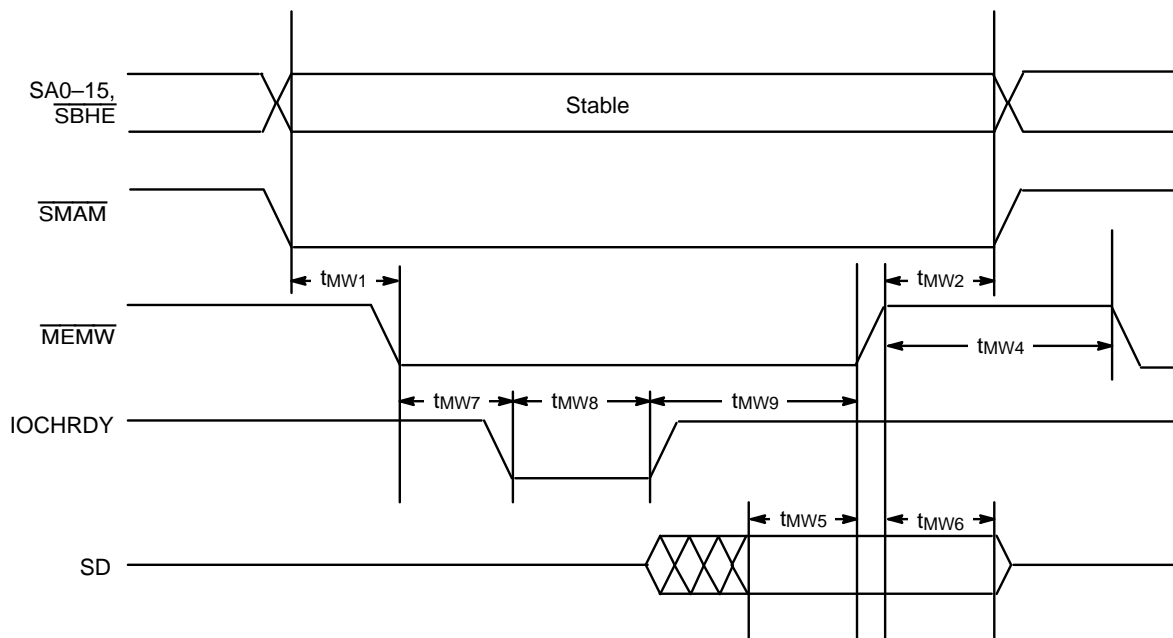
I/O Read with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-51

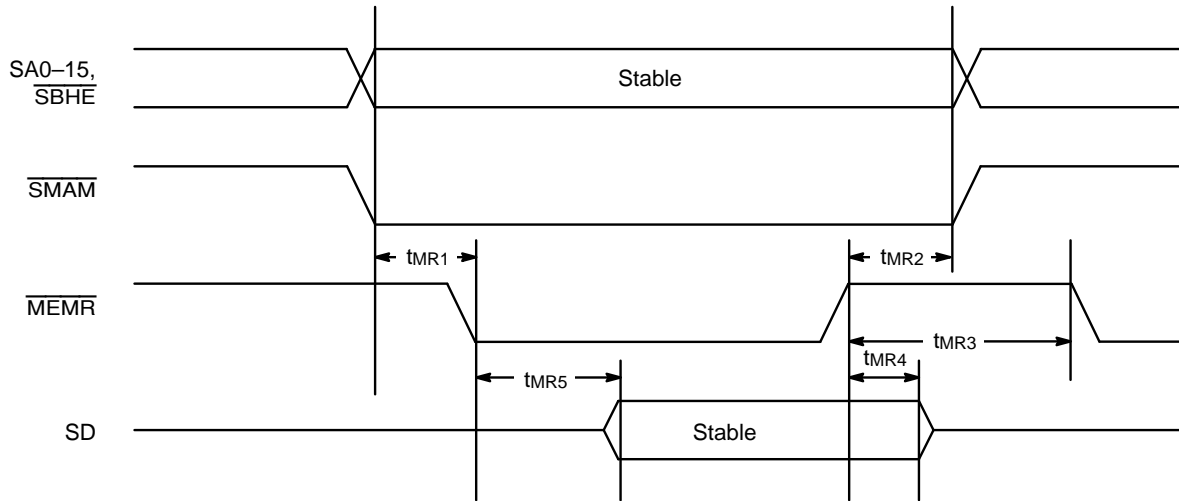
Memory Write without Wait States



18183B-52

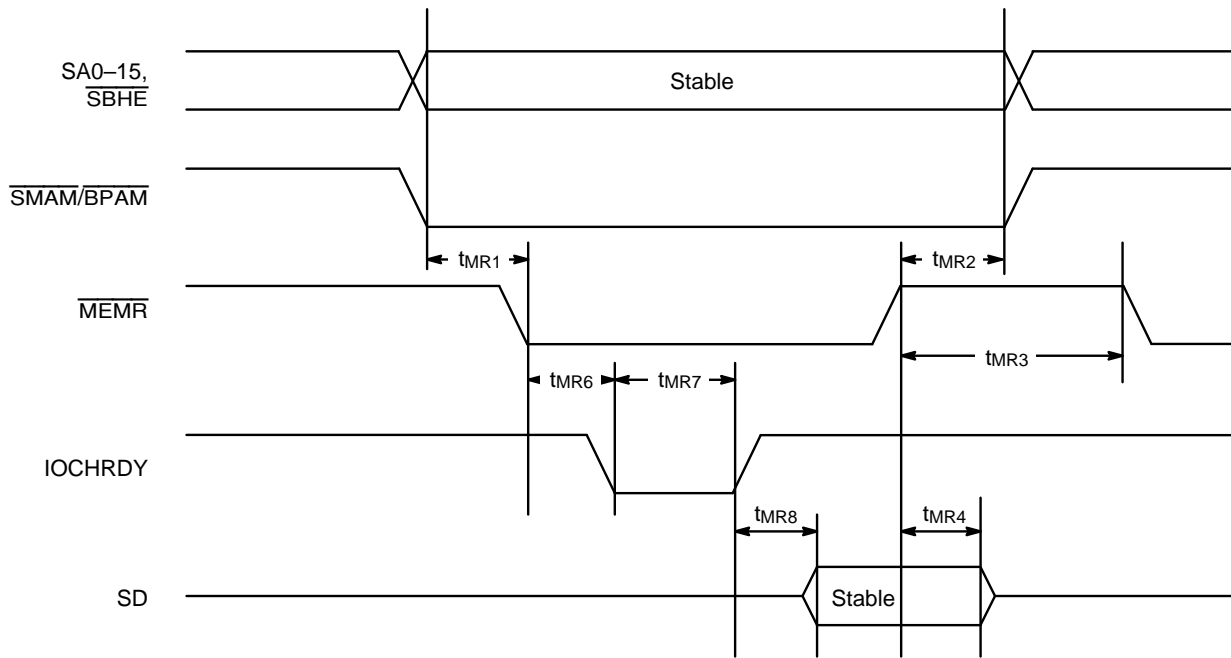
Memory Write with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-53

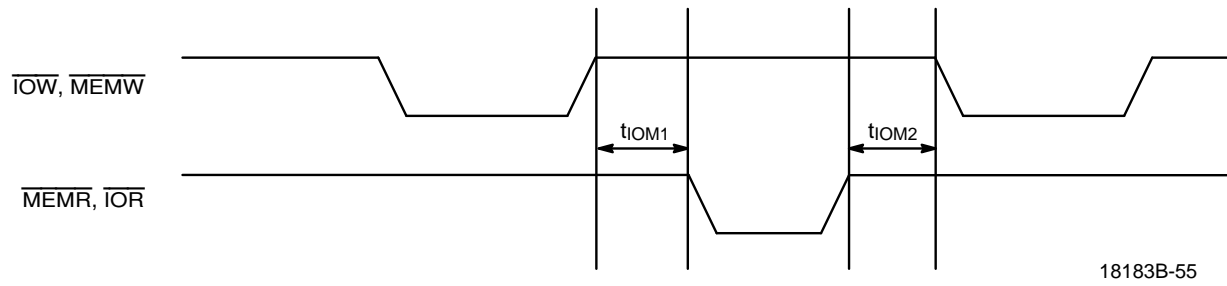
Memory Read without Wait States



18183B-54

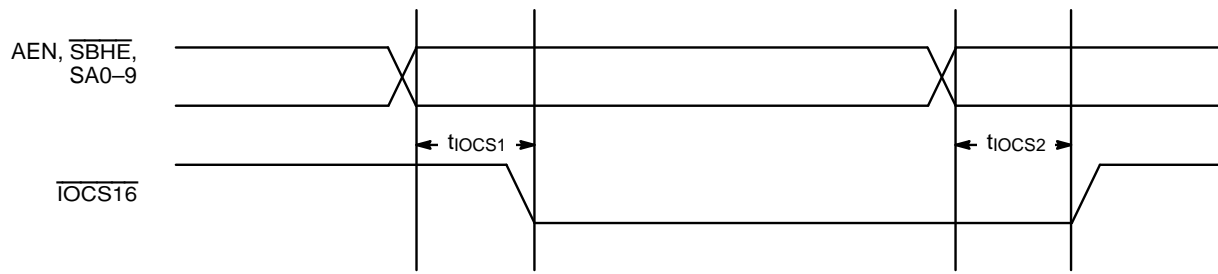
Memory Read with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-55

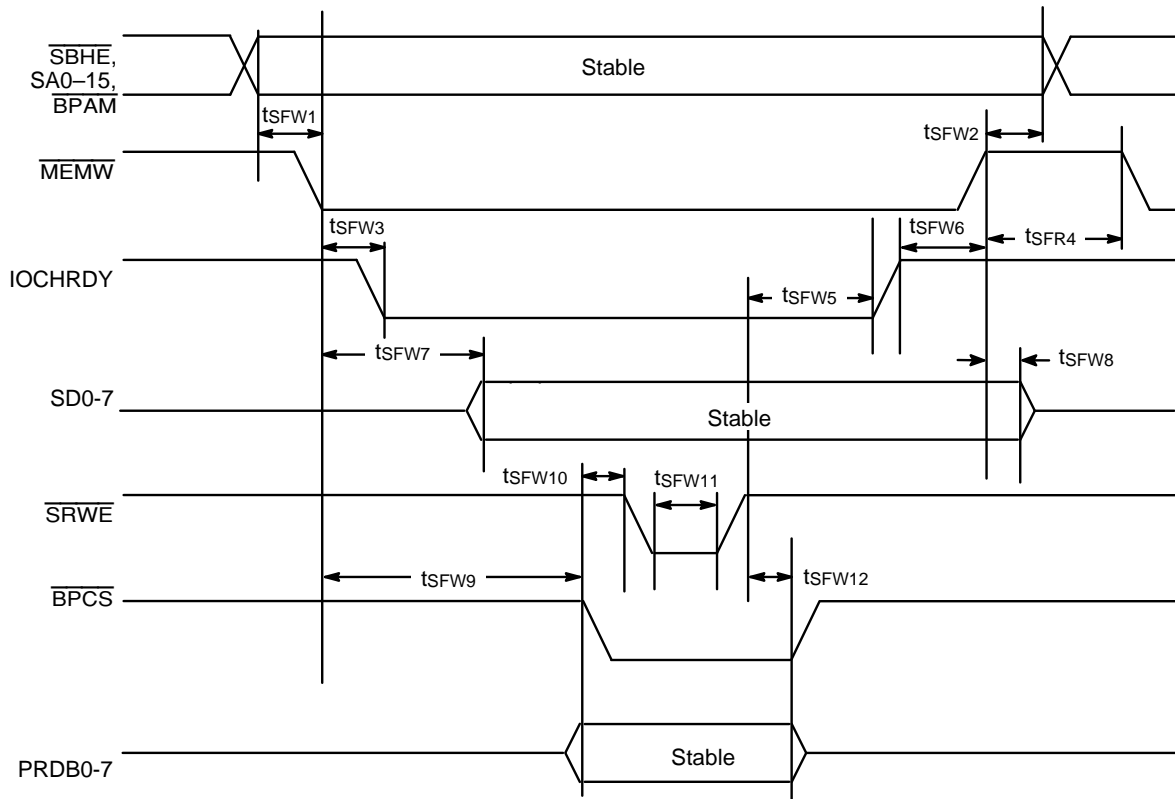
I/O to Memory Command Inactive Time



18183B-56

$\overline{\text{IOCS16}}$ Timings

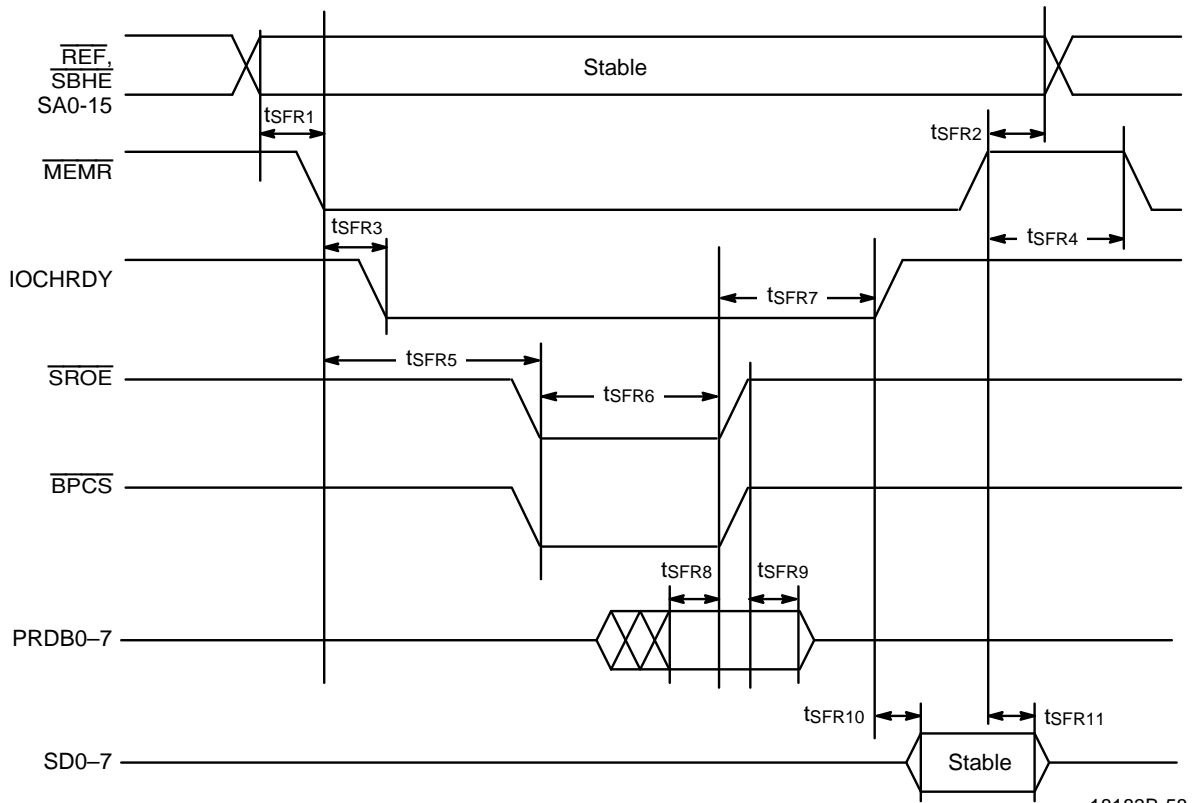
SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-57

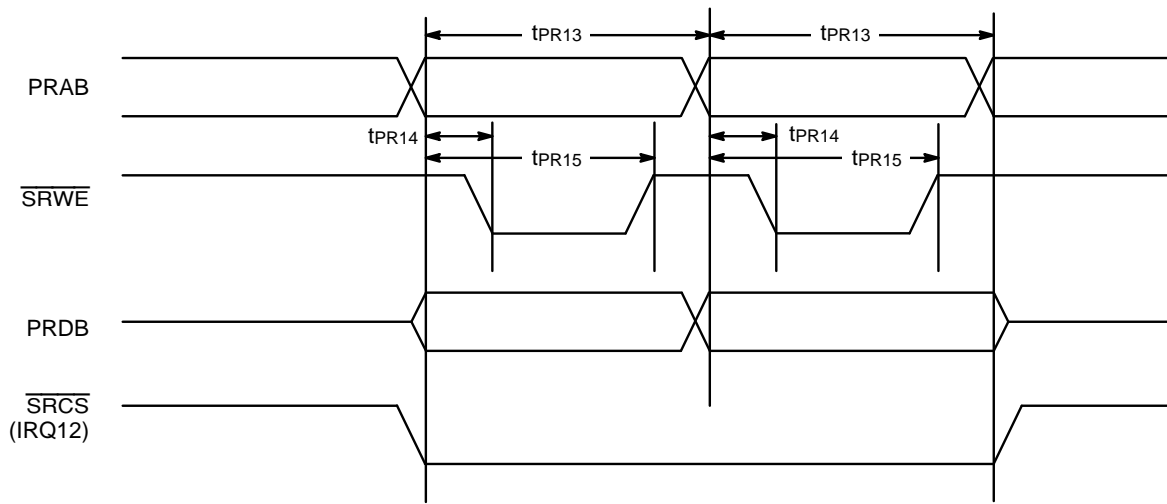
Flash Write Cycle

SWITCHING WAVEFORMS: SHARED MEMORY MODE



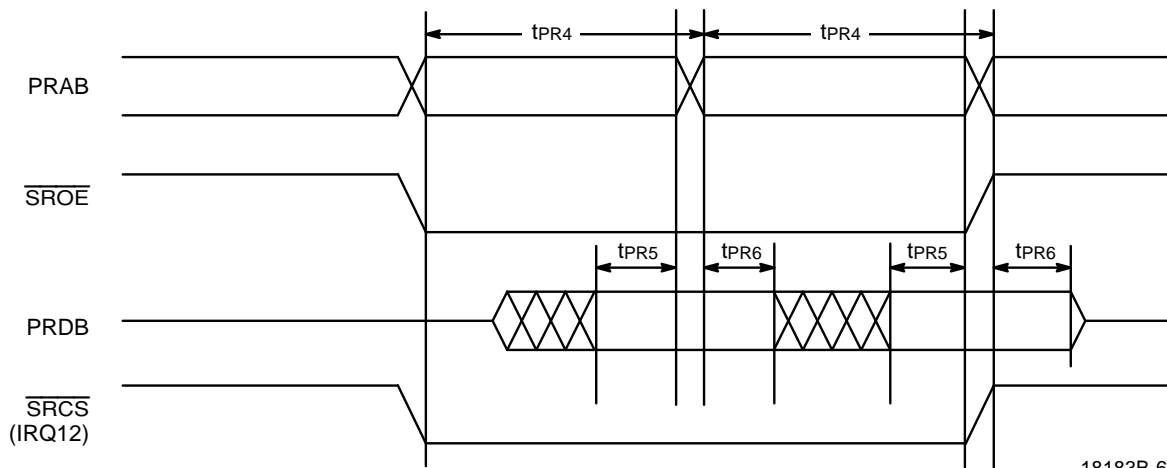
Flash Read Cycle

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-59

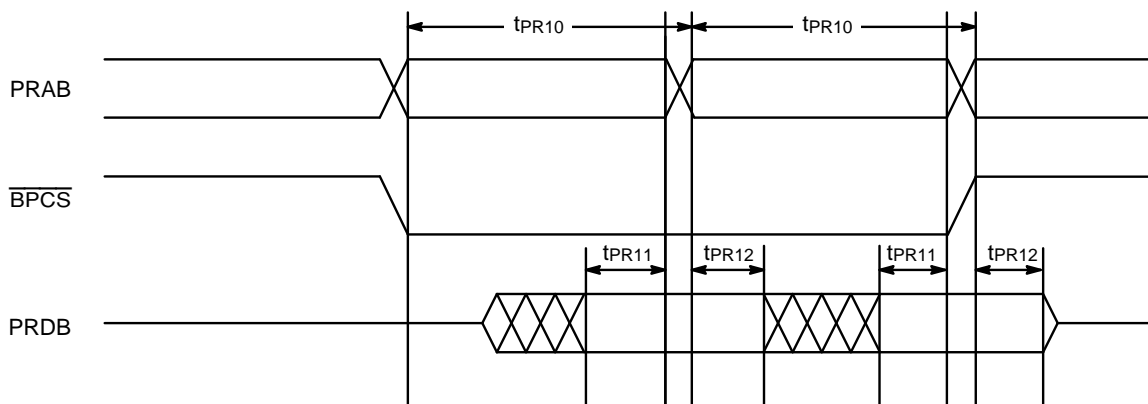
SRAM Write on Private Bus (When FL_Sel is Enabled)



18183B-60

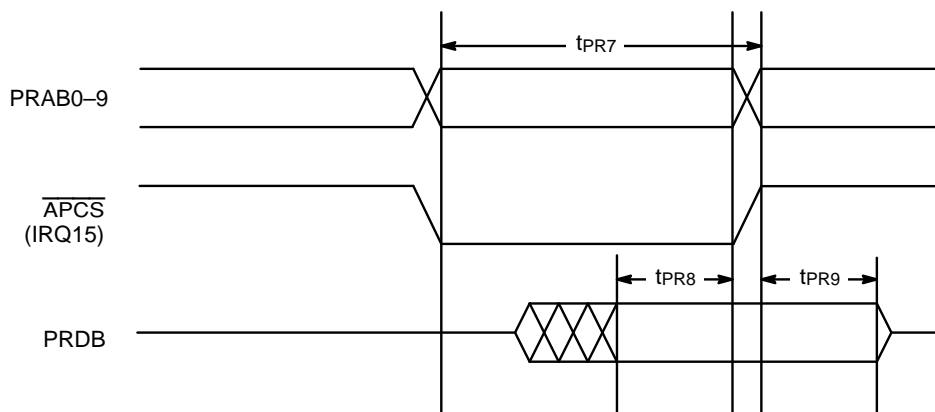
SRAM Read on Private Bus (When FL_Sel is Enabled)

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-61

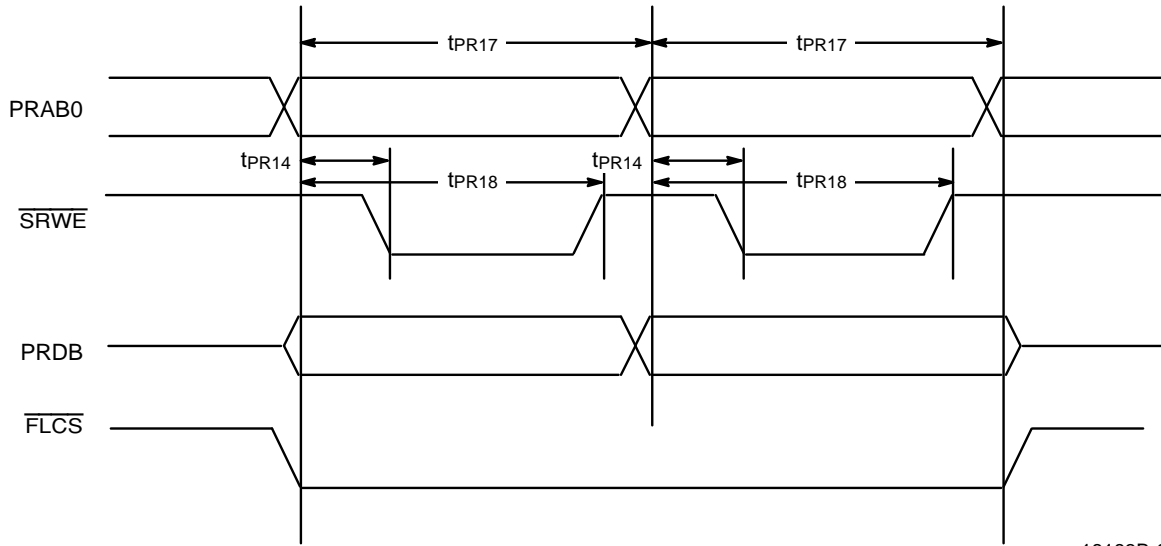
Boot PROM Read on Private Bus



18183B-62

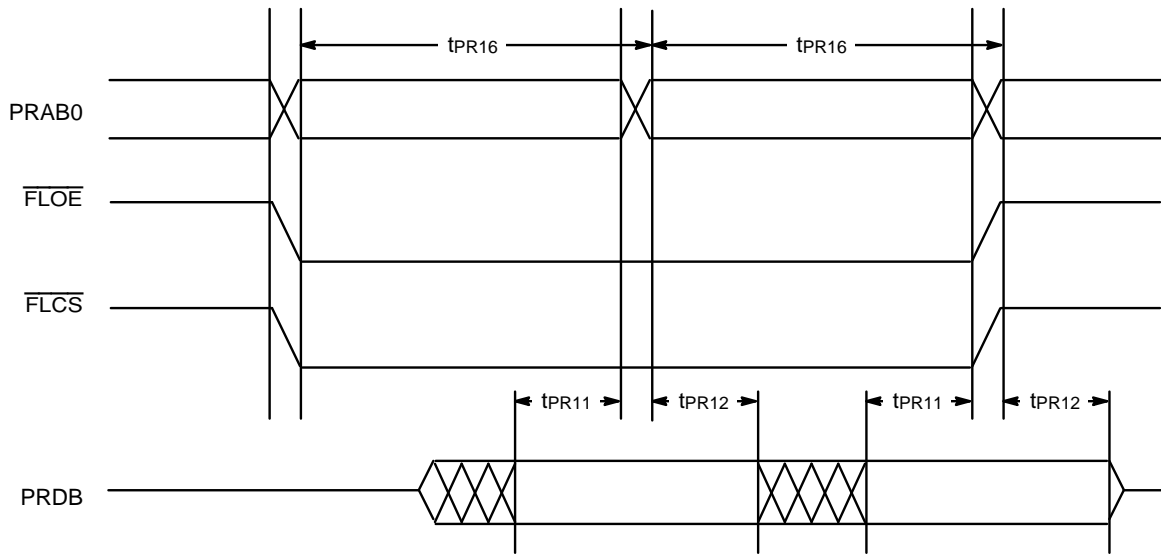
Address PROM Read on Private Bus

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-63

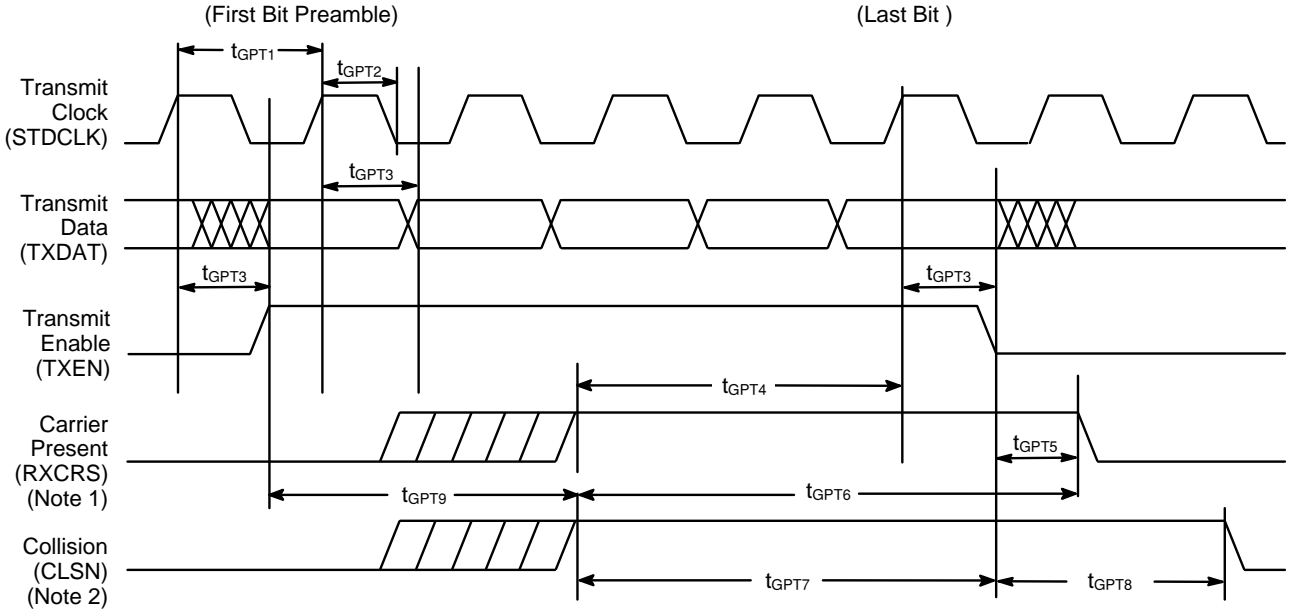
Flash Write on Private Bus



18183B-64

Flash Read on Private Bus

SWITCHING WAVEFORMS: GPSI

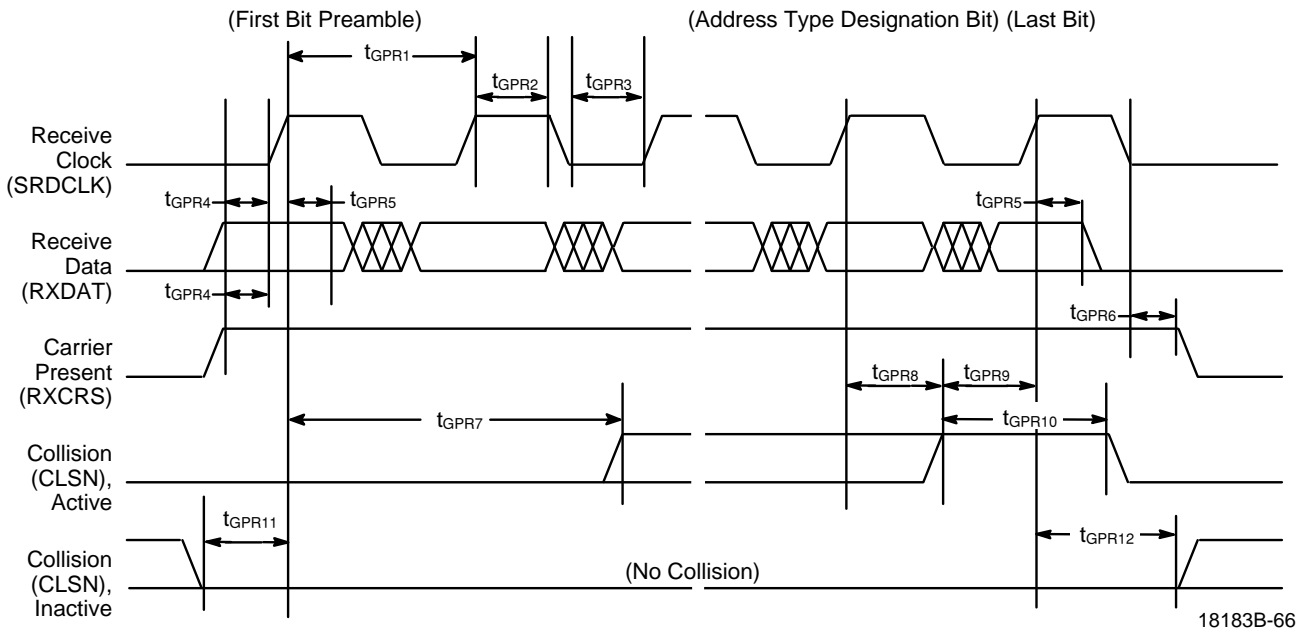


18183B-65

Notes:

1. If RXCRS is not present during transmission, LCAR bit in TMD3 will be set.
2. If CLSN is not present during or shortly after transmission, CERR in CSR0 will be set.

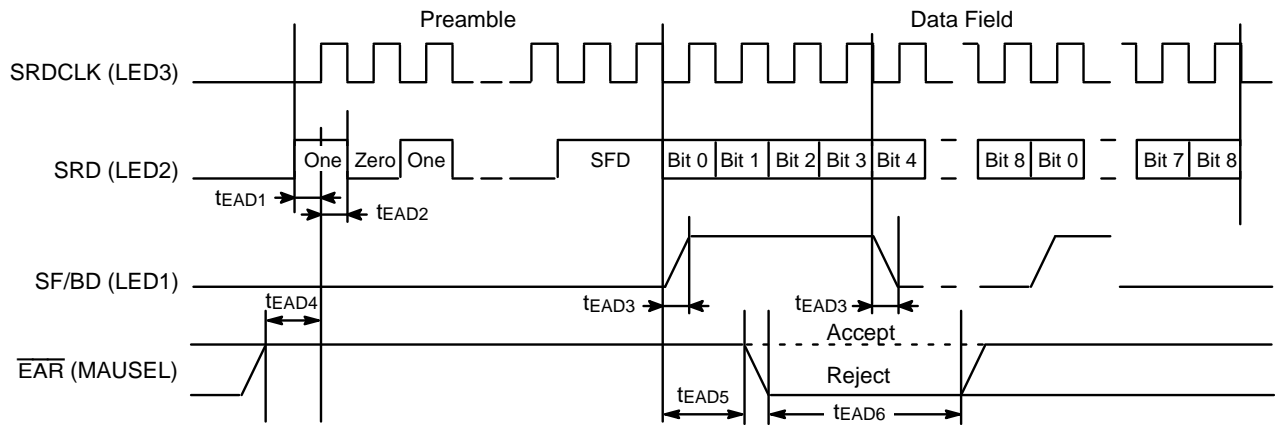
Transmit Timing



18183B-66

Receive Timing

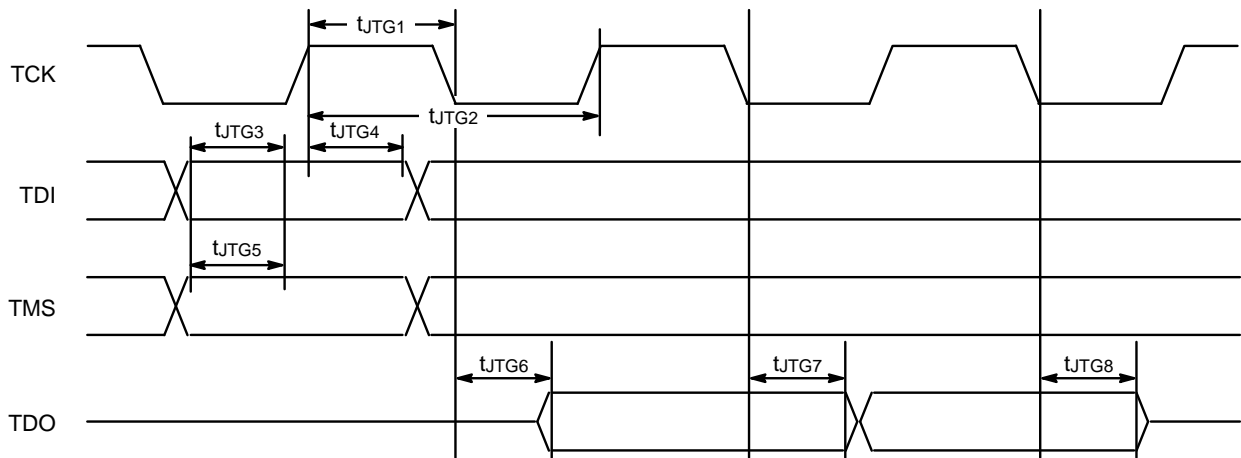
SWITCHING WAVEFORMS: EADI



18183B-67

EADI Reject Timing

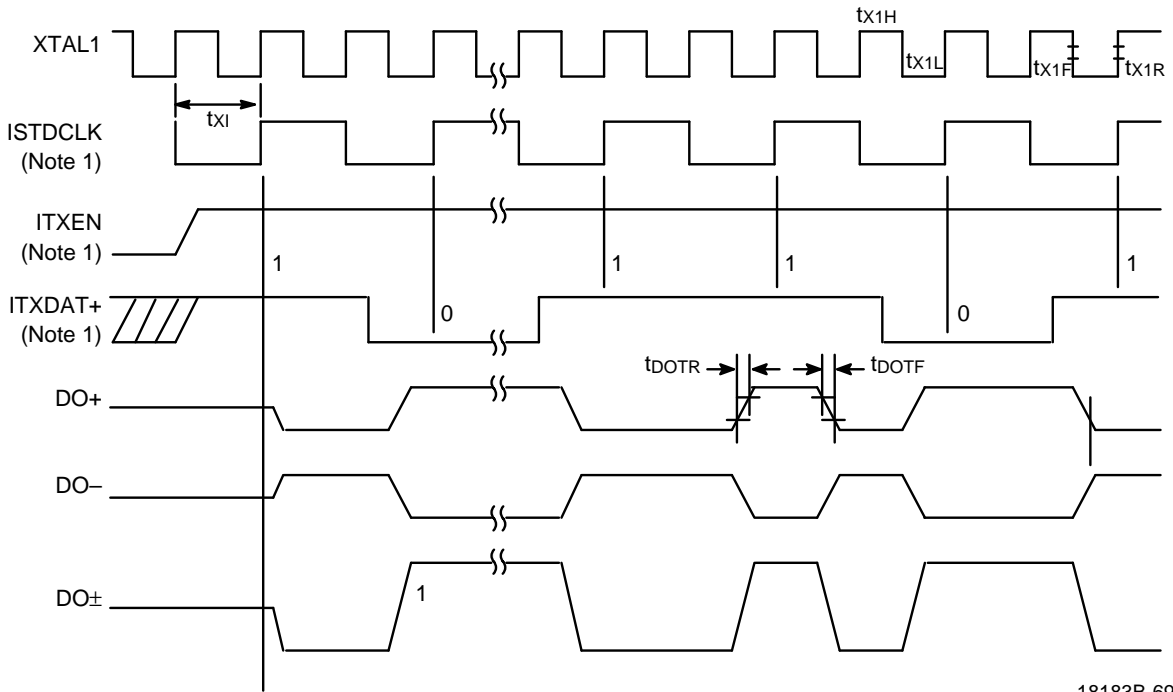
SWITCHING WAVEFORMS: JTAG (IEEE 1149.1) INTERFACE



18183B-68

Test Access Port Timing

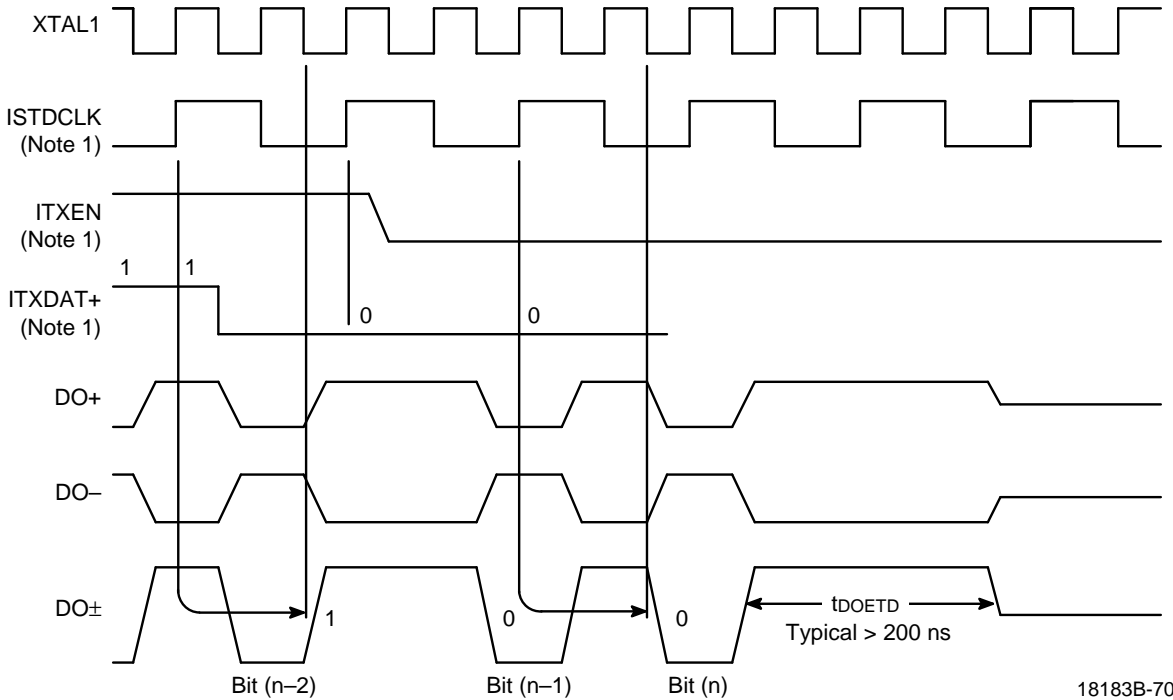
SWITCHING WAVEFORMS: AUI



18183B-69

Note:
1. Internal signal and is shown for clarification only.

Transmit Timing—Start of Packet

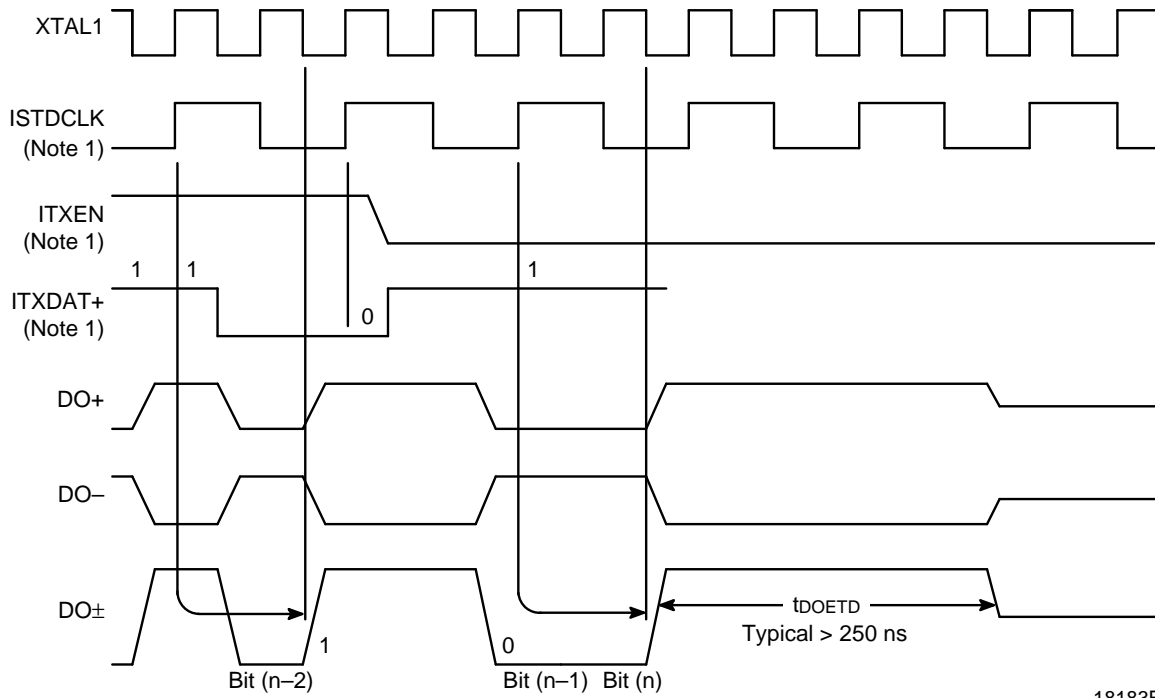


18183B-70

Note:
1. Internal signal and is shown for clarification only.

Transmit Timing—End of Packet (Last Bit = 0)

SWITCHING WAVEFORMS: AUI



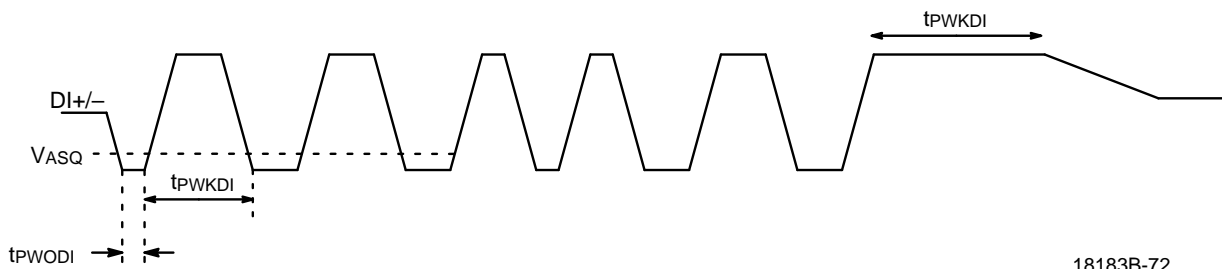
18183B-71

Note:

1. Internal signal and is shown for clarification only.

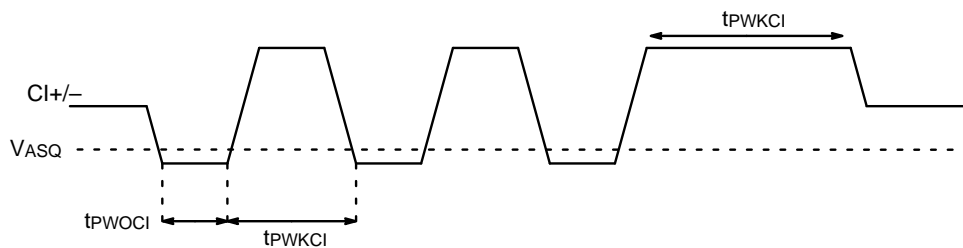
Transmit Timing—End of Packet (Last Bit = 1)

SWITCHING WAVEFORMS: AUI



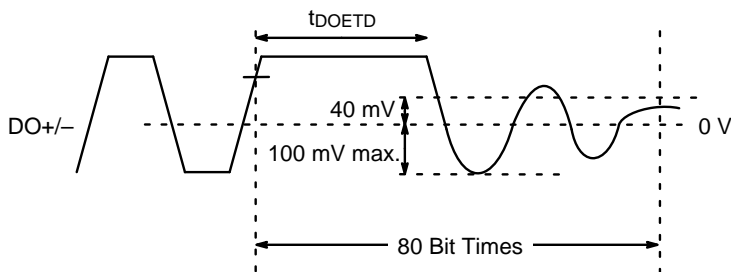
18183B-72

Receive Timing Diagram



18183B-73

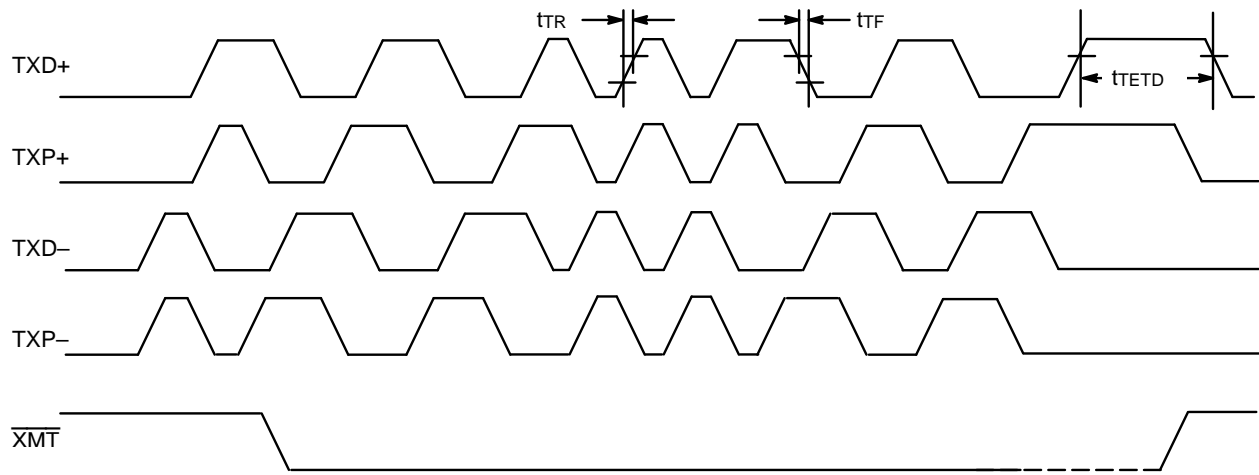
Collision Timing Diagram



18183B-74

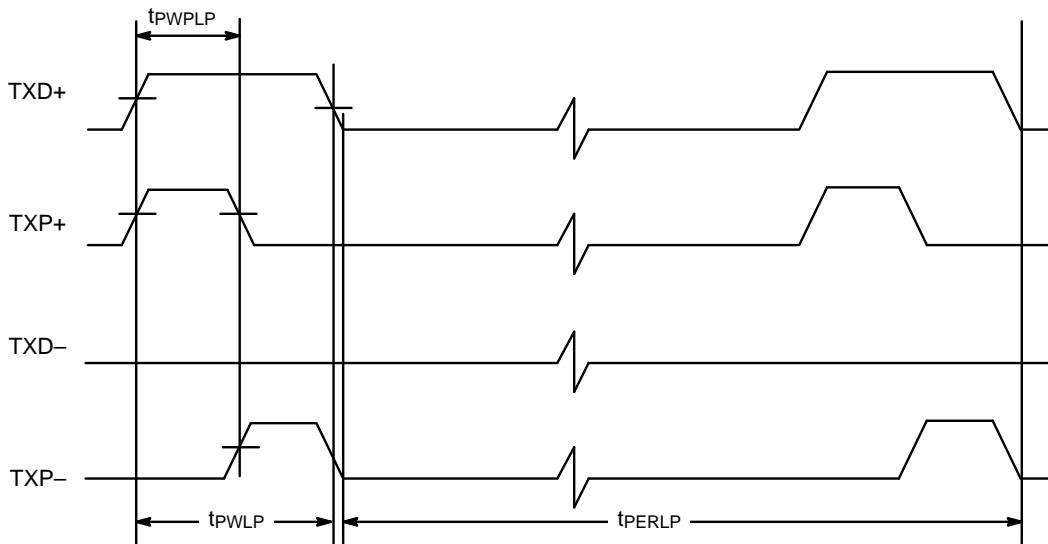
Port DO ETD Waveform

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



18183B-75

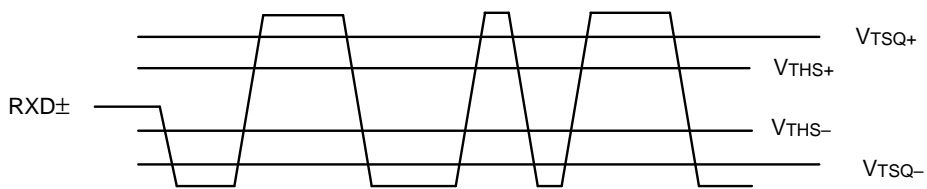
Transmit Timing



18183B-76

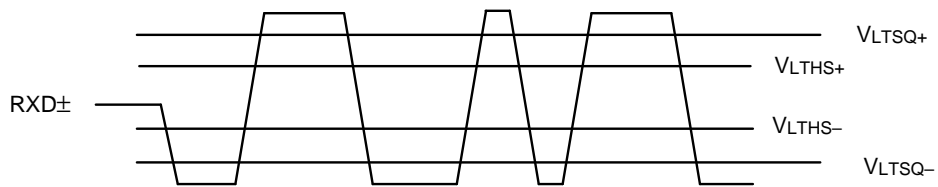
Idle Link Test Pulse

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



18183B-77

Receive Thresholds (LRT = 0 in CSR15 bit 9)



18183B-78

Receive Thresholds (LRT = 1 in CSR15 bit 9)



PCnet-ISA⁺ Compatible Media Interface Modules

PCnet-ISA⁺ COMPATIBLE 10BASE-T FILTERS AND TRANSFORMERS

The table below provides a sample list of PCnet-ISA⁺ compatible 10BASE-T filter and transformer modules

available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

| Manufacturer | Part No. | Package | Filters and Transformers | Filters Transformers and Choke | Filters Transformers Dual Choke | Filters Transformers Dual Chokes |
|--------------------------|--------------|-----------------|--------------------------|--------------------------------|---------------------------------|----------------------------------|
| Bel Fuse | A556-2006-DE | 16-pin 0.3" DIL | √ | | | |
| Bel Fuse | 0556-2006-00 | 14-pin SIP | √ | | | |
| Bel Fuse | 0556-2006-01 | 14-pin SIP | | | √ | |
| Bel Fuse | 0556-6392-00 | 16-pin 0.5" DIL | | | √ | |
| Halo Electronics | FD02-101G | 16-pin 0.3" DIL | √ | | | |
| Halo Electronics | FD12-101G | 16-pin 0.3" DIL | | √ | | |
| Halo Electronics | FD22-101G | 16-pin 0.3" DIL | | | √ | |
| PCA Electronics | EPA1990A | 16-pin 0.3" DIL | √ | | | |
| PCA Electronics | EPA2013D | 16-pin 0.3" DIL | | √ | | |
| PCA Electronics | EPA2162 | 16-pin 0.3" SIP | | | √ | |
| Pulse Engineering | PE-65421 | 16-pin 0.3" DIL | √ | | | |
| Pulse Engineering | PE-65434 | 16-pin 0.3" SIL | | | √ | |
| Pulse Engineering | PE-65445 | 16-pin 0.3" DIL | | | √ | |
| Pulse Engineering | PE-65467 | 12-pin 0.5" SMT | | | | √ |
| Valor Electronics | PT3877 | 16-pin 0.3" DIL | √ | | | |
| Valor Electronics | FL1043 | 16-pin 0.3" DIL | | | √ | |

PCnet-ISA⁺ Compatible AUI Isolation Transformers

The table below provides a sample list of PCnet-ISA⁺ compatible AUI isolation transformers available from

various vendors. Contact the respective manufacturer for a complete and updated listing of components.

| Manufacturer | Part No. | Package | Description |
|--------------------------|--------------|-----------------|-------------|
| Bel Fuse | A553-0506-AB | 16-pin 0.3" DIL | 50 μ H |
| Bel Fuse | S553-0756-AE | 16-pin 0.3" SMD | 75 μ H |
| Halo Electronics | TD01-0756K | 16-pin 0.3" DIL | 75 μ H |
| Halo Electronics | TG01-0756W | 16-pin 0.3" SMD | 75 μ H |
| PCA Electronics | EP9531-4 | 16-pin 0.3" DIL | 50 μ H |
| Pulse Engineering | PE64106 | 16-pin 0.3" DIL | 50 μ H |
| Pulse Engineering | PE65723 | 16-pin 0.3" SMT | 75 μ H |
| Valor Electronics | LT6032 | 16-pin 0.3" DIL | 75 μ H |
| Valor Electronics | ST7032 | 16-pin 0.3" SMD | 75 μ H |



PCnet-ISA+ Compatible DC/DC Converters

The table below provides a sample list of PCnet-ISA+ compatible DC/DC converters available from various

vendors. Contact the respective manufacturer for a complete and updated listing of components.

| Manufacturer | Part No. | Package | Voltage | Remote On/Off |
|--------------------------|------------|------------|---------|---------------|
| Halo Electronics | DCU0-0509D | 24-pin DIP | 5/-9 | No |
| Halo Electronics | DCU0-0509E | 24-pin DIP | 5/-9 | Yes |
| PCA Electronics | EPC1007P | 24-pin DIP | 5/-9 | No |
| PCA Electronics | EPC1054P | 24-pin DIP | 5/-9 | Yes |
| PCA Electronics | EPC1078 | 24-pin DIP | 5/-9 | Yes |
| Valor Electronics | PM7202 | 24-pin DIP | 5/-9 | No |
| Valor Electronics | PM7222 | 24-pin DIP | 5/-9 | Yes |

MANUFACTURER CONTACT INFORMATION

Contact the following companies for further information on their products:

| Company | | U.S. and Domestic | Asia | Europe |
|---------------------------------------|----------------|----------------------------------|------------------------------|--------------------------------|
| Bel Fuse | Phone: FAX: | (201) 432-0463 (201) 432-9542 | 852-328-5515 852-352-3706 | 33-1-69410402 33-1-69413320 |
| Halo Electronics | Phone: FAX: | (415) 969-7313 (415) 367-7158 | 65-285-1566 65-284-9466 | |
| PCA Electronics (HPC in Hong Kong) | Phone: FAX: | 818-892-0761 818-894-5791 | 852-553-0165 852-873-1550 | 33-1-44894800 33-1-42051579 |
| Pulse Engineering | Phone: FAX: | (619) 674-8100 (619) 675-8262 | 852-425-1651 852-480-5974 | 353-093-24107 353-093-24459 |
| Valor Electronics | Phone: FAX: | (619) 537-2500 (619) 537-2525 | 852-513-8210 852-513-8214 | 49-89-6923122 49-89-6926542 |



Layout Recommendations for Reducing Noise

DECOUPLING LOW-PASS R/C FILTER DESIGN

The PCnet-ISA+ controller is an integrated, single-chip Ethernet controller, which contains both digital and analog circuitry. The analog circuitry contains a high speed Phase-Locked Loop (PLL) and Voltage Controlled Oscillator (VCO). Because of the mixed signal characteristics of this chip, some extra precautions must be taken into account when designing with this device.

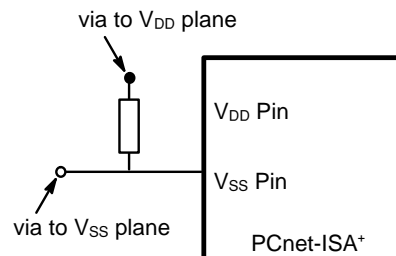
Described in this section is a simple decoupling low-pass R/C filter that can significantly increase noise immunity of the PLL circuit, thus, prevent noise from disrupting the VCO. Bit error rate, a common measurement of network performance, as a result can be drastically reduced. In certain cases the bit error rate can be reduced by orders of magnitude.

Implementation of this filter is not necessary to achieve a functional product that meets the IEEE 802.3 specification and provides adequate performance. However, this filter will help designers meet those specifications with more margin.

Digital Decoupling

The DVSS pins that are sinking the most current are those that provide the ground for the ISA bus output signals since these outputs require 24 mA drivers. The DVSS10 and DVSS12 pins provide the ground for the internal digital logic. In addition, DVSS11 provides ground for the internal digital and for the Input and I/O pins.

The CMOS technology used in fabricating the PCnet-ISA+ controller employs an n-type substrate. In this technology, all V_{DD} pins are electrically connected to each other internally. Hence, in a four-layer board, when decoupling between V_{DD} and critical V_{SS} pins, the specific V_{DD} pin that you connect to is not critical. In fact, the V_{DD} connection of the decoupling capacitor can be made directly to the power plane, near the closest V_{DD} pin to the V_{SS} pin of interest. However, we recommend that the V_{SS} connection of the decoupling capacitor be made directly to the V_{SS} pin of interest as shown.



AMD recommends that at least one low-frequency bulk decoupling capacitor be used in the area of the PCnet-ISA+ controller. 22 μF capacitors have worked well for this. In addition, a total of four or five 0.1 μF capacitors have proven sufficient around the DVSS and DVDD pins that supply the drivers of the ISA bus output pins.

Analog Decoupling

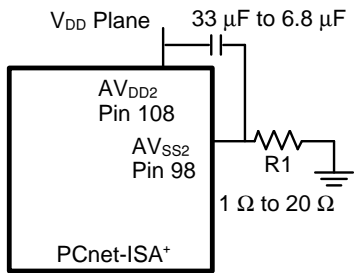
The most critical pins are the analog supply and ground pins. All of the analog supply and ground pins are located in one corner of the device. Specific requirements of the analog supply pins are listed below.

AVSS1 and AVDD3

These pins provide the power and ground for the Twisted Pair and AUI drivers. Hence, they are very noisy. A dedicated 0.1 μF capacitor between these pins is recommended.

AVSS2 and AVDD2

These pins are the most critical pins on the PCnet-ISA+ controller because they provide the power and ground for the PLL portion of the chip. The V_{CO} portion of the PLL is sensitive to noise in the 60 kHz–200 kHz. range. To prevent noise in this frequency range from disrupting the VCO, AMD strongly recommends that the low-pass filter shown below be implemented on these pins. Tests using this filter have shown significantly increased noise immunity and reduced Bit Error Rate (BER) statistics in designs using the PCnet-ISA+ controller.



To determine the value for the resistor and capacitor, the formula is:

$$R * C \geq 88$$

Where R is in ohms and C is in microfarads. Some possible combinations are given below. To minimize the

voltage drop across the resistor, the R value should not be more than 20 Ω.

| R | C |
|-------|--------|
| 2.7 Ω | 33 μF |
| 4.3 Ω | 22 μF |
| 6.8 Ω | 15 μF |
| 10 Ω | 10 μF |
| 20 Ω | 6.8 μF |

AVSS2 and AVDD2/AVDD4

These pins provide power and ground for the AUI and twisted pair receive circuitry. No specific decoupling has been necessary on these pins.



Sample Configuration File

SAMPLE CONFIGURATION FILE

The following is a sample configuration file for the PCnet-ISA+ device used in an AMD Ethernet card. This card requires one DMA channel, one interrupt, one I/O port in the 0x200-0x3FF range (0x20 bytes aligned). The vendor ID of AMD is AMD. The vendor assigned part number for this card is 2100 and the serial number is 0x12345678. The card has only one logical device,

that is an ethernet controller. There are no compatible devices with this logical device. The following record should be returned by the card during the identification process.

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

```

.....
; Plug and Play Header
.....
DB 0x04                ; Vendor EISA ID Byte 0
DB 0x43                ; Vendor EISA ID Byte 1
DB 0x00                ; Vendor Assigned ID Byte 0
DB 0x21                ; Vendor Assigned ID Byte 1
DB 0x78                ; Serial Number byte 0
DB 0x56                ; Serial Number byte 1
DB 0x34                ; Serial Number byte 2
DB 0x12                ; Serial Number byte 3
DB Checksum           ; Checksum calculated on above bits
.....
; Plug and Play Version
.....
DB 0x0A                ; Small Item, Plug and Play version
DB 0x10                ; BCD major version [7:4] = 1
                        ; BCD minor version [3:0] = 0
DB 0x00                ; Vendor specific version number
.....
; Identifier String
.....
DB 0x82                ; Large Item, Type Identifier string (ANSI)
DB 0x1c                ; Length Byte 0 (28 bytes)
DB 0x00                ; Length Byte 1
DB "AMD Ethernet Network Adapter" ; Identifier String

```

```

.....
; Logical Device ID
.....
DB 0x15 ; Small Item, Type Logical Device ID
DB 0x11 ; Logical Device ID byte 0
DB 0x11 ; Logical Device ID byte 1
DB 0x22 ; Logical Device ID byte 2
DB 0x22 ; Logical Device ID byte 3
DB 0x01 ; Logical Device Flags [0] - required for boot
.....
; I/O Port Descriptor
.....
DB 0x47 ; Small Item, type I/O Port
DB 0x00 ; Information, [0] = 0, 10 bit Decode
DB 0x00 ; Minimum Base Address [07:00]
DB 0x02 ; Minimum Base Address [15:08]
DB 0xE0 ; Maximum Base Address [07:00]
DB 0x03 ; Maximum Base Address [15:08]
DB 0x20 ; Base Address Increment (32 ports)
DB 0x18 ; Number of ports required
.....
; DMA Descriptor
.....
DB 0x2A ; Small Item, type DMA Format
DB 0xE8 ; DMA channel mask ch 3, 5, 6, 7
DB 0x06 ; 16-Bit only, Bus Master
.....
; IRQ Format
.....
DB 0x23 ; Small Item, type IRQ Format
DB 0x38 ; IRQs supported [7:0] } 3, 4, 5
DB 0x9E ; IRQs supported [15:8] } 9, 10, 11, 12, 15
DB 0x01 ; Information: High true, edge
.....
; End Tag
.....
DB 0x78 ; Small item, type END TAG
DB Checksum ; Checksum

```




Alternative Method for Initialization

The PCnet-ISA+ controller may be initialized by performing I/O writes only. That is, data can be written directly to the appropriate control and status registers (CSR) instead of reading from the Initialization Block in memory. The registers that must be written are shown in the table below. These are followed by writing the START bit in CSR0.

| Control and Status Register | Comment |
|-----------------------------|--------------|
| CSR8 | LADRF[15:0] |
| CSR9 | LADRF[31:16] |
| CSR10 | LADRF[47:32] |
| CSR11 | LADRF[63:48] |
| CSR12 | PADR[15:0] |
| CSR13 | PADR[31:16] |
| CSR14 | PADR[47:32] |
| CSR15 | Mode |
| CSR24-25 | BADR |
| CSR30-31 | BADX |
| CSR47 | POLLINT |
| CSR76 | RCVRL |
| CSR78 | XMTRL |

Note: The INIT bit must not be set or the initialization block will be accessed instead.



Introduction of the Look Ahead Packet Processing (LAPP) Concept

A driver for the PCnet-ISA⁺ controller would normally require that the CPU copy receive frame data from the controller's buffer space to the application's buffer space *after* the entire frame has been received by the controller. For applications that use a ping-pong windowing style, the traffic on the network will be halted until the current frame has been completely processed by the entire application stack. This means that the time between last byte of a receive frame arriving at the client's Ethernet controller and the client's transmission of the first byte of the next outgoing frame will be separated by:

- 1) the time that it takes the client's CPU's interrupt procedure to pass software control from the current task to the driver
- 2) plus the time that it takes the client driver to pass the header data to the application and request an application buffer
- 3) plus the time that it takes the application to generate the buffer pointer and then return the buffer pointer to the driver
- 4) plus the time that it takes the client driver to transfer all of the frame data from the controller's buffer space into the application's buffer space and then call the application again to process the complete frame
- 5) plus the time that it takes the application to process the frame and generate the next outgoing frame
- 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 CSRO

The sum of these times can often be about the same as the time taken to actually transmit the frames on the wire, thereby yielding a network utilization rate of less than 50%.

An important thing to note is that the PCnet-ISA⁺ controller's data transfers to its buffer space are such that the system bus is needed by the PCnet-ISA⁺ controller for approximately 4% of the time. This leaves 96% of the system bus bandwidth for the CPU to perform some of the inter-frame operations *in advance of the completion of network receive activity*, if possible. The question then becomes: how much of the tasks that need to be performed between reception of a frame and transmission of the next frame can be performed *before*

the reception of the frame actually ends at the network, and how can the CPU be instructed to perform these tasks during the network reception time?

The answer depends upon exactly what is happening in the driver and application code, but the steps that can be performed at the same time as the receive data are arriving include as much as the first 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 PCnet-ISA⁺ controller could place the frame data directly into the application's buffer space; (i.e. eliminate the need for step four.) In order to make this work, it is necessary that the application buffer pointer be determined before the frame has completely arrived, then the buffer pointer in the next descriptor for the receive frame would need to be modified in order to direct the PCnet-ISA⁺ 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 four unchanged in that the CPU is still required to perform the copy operation, but it allows a large portion of the copy operation to be done before the frame has been completely received by the controller, (i.e. the CPU can perform the copy operation of the receive data from the PCnet-ISA⁺ 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 four to be performed concurrently with the arrival of network data, rather than sequentially, following the end of network receive activity.

Outline of the LAPP Flow:

This section gives a suggested outline for a driver that utilizes the LAPP feature of the PCnet-ISA⁺ controller.

Note: The labels in the following text are used as references in the timeline diagram that follows.

SETUP:

The driver should set up descriptors in groups of 3, with the OWN and STP bits of each set of three descriptors to read as follows: 11b, 10b, 00b.

An option bit (LAPPEN) exists in CSR3, bit position 5. The software should set this bit. When set, the LAPPEN bit directs the PCnet-ISA+ to generate an INTERRUPT when STP has been written to a receive descriptor by the PCnet-ISA+ controller.

FLOW:

The PCnet-ISA+ controller polls the current receive descriptor at some point in time before a message arrives. The PCnet-ISA+ controller determines that this receive buffer is OWNed by the PCnet-ISA+ controller and it stores the descriptor information to be used when a message does arrive.

- N0: Frame preamble appears on the wire, followed by SFD and destination address.
- N1: The 64th byte of frame data arrives from the wire. This causes the PCnet-ISA+ controller to begin frame data DMA operations to the first buffer.
- C0: When the 64th byte of the message arrives, the PCnet-ISA+ controller performs a lookahead operation to the next receive descriptor. This descriptor should be owned by the PCnet-ISA+ controller.
- C1: The PCnet-ISA+ controller intermittently requests the bus to transfer frame data to the first buffer as it arrives on the wire.
- S0: The driver remains idle.
- C2: When the PCnet-ISA+ controller has completely filled the first buffer, it writes status to the first descriptor.
- C3: When the first descriptor for the frame has been written, changing ownership from the PCnet-ISA+ controller to the CPU, the PCnet-ISA+ controller will generate an SRP INTERRUPT. (This interrupt appears as a RINT interrupt in CSR0.)
- S1: The SRP INTERRUPT causes the CPU to switch tasks to allow the PCnet-ISA+ controller's driver to run.
- C4: During the CPU interrupt-generated task switching, the PCnet-ISA+ controller is performing a lookahead operation to the third descriptor. At this point in time, the third descriptor is owned by the CPU. [**Note:** *Even though the third buffer is not owned by the PCnet-ISA+ 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 the point that it discovers that it does not yet own descriptor number 3.]

- S2: The first task of the driver's interrupt service routine is to collect the header information from the PCnet-ISA+ 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 PCnet-ISA+ controller will be placing the first portion of the message into the first and second buffers. (The modified application data buffer pointer will only be directly used by the PCnet-ISA+ controller when it reaches the third buffer.) The driver will place the modified data buffer pointer into the final descriptor of the group (#3) and will grant ownership of this descriptor to the PCnet-ISA+ controller.
- C5: Interleaved with S2, S3 and S4 driver activity, the PCnet-ISA+ controller will write frame data to buffer number 2.
- S4: The driver will next proceed to copy the contents of the PCnet-ISA+ 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 PCnet-ISA+ controller to finish filling the second buffer.
- C6: At this point, knowing that it had not previously owned the third descriptor, and knowing that the current message has not ended (there is more data in the fifo), the PCnet-ISA+ controller will make a "last ditch lookahead" to the final (third) descriptor; This time, the ownership will be TRUE (i.e. the descriptor belongs to the controller), because the driver wrote the application pointer into this descriptor and then changed the ownership to give the descriptor to the PCnet-ISA+ controller back at S3. Note that if steps S1, S2 and S3 have not completed at this time, a BUFF error will result.
- C7: After filling the second buffer and performing the last chance lookahead to the next descriptor, the PCnet-ISA+ controller will write the status and change the ownership bit of descriptor number 2.
- S6: After the ownership of descriptor number 2 has been changed by the PCnet-ISA+ controller, the **next driver** poll of the 2nd descriptor will show ownership granted to the CPU. The driver now copies the data from buffer number 2 into the "middle section" of the application buffer space. This

operation is interleaved with the C7 and C8 operations.

C8: The PCnet-ISA+ controller will perform data DMA to the last buffer, whose pointer is pointing to application space. Data entering the last buffer will not need the infamous “double copy” that is required by existing drivers, since it is being placed directly into the application buffer space.

N2: The message on the wire ends.

S7: When the driver completes the copy of buffer number 2 data to the application buffer space, it begins polling descriptor number 3.

C9: When the PCnet-ISA+ controller has finished all data DMA operations, it writes status and changes ownership of descriptor number 3.

S8: The driver sees that the ownership of descriptor number 3 has changed, and it calls the application to tell the application that a frame has arrived.

S9: The application processes the received frame and generates the next TX frame, placing it into a TX buffer.

S10: The driver sets up the TX descriptor for the PCnet-ISA+ controller.

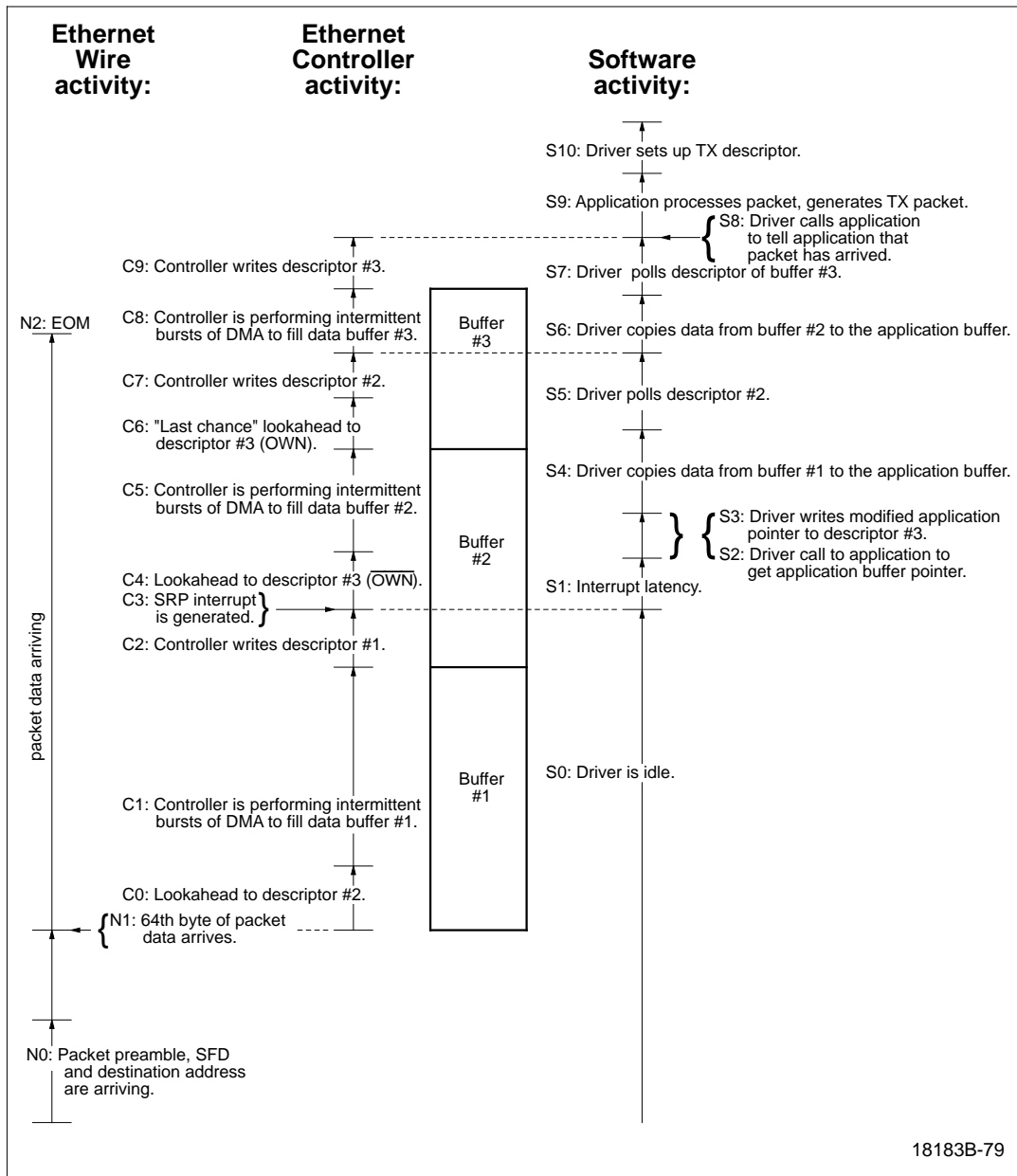


Figure 1. Look Ahead Packet Processing (LAPP) Timeline

LAPP Enable Software Requirements

Software needs to set up a receive ring with descriptors formed into groups of 3. The first descriptor of *each group* should have OWN = 1 and STP = 1, the second descriptor of *each group* should have OWN = 1 and STP = 0. The third descriptor of *each group* should have OWN = 0 and STP = 0. The size of the first buffer (as indicated in the first descriptor), should be **at least** equal to the largest expected header size; However, for maximum efficiency of CPU utilization, the first buffer size should be larger than the header size. It should be equal to the expected number of message bytes, minus the time needed for Interrupt latency and minus the application call latency, minus the time needed for the driver to write to the third descriptor, minus the time needed for the driver to copy data from buffer #1 to the application buffer space, and minus the time needed for the driver to copy data from buffer #2 to the application buffer space. Note that the time needed for the copies performed by the driver depends upon the sizes of the 2nd and 3rd buffers, and that the sizes of the second and third buffers need to be set according to the time needed for the data copy operations! This means that an iterative self-adjusting mechanism needs to be placed into the software to determine the correct buffer sizing for optimal operation. Fixed values for buffer sizes may be used; In such a case, the LAPP method will still provide a significant performance increase, but the performance increase will not be maximized.

The following diagram illustrates this setup for a receive ring size of 9:

| | |
|---------------|--|
| Descriptor #9 | OWN = 0 STP = 0 SIZE = S6 |
| Descriptor #8 | OWN = 1 STP = 0 SIZE = S1+S2+S3+S4 |
| Descriptor #7 | OWN = 1 STP = 1 SIZE = A-(S1+S2+S3+S4+S6) |
| Descriptor #6 | OWN = 0 STP = 0 SIZE = S6 |
| Descriptor #5 | OWN = 1 STP = 0 SIZE = S1+S2+S3+S4 |
| Descriptor #4 | OWN = 1 STP = 1 SIZE = A-(S1+S2+S3+S4+S6) |
| Descriptor #3 | OWN = 0 STP = 0 SIZE = S6 |
| Descriptor #2 | OWN = 1 STP = 0 SIZE = S1+S2+S3+S4 |
| Descriptor #1 | OWN = 1 STP = 1 SIZE = A-(S1+S2+S3+S4+S6) |

LAPP Enable Rules for Parsing of Descriptors

When using the LAPP method, software must use a modified form of descriptor *parsing* as follows:

Software will examine OWN and STP to determine where a RCV frame begins. RCV frames will only begin in buffers that have OWN = 0 and STP = 1.

Software shall assume that a frame continues until it finds *either* ENP = 1 or ERR= 1.

Software must discard all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for the beginning of a new frame; ENP and ERR should be ignored by software during this search.

Software cannot change an STP value in the receive descriptor ring after the initial setup of the ring is complete, even if software has ownership of the STP descriptor *unless* the previous STP descriptor in the ring is *also* OWNED by the software.

When LAPPEN = 1, then hardware will use a modified form of descriptor *parsing* as follows:

The controller will examine OWN and STP to determine where to begin placing a RCV frame. A new RCV frame will only begin in a buffer that has OWN = 1 and STP = 1.

The controller will always obey the OWN bit for determining whether or not it may use the next buffer for a chain.

The controller will always mark the end of a frame with *either* ENP = 1 or ERR= 1.

- A = Expected message size in bytes
- S1 = Interrupt latency
- S2 = Application call latency
- S3 = Time needed for driver to write to third descriptor
- S4 = Time needed for driver to copy data from buffer #1 to application buffer space
- S6 = Time needed for driver to copy data from buffer #2 to application buffer space

Note that the times needed for tasks S1, S2, S3, S4, and S6 should be divided by 0.8 ms to yield an equivalent number of network byte times before subtracting these quantities from the expected message size A.

18183B-80

Figure 2. LAPP 3 Buffer Grouping

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.

- 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 Arrived | | | After the Frame Has Arrived | | | Comments (After Frame Arrival) |
|-------------------|--------------------------|-----|------|-----------------------------|-----|------|--------------------------------|
| | OWN | STP | ENP* | OWN | STP | ENP* | |
| 1 | 1 | 1 | X | 0 | 1 | 0 | Bytes 1–800 |
| 2 | 1 | 0 | X | 0 | 0 | 0 | Bytes 801–1000 |
| 3 | 0 | 0 | X | 0 | 0 | 1 | Bytes 1001–1060 |
| 4 | 1 | 1 | X | 1 | 1 | X | Controller's current location |
| 5 | 1 | 0 | X | 1 | 0 | X | Not yet used |
| 6 | 0 | 0 | X | 0 | 0 | X | Not yet used |
| etc. | 1 | 1 | X | 1 | 1 | X | Not yet used |

*ENP or ERR

- 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 Arrived | | | After the Frame Has Arrived | | | Comments (After Frame Arrival) |
|-------------------|--------------------------|-----|------|-----------------------------|-----|------|--------------------------------|
| | OWN | STP | ENP* | OWN | STP | ENP* | |
| 1 | 1 | 1 | X | 0 | 1 | 0 | Bytes 1–800 |
| 2 | 1 | 0 | X | 0 | 0 | 1 | Bytes 801–900 |
| 3 | 0 | 0 | X | 0 | 0 | ?** | Discarded buffer |
| 4 | 1 | 1 | X | 1 | 1 | X | Controller's current location |
| 5 | 1 | 0 | X | 1 | 0 | X | Not yet used |
| 6 | 0 | 0 | X | 0 | 0 | X | Not yet used |
| etc. | 1 | 1 | X | 1 | 1 | X | Not yet used |

*ENP or ERR

** Note that the PCnet-ISA⁺ controller might write a ZERO to ENP location in the 3rd descriptor. Here are the two possibilities:

- 1) If the controller finishes the data transfers into buffer number 2 after the driver writes the application's 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 application's modified buffer pointer into the third descriptor, then the controller will complete the frame in buffer number two and then skip the then un-owned third buffer. In this case, the PCnet-ISA⁺ 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.

- 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.

| Descriptor Number | Before the Frame Arrived | | | After the Frame Has Arrived | | | Comments (After Frame Arrival) |
|-------------------|--------------------------|-----|------|-----------------------------|-----|------|--------------------------------|
| | OWN | STP | ENP* | OWN | STP | ENP* | |
| 1 | 1 | 1 | X | 0 | 1 | 1 | Bytes 1–100 |
| 2 | 1 | 0 | X | 0 | 0 | 0*** | Discarded buffer |
| 3 | 0 | 0 | X | 0 | 0 | ?** | Discarded buffer |
| 4 | 1 | 1 | X | 1 | 1 | X | Controller's current location |
| 5 | 1 | 0 | X | 1 | 0 | X | Not yet used |
| 6 | 0 | 0 | X | 0 | 0 | X | Not yet used |
| etc. | 1 | 1 | X | 1 | 1 | X | Not yet used |

*ENP or ERR

** Same as note in case 2 above, except that in this case, it is very unlikely that the driver can respond to the interrupt and get the pointer from the application before the PCnet-ISA⁺ controller has completed its poll of the next descriptors. This means that for almost all occurrences of this case, the PCnet-ISA⁺ controller will not find the OWN bit set for this descriptor and therefore, the ENP bit will almost always contain the old value, since the PCnet-ISA⁺ controller will not have had an opportunity to modify it.

*** Note that even though the PCnet-ISA⁺ 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.

Buffer Size Tuning

For maximum performance, buffer sizes should be adjusted depending upon the expected frame size and the values of the interrupt latency and application call latency. The best driver code will minimize the CPU utilization while also minimizing the latency from frame end on the network to frame sent to application from driver (frame latency). These objectives are aimed at increasing throughput on the network while decreasing CPU utilization.

Note that the buffer sizes in the ring may be altered at any time that the CPU has ownership of the corresponding descriptor. The best choice for buffer sizes will maximize the time that the driver is swapped out, while minimizing the time from the last byte written by the PCnet-ISA⁺ controller to the time that the data is passed from the driver to the application. In the diagram, this corresponds to maximizing S0, while minimizing the time between C9 and S8. (The timeline happens to show a minimal time from C9 to S8.)

Note that by increasing the size of buffer number 1, we increase the value of S0. However, when we increase the size of buffer number 1, we also increase the value of S4. If the size of buffer number 1 is too large, then the driver will not have enough time to perform tasks S2, S3, S4, S5 and S6. The result is that there will be delay from the execution of task C9 until the execution of task S8. A

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 the software should adjust the buffer size to a smaller value by, subtracting “y” bytes from the buffer count. Experiments with such a tuning mechanism must be performed to determine the best values for “X” and “y.”

Note whenever the size of buffer number 1 is adjusted, buffer sizes for buffer number 2 and buffer 3 should also be adjusted.

In some systems the typical mix of receive frames on a network for a client application consists mostly of large data frames, with very few small frames. In this case, for maximum efficiency of buffer sizing, *when a frame arrives under a certain size limit, the driver should **not** adjust the buffer sizes in response to the short frame.*

An Alternative LAPP Flow – the TWO Interrupt Method

An alternative to the above suggested flow is to use two interrupts, one at the start of the Receive frame and the other at the end of the receive frame, instead of just looking for the SRP interrupt as was described above. This alternative attempts to reduce the amount of time that the software “wastes” while polling for descriptor own bits. This time would then be available for other CPU tasks. It also minimizes the amount of time the CPU needs for data copying. This savings can be applied to other CPU tasks.

The time from the end of frame arrival on the wire to delivery of the frame to the application is labeled as frame latency. For the one-interrupt method, frame latency is minimized, while CPU utilization increases. For the two-interrupt method, frame latency becomes greater, while CPU utilization decreases.

Note that some of the CPU time that can be applied to non-Ethernet tasks is used for task switching in the CPU. One task switch is required to swap a non-Ethernet task into the CPU (after S7A) and a second task switch is needed to swap the Ethernet driver back in again (at S8A). If the time needed to perform these task switches exceeds the time saved by not polling descriptors, then there is a net loss in performance with this method. Therefore, the NEW WORD method implemented should be carefully chosen.

Figure 3 shows the event flow for the two-interrupt method.

Figure 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.

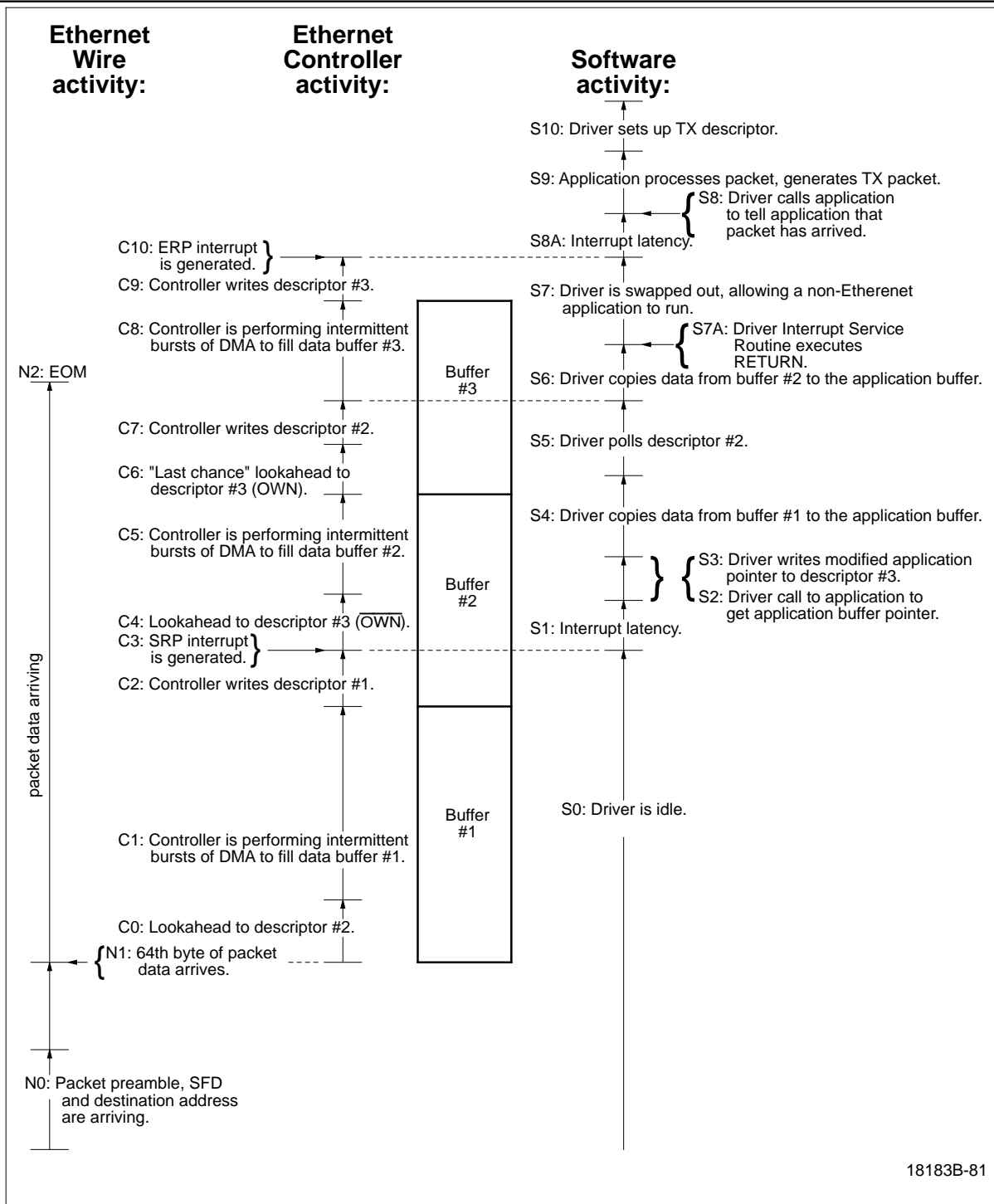


Figure 3. LAPP Timeline for TWO-INTERRUPT Method



Some Characteristics of the XXC56 Serial EEPROMs

SWITCHING CHARACTERISTICS of a TYPICAL XXC56 SERIAL EEPROM INTERFACE

Applicable over recommended operating range from $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$, $V_{CC} = +1.8\text{ V}$ to $+5.5\text{ V}$, $C_L = 1\text{ TTL Gate}$ and 100 pF (unless otherwise noted)

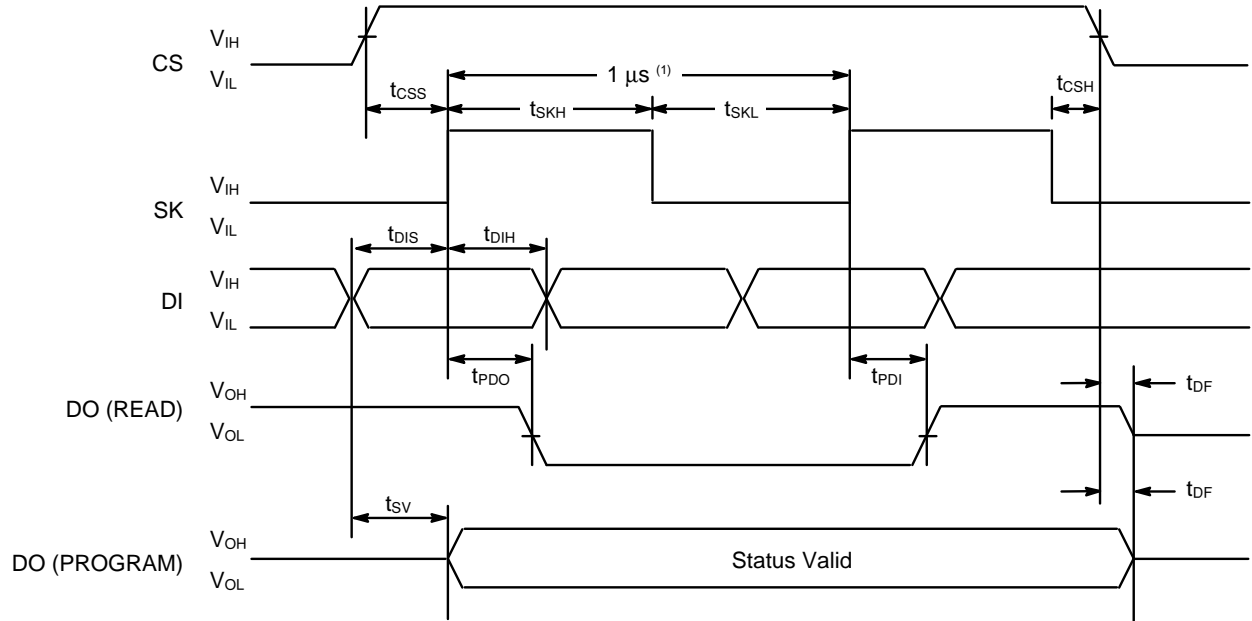
| Parameter Symbol | Parameter Description | Test Conditions | Min | Max | Unit |
|------------------|----------------------------|--------------------------------|--------------------|------|--------|
| f_{SK} | SK Clock Frequency | | 0 | 0.5 | MHz |
| t_{SKH} | SK High Time | (Note 1) | 500 | | ns |
| t_{SKL} | SK Low Time | (Note 1) | 500 | | ns |
| t_{CS} | Minimum CS Low Time | (Note 2) | 500 | | ns |
| t_{CSS} | CS Setup Time | Relative to SK | 100 | | ns |
| t_{DIS} | DI Setup Time | Relative to SK | 200 | | ns |
| t_{CSH} | CS Hold Time | Relative to SK | 0 | | ns |
| t_{DIH} | DI Hold Time | Relative to SK | 200 | | ns |
| t_{PD1} | Output Delay to '1' | AC Test | | 1000 | ns |
| t_{PD0} | Output Delay to '0' | AC Test | | 1000 | ns |
| t_{SV} | CS to Status Valid | AC Test | | 1000 | ns |
| t_{DF} | CS to DO in High Impedance | AC Test; $CS = V_{IL}$ | | 200 | ns |
| t_{WP} | Write Cycle Time | | | 10 | ms |
| | Endurance | Number of Data Changes per Bit | Typical 100,000 | | Cycles |

Notes:

- The SK frequency specifies a minimum SK clock period of $2\mu\text{s}$, therefore in an SK clock cycle $t_{SKH} + t_{SKL}$ must be greater than or equal to $2\mu\text{s}$. For example, if the $t_{SKL} = 500\text{ ns}$ then the minimum $t_{SKH} = 1.5\mu\text{s}$ in order to meet the SK frequency specification.
- CS must be brought low for a minimum of 500 ns (t_{CS}) between consecutive instruction cycles.

INSTRUCTION SET FOR THE XXC56 SERIES OF EEPROMs

| Instruction | SB | Op Code | Address | | Data | | Comments |
|-------------|----|---------|------------|------------|-------|--------|---|
| | | | x8 | x16 | x8 | x16 | |
| READ | 1 | 10 | A8–A0 | A7–A0 | | | Reads data stored in memory, at specified address |
| EWEN | 1 | 00 | 11XXXXXXXX | 11XXXXXXXX | | | Write enable must precede all programming modes |
| ERASE | 1 | 11 | A8–A0 | A7–A0 | | | Erases memory location An–A0 |
| WRITE | 1 | 01 | A0–A0 | A7–A0 | D7–D0 | D15–D0 | Writes memory location An–A0 |
| ERAL | 1 | 00 | 10XXXXXXXX | 10XXXXXXXX | | | Erases all memory locations. Valid only at $V_{CC} = 4.5\text{ V}$ to 5.5 V |
| WRAL | 1 | 00 | 01XXXXXXXX | 01XXXXXXXX | D7–D0 | D15–D0 | Writes all memory locations. Valid when $V_{CC} = 5.0\text{ V} \pm 10\%$ and Disable Register cleared |
| EWDS | 1 | 00 | 00XXXXXXXX | 00XXXXXXXX | | | Disables all programming instructions |



Note:

1. This is the minimum SK period.

18183B-57

**Typical XXC56 Series
Serial EEPROM Control Timing**