PRELIMINARY

AMD

Am79C976

PCnet-PRO™
10/100 Mbps PCI Ethernet Controller

DISTINCTIVE CHARACTERISTICS

- Integrated Fast Ethernet controller for the Peripheral Component Interconnect (PCI) bus
  - 32-bit glueless PCI host interface
  - Supports PCI clock frequency from DC to 33 MHz independent of network clock
  - Supports network operation with PCI clock from 15 MHz to 33 MHz
  - High performance bus mastering architecture with integrated Direct Memory Access (DMA) Buffer Management Unit for low CPU and bus utilization
  - PCI specification revision 2.2 compliant
  - Supports PCI Subsystem/Subvendor ID/Vendor ID programming through the EEPROM interface
  - Supports both PCI 3.3-V and 5.0-V signaling environments
  - Plug and Play compatible
  - Uses advanced PCI commands (MWI, MRL, MRM)
  - Optionally supports PCI bursts aligned to cache line boundaries
  - Supports big endian and little endian byte alignments
  - Implements optional PCI power management event (PME) pin
  - Supports 40-bit addressing (using PCI Dual Address Cycles)
- Media Independent Interface (MII) for connecting external 10/100 megabit per second (Mbps) transceivers
  - IEEE 802.3-compliant MII
  - Intelligent Auto-Poll™ external PHY status monitor and interrupt
  - Supports both auto-negotiable and non auto-negotiable external PHYs
  - Supports 10BASE-T, 100BASE-TX/FX, 100BASE-T4, and 100BASE-T2 IEEE 802.3-compliant MII PHYs at full- or half-duplex
- Full-duplex operation supported with independent Transmit (TX) and Receive (RX) channels
- Includes support for IEEE 802.1Q VLANs
  - Automatically inserts, deletes, or modifies VLAN tag
  - Optionally filters untagged frames
- Provides optional flow control features
  - Recognizes and transmits IEEE 802.3x MAC flow control frames
  - Asserts collision-based back pressure in half-duplex mode
- Provides internal Management Information Base (MIB) counters for network statistics
- Supports PC97, PC98, PC99, and Net PC requirements
  - Implements full OnNow features including pattern matching and link status wake-up
  - Implements Magic Packet™ mode
  - Magic Packet mode and the physical address loaded from EEPROM at power up without requiring PCI clock
  - Supports PCI Bus Power Management Interface Specification Version 1.1
  - Supports Advanced Configuration and Power Interface (ACPI) Specification Version 1.0
  - Supports Network Device Class Power Management Specification Version 1.0
- Large independent external TX and RX FIFOs
  - Supports up to 4 megabytes (Mbytes) external SSRAM for RX and TX frame storage
  - Programmable FIFO watermarks for both transmit and receive operations
  - Receive frame queuing for high latency PCI bus host operation
  - Programmable allocation of buffer space between transmit and receive queues

This document contains information on a product under development at Advanced Micro Devices. The information is intended to help you evaluate this product. AMD reserves the right to change or discontinue work on this proposed product without notice.

Publication# **22929** Rev: **E** Amendment/**0**
Issue Date: **September 2000**

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

Downloaded from Arrow.com.

- Dual-speed CSMA/CD (10 Mbps and 100 Mbps) Media Access Controller (MAC) compliant with IEEE/ANSI 802.3 and Blue Book Ethernet standards
- Programmable internal/external loopback capabilities
- Supports patented External Address Detection Interface (EADI) with receive frame tagging support for internetworking applications
- EEPROM interface supports jumperless design and provides through-chip programming
  - Supports full programmability of all internal registers through EEPROM mapping
- Programmable PHY reset output pin capable of resetting external PHY without needing buffering
- Integrated oscillator circuit is controlled by external crystal
- Extensive programmable LED status support
- Supports up to 16 Mbyte optional Boot PROM or Flash for diskless node application
- Look-Ahead Packet Processing (LAPP) data handling technique reduces system overhead by allowing protocol analysis to begin before the end of a receive frame
- Programmable Inter Packet Gap (IPG) to address less aggressive network MAC controllers
- Offers the Modified Back-Off algorithm to address the *Ethernet Capture Effect*
- Optionally sends and receives non-standard frames of up to 64K octets in length
- IEEE 1149.1-compliant JTAG Boundary Scan test access port interface for board-level production connectivity test
- Provides built-in self test (MBIST) for the external SSRAM
- Software compatible with AMD PCnet Family and LANCE/C-LANCE register and descriptor architecture
- Compatible with the existing PCnet Family driver and diagnostic software (except for statistics)
- Available in 208-pin PQFP package
- +3.3-V power supply with 5-V tolerant I/Os enables broad system compatibility
- Support for operation in Industrial temperature range (-40° C to +85° C) available.

Downloaded from Arrow.com.

GENERAL DESCRIPTION

The Am79C976 controller is a highly-integrated 32-bit full-duplex, 10/100-Megabit per second (Mbps) Ethernet controller solution, designed to address high-performance system application requirements. It is a flexible bus mastering device that can be used in any application, including network-ready PCs and bridge/router designs. The bus master architecture provides high data throughput and low CPU and system bus utilization. The Am79C976 controller is fabricated with advanced low-power 3.3-V CMOS process to provide low operating current for power sensitive applications.

The Am79C976 controller also has several enhancements over its predecessor, the Am79C971 PCnet-*FAST* device. In addition to providing access to a larger SSRAM, it further reduces system implementation cost by the addition of a new EEPROM programmable pin (PHY_RST) and the integration of the PAL function needed for Magic Packet application. The PHY_RST pin is implemented to reset the external PHY without increasing the load to the PCI bus and to block $\overline{\text{RST}}$ to the PHY when PG input is LOW.

The 32-bit multiplexed bus interface unit provides a direct interface to the PCI local bus, simplifying the design of an Ethernet node in a PC system. The Am79C976 controller provides the complete interface to an Expansion ROM or Flash device allowing add-on card designs with only a single load per PCI bus interface pin. With its built-in support for both little and big endian byte alignment, this controller also addresses non-PC applications. The Am79C976 controller's advanced CMOS design allows the bus interface to be connected to either a +5-V or a +3.3-V signaling environment. An IEEE 1149.1-compliant JTAG test interface for board-level testing is also provided.

The Am79C976 controller is also compliant with the PC97, PC98, PC99, and Network PC (Net PC) specifications. It includes the full implementation of the Microsoft OnNow and ACPI specifications, which are backward compatible with the Magic Packet technology, and it is compliant with the PCI Bus Power Management Interface Specification by supporting the four power management states (D0, D1, D2, and D3), the optional $\overline{\text{PME}}$ pin, and the necessary configuration and data registers.

The Am79C976 controller is ideally suited for Net PC, motherboard, network interface card (NIC), and embedded designs. It is available in a 208-pin Plastic Quad Flat Pack (PQFP) package.

The Am79C976 controller contains a bus interface unit, a DMA Buffer Management Unit, an ISO/IEC 8802-3 (IEEE 802.3)-compliant Media Access Controller (MAC), and an IEEE 802.3-compliant MII. An interface to an external RAM of up to 4 Mbytes is provided for frame storage. The MII supports IEEE 802.3-compliant full-duplex and half-duplex operations at 10 Mbps or 100 Mbps. The MII TX and RX clock signals can be stopped independently for home networking applications.

The Am79C976 controller is register compatible with the LANCE™ (Am7990) and C-LANCE™ (Am79C90) Ethernet controllers, and all Ethernet controllers in the PCnet Family *except* ILACC™ (Am79C900), including the PCnet™-ISA controller (Am79C960), PCnet™-ISA+ (Am79C961), PCnet™-ISA II (Am79C961A), PCnet™-32 (Am79C965), PCnet™-PCI (Am79C970), PCnet™-PCI II (Am79C970A), and the PCnet™-FAST (Am79C971).

The Buffer Management Unit supports the LANCE and PCnet descriptor software models.

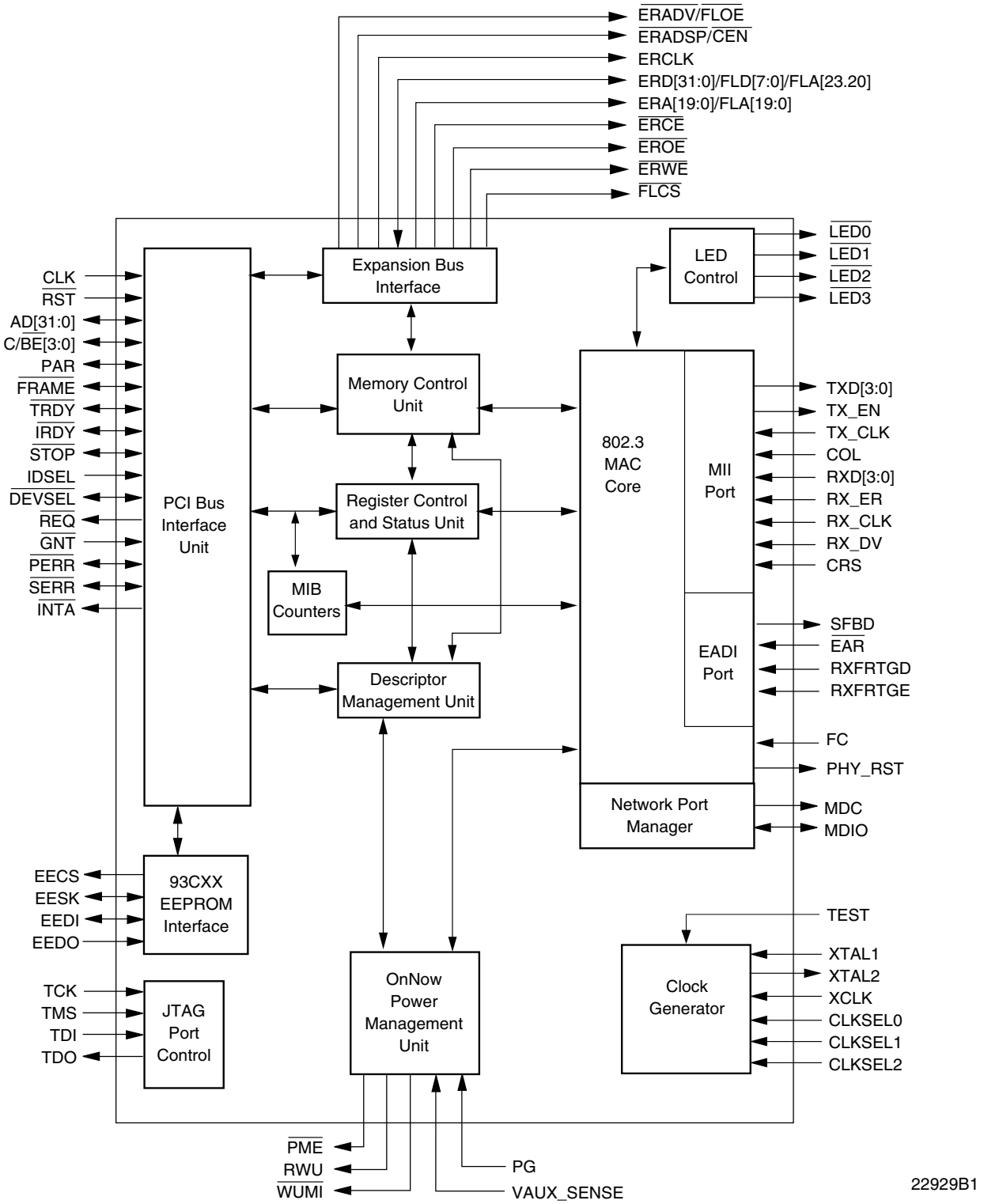
The Am79C976 controller supports auto-configuration in the PCI configuration space. Additional Am79C976 controller configuration parameters, including the unique IEEE physical address, can be read from an external nonvolatile memory (EEPROM) immediately following system reset.

In addition, the device provides programmable on-chip LED drivers for transmit, receive, collision, link integrity, Magic Packet status, activity, address match, full-duplex, or 100 Mbps status. The Am79C976 controller also provides an EADI to allow external hardware address filtering in internetworking applications and a receive frame tagging feature.

With the rise of embedded networking applications operating in harsh environments where temperatures may exceed the normal commercial temperature (0° C to +70° C) window, an industrial temperature (-40° C to +85° C) version is available. This industrial temperature version of the PCnet-PRO Ethernet controller is characterized across the industrial temperature range (-40° C to +85° C) within the published power supply specification (2.97V to 3.63V; ±10% Vcc). Thus, conformance of the PCnet-PRO performance over this temperature range is guaranteed by a design and characterization monitor.

Downloaded from Arrow.com.

BLOCK DIAGRAM



Downloaded from Arrow.com.

TABLE OF CONTENTS

DISTINCTIVE CHARACTERISTICS 1
GENERAL DESCRIPTION 3
BLOCK DIAGRAM 4
TABLE OF CONTENTS 5
LIST OF FIGURES 13
LIST OF TABLES 15
ORDERING INFORMATION 18
Standard Products 18
CONNECTION DIAGRAM (PQR208) 19
PIN DESIGNATIONS (PQR208) 20
Listed By Pin Number 20
Listed By Group 22
PIN DESCRIPTIONS 25
PCI Interface 25
Board Interface 27
EEPROM Interface 29
External Memory Interface 29
Media Independent Interface 30
External Address Detection Interface 31
IEEE 1149.1 (1990) Test Access Port Interface 32
Power Supply Pins 32
BASIC FUNCTIONS 33
System Bus Interface 33
Software Interface 33
Network Interface 33
DETAILED FUNCTIONS 34
Slave Bus Interface Unit 34
Slave Configuration Transfers 34
Slave I/O Transfers 34
Expansion ROM Transfers 38
Slave Cycle Termination 38
Disconnect When Busy 38
Disconnect Of Burst Transfer 39
Parity Error Response 40
Master Bus Interface Unit 41
Bus Acquisition 41
Bus Master DMA Transfers 41
Basic Non-Burst Read Transfer 41
Basic Burst Read Transfer 42
Basic Non-Burst Write Transfer 43
Basic Burst Write Transfer 45
DMA Burst Alignment 46
Target Initiated Termination 46
Disconnect With Data Transfer 46
Disconnect Without Data Transfer 47
Target Abort 48
Master Initiated Termination 49
Preemption During Non-Burst Transaction 49
Preemption During Burst Transaction 49
Master Abort 50
Parity Error Response 52
Initialization Block DMA Transfers 53
Descriptor DMA Transfers 53
FIFO DMA Transfers 58
Descriptor Management Unit 61
Initialization 61

Downloaded from Arrow.com.

Re-Initialization 62
Run and Suspend 62
Descriptor Management 63
Descriptor Rings 63
Polling 65
Transmit Polling 66
Receive Polling 67
Look Ahead Packet Processing 67
Software Interrupt Timer 68
Media Access Control 68
Transmit and Receive Message Data Encapsulation 68
Framing 68
Destination Address Handling 69
Error Detection 69
Media Access Management 69
Medium Allocation 69
Signal Quality Error (SQE) Test 70
Collision Handling 70
Transmit Operation 71
Transmit Function Programming 71
Automatic Pad Generation 71
Transmit FCS Generation 72
Transmit Error Conditions 72
Loss of Carrier 73
Late Collision 73
Transmit FIFO Underflow 73
Receive Operation 73
Receive Function Programming 73
Address Matching 73
Automatic Pad Stripping 74
Receive FCS Checking 75
Receive Exception Conditions 75
Statistics Counters 75
Receive Statistics Counters 76
Transmit Statistics Counters 78
VLAN Support 80
VLAN Frame Size 81
Admit Only VLAN Frames Option 81
VLAN Tags in Descriptors 81
Loopback Operation 82
Miscellaneous Loopback Features 82
Full-Duplex Operation 82
Full-Duplex Link Status LED Support 83
Media Independent Interface 83
MII Transmit Interface 83
MII Receive Interface 83
MII Network Status Interface 84
MII Management Interface 84
MII Management Frames 84
Host CPU Access to External PHY 85
Auto-Poll State Machine 86
Network Port Manager 87
Auto-Negotiation 87
Auto-Negotiation With Multiple PHY Devices 88
Operation Without MMI Management Interface 88
Regulating Network Traffic 89
MAC Control Pause Frames 89
Back Pressure 89

Downloaded from Arrow.com.

Enabling Traffic Regulation . . . 89
Hardware Control of Traffic Regulation . . . 90
Software Control of Traffic Regulation . . . 90
Programming the Pause Length Register . . . 91
PAUSE Frame Reception . . . 91
External Address Detection Interface . . . 92
External Address Detection Interface: Receive Frame Tagging . . . 93
External Memory Interface . . . 93
Expansion ROM - Boot Device Access . . . 94
Direct Flash Access . . . 96
Flash/EPROM Read . . . 96
SRAM Configuration . . . 97
EEPROM Interface . . . 98
Automatic EEPROM Read Operation . . . 98
EEPROM Auto-Detection . . . 100
Direct Access to the Interface . . . 100
EEPROM CRC Calculation . . . 100
LED Support . . . 100
Power Savings Mode . . . 101
Power Management Support . . . 101
OnNow Wake-Up Sequence . . . 102
RWU Wake-Up Sequence . . . 103
Link Change Detect . . . 103
Magic Packet Mode . . . 103
OnNow Pattern Match Mode . . . 104
Pattern Match RAM (PMR) . . . 104
IEEE 1149.1 (1990) Test Access Port Interface . . . 105
Boundary Scan Circuit . . . 105
TAP Finite State Machine . . . 105
Supported Instructions . . . 105
Boundary Scan Register . . . 106
Other Data Registers . . . 106
Reset . . . 106
H_RESET . . . 106
EE_RESET . . . 106
S_RESET . . . 107
STOP . . . 107
Power on Reset . . . 107
External PHY Reset . . . 107
Software Access . . . 108
PCI Configuration Registers . . . 108
I/O Resources . . . 108
I/O Registers . . . 109
Address PROM Space . . . 109
Reset Register . . . 109
Word I/O Mode . . . 110
Double Word I/O Mode . . . 110
USER ACCESSIBLE REGISTERS . . . 113
PCI Configuration Registers . . . 113
PCI Vendor ID Register . . . 113
PCI Device ID Register . . . 113
PCI Command Register . . . 113
PCI Status Register . . . 115
PCI Revision ID Register . . . 116
PCI Programming Interface Register . . . 116
PCI Sub-Class Register . . . 116
PCI Base-Class Register . . . 116
PCI Cache Line Size Register . . . 117

Downloaded from Arrow.com.

PCI Latency Timer Register . . . 117
PCI Header Type Register . . . 117
PCI I/O Base Address Register . . . 117
PCI Memory Mapped I/O Base Address Register . . . 118
PCI Subsystem Vendor ID Register . . . 118
PCI Subsystem ID Register . . . 119
PCI Expansion ROM Base Address Register . . . 119
PCI Capabilities Pointer Register . . . 120
PCI Interrupt Line Register . . . 120
PCI Interrupt Pin Register . . . 120
PCI MIN_GNT Register . . . 120
PCI MAX_LAT Register . . . 121
PCI Capability Identifier Register . . . 121
PCI Next Item Pointer Register . . . 121
PCI Power Management Capabilities Register (PMC) . . . 121
PCI Power Management Control/Status Register (PMCSR) . . . 122
PCI PMCSR Bridge Support Extensions Register . . . 123
PCI Data Register . . . 123
Memory-Mapped Registers . . . 124
MIB Offset . . . 124
AP_VALUE0: Auto-Poll Value0 Register . . . 124
AP_VALUE1: Auto-Poll Value1 Register . . . 124
AP_VALUE2: Auto-Poll Value2 Register . . . 125
AP_VALUE3: Auto-Poll Value3 Register . . . 125
AP_VALUE4: Auto-Poll Value4 Register . . . 125
AP_VALUE5: Auto-Poll Value5 Register . . . 125
AUTOPOLL0: Auto-Poll0 Register . . . 125
AUTOPOLL1: Auto-Poll1 Register . . . 126
AUTOPOLL2: Auto-Poll2 Register . . . 127
AUTOPOLL3: Auto-Poll3 Register . . . 128
AUTOPOLL4: Auto-Poll4 Register . . . 129
AUTOPOLL5: Auto-Poll5 Register . . . 130
BADR: Receive Ring Base Address Register . . . 130
BADX: Transmit Ring Base Address Register . . . 131
CHIPID: Chip ID Register . . . 131
CHPOLLTIME: Chain Poll Timer Register . . . 131
CMD0: Command0 . . . 132
CMD2: Command2 . . . 133
CMD3: Command3 . . . 137
CMD7: Command7 . . . 140
CTRL0: Control0 Register . . . 141
CTRL1: Control1 Register . . . 142
CTRL2: Control2 Register . . . 145
CTRL3: Control3 Register . . . 146
DATAMBIST: Memory Built-in Self-Test Access Register . . . 147
Reserved Register . . . 149
EEPROM_ACC: EEPROM Access Register . . . 150
FLASH_ADDR: Flash Address Register . . . 152
FLASH_DATA: Flash Data Register . . . 152
FLOW: Flow Control Register . . . 153
IFS1: Inter-Frame Spacing Part 1 Register . . . 154
INT0: Interrupt0 . . . 155
INTEN0: Interrupt0 Enable . . . 157
IPG: Inter-Packet Gap Register . . . 159
LADRF: Logical Address Filter Register . . . 159
LED0 Control Register . . . 160
LED1 Control Register . . . 161
LED2 Control Register . . . 161

Downloaded from Arrow.com.

LED3 Control Register 161
MAX_LAT_A: PCI Maximum Latency Alias Register 162
MIN_GNT_A: PCI Minimum Grant Alias Register 162
PADR: Physical Address Register 162
Pause Count Register 163
PCIDATA0: PCI DATA Register Zero Alias Register 163
PCIDATA1: PCI DATA Register One Alias Register 163
PCIDATA2: PCI DATA Register Two Alias Register 164
PCIDATA3: PCI DATA Register Three Alias Register 164
PCIDATA4: PCI DATA Register Four Alias Register 164
PCIDATA5: PCI DATA Register Five Alias Register 164
PCIDATA6: PCI DATA Register Six Alias Register 164
PCIDATA7: PCI DATA Register Seven Alias Register 164
PHY Access Register 165
PMAT0: OnNow Pattern Register 0 166
PMAT1: OnNow Pattern Register 1 166
PMC_A: PCI Power Management Capabilities Alias Register 167
Receive Protect Register 167
RCV_RING_LEN: Receive Ring Length Register 167
ROM_CFG: ROM Base Address Configuration Register 168
SID_A: PCI Subsystem ID Alias Register 168
SRAM Boundary Register 169
SRAM Size Register 169
STAT0: Status0 170
Software Timer Value Register 171
SVID_A: PCI Subsystem Vendor ID Alias Register 172
VID_A: PCI Vendor ID Alias Register 173
XMT_RING_LEN: Transmit Ring Length Register 173
XMTPOLLTIME: Transmit Poll Timer Register 173
RAP Register 174
RAP: Register Address Port 174
Control and Status Registers 175
CSR0: Am79C976 Controller Status and Control Register 175
CSR1: Initialization Block Address 0 177
CSR2: Initialization Block Address 1 177
CSR3: Interrupt Masks and Deferral Control 177
CSR4: Test and Features Control 179
CSR5: Extended Control and Interrupt 1 181
CSR6: Reserved 183
CSR7: Extended Control and Interrupt 2 183
CSR8: Logical Address Filter 0 186
CSR9: Logical Address Filter 1 186
CSR10: Logical Address Filter 2 186
CSR11: Logical Address Filter 3 186
CSR12: Physical Address Register 0 186
CSR13: Physical Address Register 1 187
CSR14: Physical Address Register 2 187
CSR15: Mode 187
CSR16-23: Reserved 188
CSR24: Base Address of Receive Ring Lower 189
CSR25: Base Address of Receive Ring Upper 189
CSR26-29: Reserved 189
CSR30: Base Address of Transmit Ring Lower 189
CSR31: Base Address of Transmit Ring Upper 189
CSR32-46: Reserved 190
CSR47: Transmit Polling Interval 190
CSR48: Reserved 190
CSR49: Chain Polling Interval 190

Downloaded from Arrow.com.

CSR50-57: Reserved 191
CSR58: Software Style 191
CSR59-75: Reserved 193
CSR76: Receive Ring Length 193
CSR77: Reserved 193
CSR78: Transmit Ring Length. 193
CSR79: Reserved 193
CSR80: DMA Transfer Counter and FIFO Threshold Control 193
CSR81-87: Reserved 195
CSR88: Chip ID Register Lower 195
CSR89: Chip ID Register Upper 195
CSR90-99: Reserved 195
CSR100: Bus Timeout. 195
CSR101-111: Reserved. 195
CSR112: Missed Frame Count 195
CSR113: Reserved 195
CSR114: Receive Collision Count. 195
CSR115: Reserved 195
CSR116: OnNow Power Mode Register 196
CSR117-121: Reserved 197
CSR122: Reserved 197
CSR123: Reserved 197
CSR124: Test Register 1. 197
CSR125: MAC Enhanced Configuration Control. 197
Bus Configuration Registers 199
BCR0: Master Mode Read Active 200
BCR1: Master Mode Write Active 200
BCR2: Miscellaneous Configuration 201
BCR4: LED0 Status. 201
BCR5: LED1 Status. 203
BCR6: LED2 Status. 205
BCR7: LED3 Status. 207
BCR9: Full-Duplex Control 209
BCR16: I/O Base Address Lower 209
BCR17: I/O Base Address Upper 210
BCR18: Burst and Bus Control Register 210
BCR19: EEPROM Control and Status 211
BCR20: Software Style 214
BCR22: PCI Latency Register. 216
BCR23: PCI Subsystem Vendor ID Register. 216
BCR24: PCI Subsystem ID Register 217
BCR25: SRAM Size Register 217
BCR26: SRAM Boundary Register 217
BCR27: SRAM Interface Control Register 218
BCR28: Expansion Bus Port Address Lower (Used for Flash/EPROM and SRAM Accesses) . 218
BCR29: Expansion Port Address Upper (Used for Flash/EPROM Accesses) 218
BCR30: Expansion Bus Data Port Register 219
BCR31: Software Timer Register 219
BCR32: MII Control and Status Register. 219
BCR33: MII Address Register 221
BCR34: MII Management Data Register. 222
BCR35: PCI Vendor ID Register 222
BCR36: PCI Power Management Capabilities (PMC) Alias Register 222
BCR37: PCI DATA Register Zero (DATA0) Alias Register 223
BCR38: PCI DATA Register One (DATA1) Alias Register. 223
BCR39: PCI DATA Register Two (DATA2) Alias Register 223
BCR40: PCI DATA Register Three (DATA3) Alias Register 224
BCR41: PCI DATA Register Four (DATA4) Alias Register 224

Downloaded from Arrow.com.

BCR42: PCI DATA Register Five (DATA5) Alias Register . . . 224
BCR43: PCI DATA Register Six (DATA6) Alias Register . . . 225
BCR44: PCI DATA Register Seven (DATA7) Alias Register . . . 225
BCR45: OnNow Pattern Matching Register 1 . . . 225
BCR46: OnNow Pattern Matching Register 2 . . . 226
BCR47: OnNow Pattern Matching Register 3 . . . 226
Initialization Block . . . 227
RLEN and TLEN . . . 228
RDRA and TDRA . . . 228
LADRF . . . 228
PADR . . . 228
Mode . . . 228
Receive Descriptors . . . 229
Transmit Descriptors . . . 240
REGISTER SUMMARY . . . 247
PCI Configuration Registers . . . 247
Memory-Mapped Registers . . . 248
Control and Status Registers . . . 250
Bus Configuration Registers . . . 254
REGISTER BIT CROSS REFERENCE . . . 255
REGISTER PROGRAMMING SUMMARY . . . 264
Programmable Registers . . . 264
ABSOLUTE MAXIMUM RATINGS . . . 268
OPERATING RANGES . . . 268
DC CHARACTERISTICS OVER COMMERCIAL OPERATING RANGES . . . 269
SWITCHING CHARACTERISTICS: BUS INTERFACE . . . 271
SWITCHING WAVEFORMS . . . 272
Key to Switching Waveforms . . . 272
SWITCHING TEST CIRCUITS . . . 273
SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE . . . 274
SWITCHING CHARACTERISTICS: EEPROM INTERFACE . . . 276
SWITCHING CHARACTERISTICS: JTAG TIMING . . . 278
SWITCHING CHARACTERISTICS: MEDIA INDEPENDENT INTERFACE . . . 280
SWITCHING CHARACTERISTICS: EXTERNAL ADDRESS DETECTION INTERFACE . . . 283
SWITCHING WAVEFORMS: RECEIVE FRAME TAG . . . 285
SWITCHING WAVEFORMS: EXTERNAL MEMORY INTERFACE . . . 286
PHYSICAL DIMENSIONS* . . . 287
PQFP208 . . . 287
Plastic Quad Flat Pack Trimmed and Formed . . . 287
APPENDIX A: LOOK-AHEAD PACKET PROCESSING . . . A-1
Introduction . . . A-1
Outline of LAPP Flow . . . A-2
Setup . . . A-2
Flow . . . A-2
LAPP Software Requirements . . . A-5
LAPP Rules for Parsing Descriptors . . . A-5
Some Examples of LAPP Descriptor Interaction . . . A-6
Buffer Size Tuning . . . A-7
An Alternative LAPP Flow: Two-Interrupt Method . . . A-8

Downloaded from Arrow.com.

APPENDIX B: MII MANAGEMENT REGISTERS B-1
Control Register (Register 0) B-1
Status Register (Register 1) B-2
Auto-Negotiation Advertisement Register (Register 4) B-3
Technology Ability Field Bit Assignments B-3
Auto-Negotiation Link Partner Ability Register (Register 5) B-4
INDEX INDEX-1

Downloaded from Arrow.com.

LIST OF FIGURES

Figure 1: Slave Configuration Read . . . 35
Figure 2: Slave Configuration Write . . . 35
Figure 3: Slave Read Using I/O Command . . . 36
Figure 4: Slave Write Using Memory Command . . . 37
Figure 5: Disconnect Of Slave Cycle When Busy . . . 39
Figure 6: Disconnect Of Slave Burst Transfer - No Host Wait States . . . 39
Figure 7: Disconnect Of Slave Burst Transfer - Host Inserts Wait States . . . 40
Figure 8: Address Parity Error Response . . . 40
Figure 9: Slave Cycle Data Parity Error Response . . . 41
Figure 10: Bus Acquisition . . . 42
Figure 11: Non-Burst Read Transfer . . . 43
Figure 12: Burst Read Transfer . . . 44
Figure 13: Non-Burst Write Transfer . . . 45
Figure 14: Burst Write Transfer . . . 46
Figure 15: Disconnect With Data Transfer . . . 47
Figure 16: Disconnect Without Data Transfer . . . 48
Figure 17: Target Abort . . . 49
Figure 18: Preemption During Non-Burst Transaction . . . 50
Figure 19: Preemption During Burst Transaction . . . 51
Figure 20: Master Abort . . . 52
Figure 21: Master Cycle Data Parity Error Response . . . 53
Figure 22: Descriptor Ring Read In Non-Burst Mode . . . 54
Figure 23: Descriptor Ring Read In Burst Mode . . . 55
Figure 24: Descriptor Ring Write In Non-Burst Mode . . . 57
Figure 25: Descriptor Ring Write In Burst Mode . . . 58
Figure 26: FIFO Burst Write At Start Of Unaligned Buffer . . . 59
Figure 27: FIFO Burst Write At End Of Unaligned Buffer . . . 61
Figure 28: 16-Bit Software Model . . . 64
Figure 29: 32-Bit Software Model . . . 65
Figure 30: ISO 8802-3 (IEEE/ANSI 802.3) Data Frame . . . 72
Figure 31: IEEE 802.3 Frame and Length Field Transmission Order . . . 75
Figure 32: VLAN-Tagged Frame Format . . . 81
Figure 33: Media Independent Interface . . . 84
Figure 34: Frame Format at the MII Interface Connection . . . 85
Figure 35: MII Receive Frame Tagging . . . 93
Figure 36: External SSRAM and Flash Configuration . . . 94
Figure 37: Expansion ROM Bus Read Sequence . . . 95
Figure 38: Flash Read from Expansion Bus Data Port . . . 97
Figure 39: Flash Write Sequence . . . 97
Figure 40: EEPROM Data Format . . . 99
Figure 41: EEPROM Entry Positions . . . 99
Figure 42: CRC Flow . . . 100
Figure 43: LED Control Logic . . . 101
Figure 44: OnNow Functional Diagram . . . 102
Figure 45: Pattern Match RAM . . . 105
Figure 46: PCI Expansion ROM Base Address Register . . . 119
Figure 47: Address Match Logic . . . 229
Figure 48: Normal and Tri-State Outputs . . . 273

Downloaded from Arrow.com.

Figure 49: CLK Waveform for 5 V Signaling 274
Figure 50: CLK Waveform for 3.3 V Signaling 274
Figure 51: Input Setup and Hold Timing 274
Figure 52: Output Valid Delay Timing 275
Figure 53: Output Tri-State Delay Timing 275
Figure 54: EEPROM Read Functional Timing 276
Figure 55: Automatic PREAD EEPROM Timing 277
Figure 56: JTAG (IEEE 1149.1) TCK Waveform for 5 V Signaling 278
Figure 57: JTAG (IEEE 1149.1) Test Signal Timing 279
Figure 58: Transmit Timing 281
Figure 59: Receive Timing 281
Figure 60: MDC Waveform 281
Figure 61: Management Data Setup and Hold Timing 282
Figure 62: Management Data Output Valid Delay Timing 282
Figure 63: Reject Timing - External PHY MII @ 25 MHz 283
Figure 64: Reject Timing - External PHY MII @ 2.5 MHz 284
Figure 65: Receive Frame Tag Timing with Media Independent Interface 285
Figure 66: External Memory Timing 286
Figure A-1: LAPP Timeline A-4
Figure A-2: LAPP 3 Buffer Grouping A-5
Figure A-3: LAPP Timeline for Two-Interrupt Method A-9
Figure A-4: LAPP 3 Buffer Grouping for Two-Interrupt Method A-10

Downloaded from Arrow.com.

LIST OF TABLES

Table 1: System Clock Selections . . . 28
Table 2: Slave Commands . . . 34
Table 3: PCI Commands . . . 42
Table 4: Descriptor Read Sequence . . . 55
Table 5: Descriptor Write Sequence . . . 56
Table 6: Receive Address Match . . . 74
Table 7: Receive Statistics Counters . . . 76
Table 8: Transmit Statistics Counters . . . 78
Table 9: VLAN Tag Control Command . . . 82
Table 10: VLAN Tag Type . . . 82
Table 11: Auto-Negotiation Capabilities . . . 87
Table 12: MAC Control Pause Frame Format . . . 89
Table 13: FC Pin Functions . . . 90
Table 14: FCCMD Bit Functions . . . 91
Table 15: SRAM_TYPE Field Encoding . . . 94
Table 16: LED Default Configuration . . . 101
Table 17: IEEE 1149.1 Supported Instruction Summary . . . 106
Table 18: BSR Mode Of Operation . . . 106
Table 19: Device ID Register . . . 106
Table 20: PCI Configuration Space Layout . . . 108
Table 21: Address PROM Space Contents . . . 109
Table 22: I/O Map In Word I/O Mode (DWIO = 0) . . . 110
Table 23: Legal I/O Accesses in Word I/O Mode (DWIO = 0) . . . 111
Table 24: I/O Map In DWord I/O Mode (DWIO = 1) . . . 111
Table 25: Legal I/O Accesses in Double Word I/O Mode (DWIO =1) . . . 111
Table 26: AP_VALUE0: Auto-Poll Value0 Register . . . 124
Table 27: AP_VALUE1: Auto-Poll Value1 Register . . . 125
Table 28: AP_VALUE2: Auto-Poll Value2 Register . . . 125
Table 29: AP_VALUE3: Auto-Poll Value3 Register . . . 125
Table 30: AP_VALUE4: Auto-Poll Value4 Register . . . 125
Table 31: AP_VALUE5: Auto-Poll Value5 Register . . . 125
Table 32: AUTOPOLL0: Auto-Poll0 Register . . . 126
Table 33: AUTOPOLL1: Auto-Poll1 Register . . . 126
Table 34: AUTOPOLL2: Auto-Poll2 Register . . . 127
Table 35: AUTOPOLL3: Auto-Poll3 Register . . . 128
Table 36: AUTOPOLL4: Auto-Poll4 Register . . . 129
Table 37: AUTOPOLL5: Auto-Poll5 Register . . . 130
Table 38: Receive Ring Base Address Register . . . 130
Table 39: Transmit Ring Base Address Register . . . 131
Table 40: CHIPID: Chip ID Register . . . 131
Table 41: CHPOLLTIME: Chain Polling Interval Register . . . 131
Table 42: CMD0: Command0 Register . . . 132
Table 43: CMD2: Command2 Register . . . 133
Table 44: CMD3: Command3 Register . . . 137
Table 45: CMD7: Command7 Register . . . 140
Table 46: CTRL0: Control0 Register . . . 141
Table 47: CTRL1: Control1 Register . . . 142
Table 48: CTRL2: Control2 Register . . . 145

Downloaded from Arrow.com.

Table 49: CTRL3: Control3 Register . . . 146
Table 50: Software Styles . . . 147
Table 51: DATAMBIST: Memory Built-in Self-Test Access Register . . . 147
Table 52: Reserved Register . . . 149
Table 53: EEPROM_ACC: EEPROM Access Register . . . 150
Table 54: Interface Pin Assignment . . . 151
Table 55: FLASH_ADDR: Flash Address Register . . . 152
Table 56: FLASH_DATA: Flash Data Register . . . 152
Table 57: FLOW: Flow Control Register . . . 153
Table 58: IFS1: Inter-Frame Spacing Part 1 Register . . . 154
Table 59: INT0: Interrupt0 Register . . . 155
Table 60: INTEN0: Interrupt0 Enable Register . . . 157
Table 61: IPG: Inter-Packet Gap Register . . . 159
Table 62: Logical Address Filter Register . . . 159
Table 63: LED0 Control Register . . . 160
Table 64: MAX_LAT_A: PCI Maximum Latency Alias Register . . . 162
Table 65: MIN_GNT_A: PCI Minimum Grant Alias Register . . . 162
Table 66: PADR: Physical Address Register . . . 163
Table 67: PAUSE_CNT: Pause Count Register . . . 163
Table 68: PCIDATA0: PCI DATA Register Zero Alias Register . . . 163
Table 69: PHY_ACCESS: PHY Access Register . . . 165
Table 70: PMAT0: OnNow Pattern Register 0 . . . 166
Table 71: PMAT1: OnNow Pattern Register 1 . . . 166
Table 72: Receive Protect Register . . . 167
Table 73: RCV_RING_LEN: Receive Ring Length Register . . . 167
Table 74: ROM_CFG: ROM Base Address Configuration Register . . . 168
Table 75: SID_A: PCI Subsystem ID Alias Register . . . 168
Table 76: SRAM Boundary Register . . . 169
Table 77: SRAM Size Register . . . 169
Table 78: STAT0: Status0 Register . . . 170
Table 79: Software Timer Value Register . . . 171
Table 80: SVID: PCI Subsystem Vendor ID Shadow Register . . . 172
Table 81: TEST0: Test Register . . . 172
Table 82: VID_A: PCI Vendor ID Alias Register . . . 173
Table 83: XMT_RING_LEN: Transmit Ring Length Register . . . 173
Table 84: XMTPOLLTIME: Transmit Polling Interval Register . . . 174
Table 85: Software Styles . . . 192
Table 86: Receive Watermark Programming . . . 194
Table 87: Transmit Start Point Programming . . . 194
Table 88: Transmit Watermark Programming . . . 194
Table 89: BCR Registers . . . 199
Table 90: EEDET Setting . . . 213
Table 91: Interface Pin Assignment . . . 213
Table 92: Software Styles . . . 216
Table 93: SRAM_BND Programming . . . 218
Table 94: FMDC Values . . . 220
Table 95: APDW Values . . . 220
Table 96: Initialization Block (SSIZE32 = 0) . . . 227
Table 97: Initialization Block (SSIZE32 = 1) . . . 227

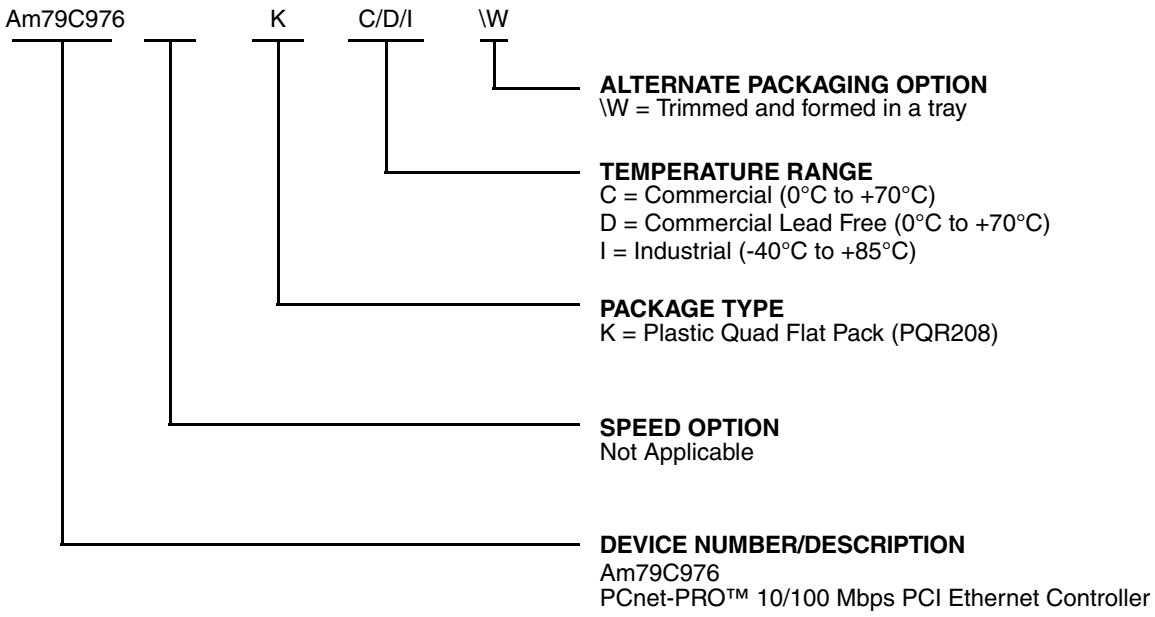
Downloaded from Arrow.com.

Table 98: R/TLEN Decoding (SSIZE32 = 0) 228
Table 99: R/TLEN Decoding (SSIZE32 = 1) 228
Table 100: Receive Descriptor (SWSTYLE = 0) 229
Table 101: Receive Descriptor (SWSTYLE = 2) 229
Table 102: Receive Descriptor (SWSTYLE = 3) 230
Table 103: Receive Descriptor (SWSTYLE = 4) 230
Table 104: Receive Descriptor (SWSTYLE = 5) 230
Table 105: Receive Descriptor, SWSTYLE = 0 231
Table 106: Receive Descriptor, SWSTYLE = 2 232
Table 107: Receive Descriptor, SWSTYLE = 3 234
Table 108: Receive Descriptor, SWSTYLE = 4 235
Table 109: Receive Descriptor, SWSTYLE = 5 237
Table 110: Transmit Descriptor (SWSTYLE = 0) 240
Table 111: Transmit Descriptor (SWSTYLE = 2) 240
Table 112: Transmit Descriptor (SWSTYLE = 3) 240
Table 113: Transmit Descriptor (SWSTYLE = 4) 240
Table 114: Transmit Descriptor (SWSTYLE = 5) 241
Table 115: Transmit Descriptor, SWSTYLE = 0 241
Table 116: Transmit Descriptor, SWSTYLE = 2 242
Table 117: Transmit Descriptor, SWSTYLE = 3 243
Table 118: Transmit Descriptor, SWSTYLE = 4 244
Table 119: Transmit Descriptor, SWSTYLE = 5 245
Table 120: Register Bit Cross Reference 255
Table 121: Control and Status Registers 264
Table 122: Bus Configuration Registers 266

Downloaded from Arrow.com.

ORDERING INFORMATION
Standard Products

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



Valid Combinations

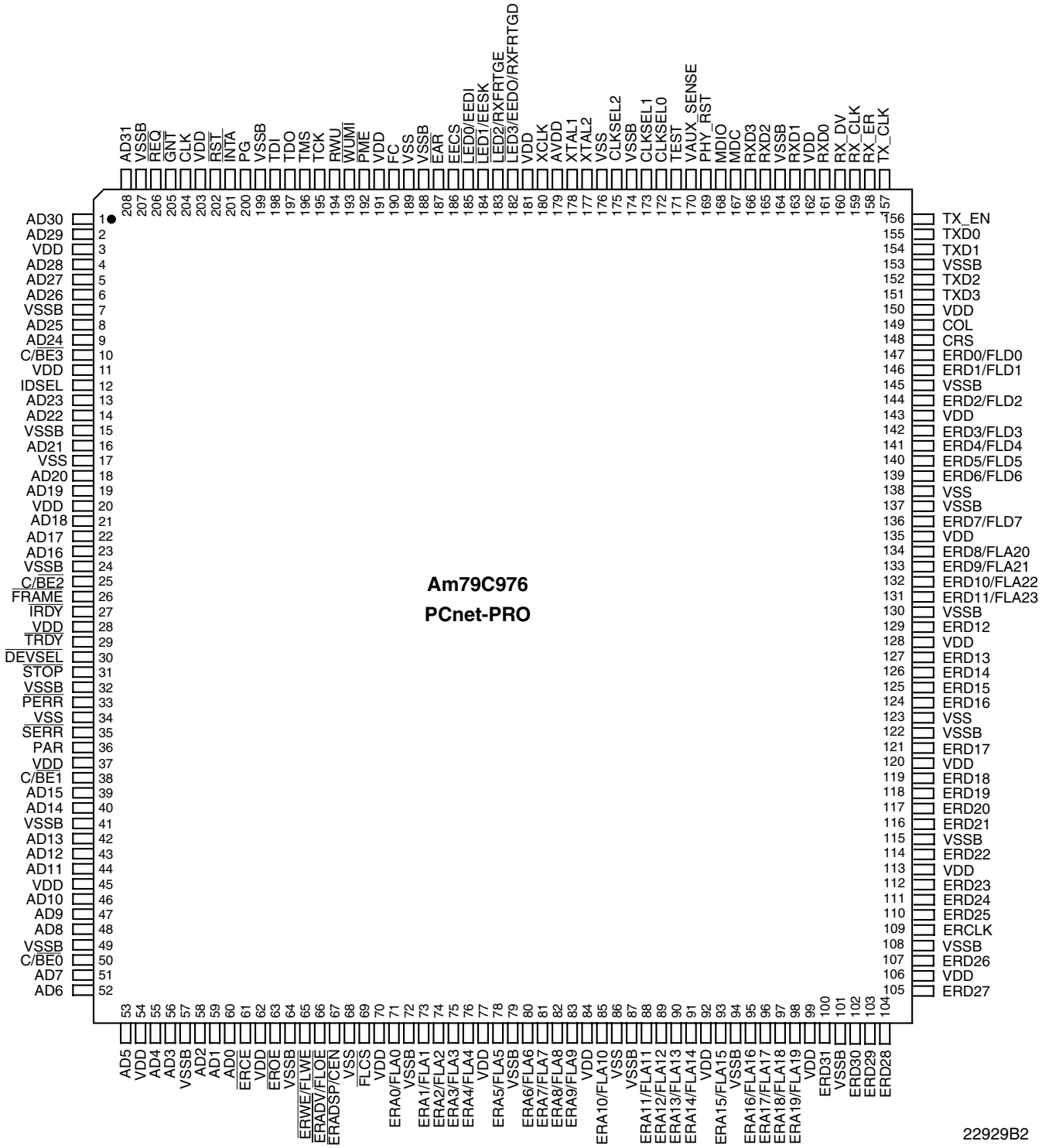
Am79C976	KC/W KD/W KI/W

Valid Combinations

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

Downloaded from Arrow.com.

CONNECTION DIAGRAM (PQR208)



Downloaded from Arrow.com.

PIN DESIGNATIONS (PQR208)

Listed By Pin Number

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	AD30	53	AD5	105	ERD27	157	TX_CLK
2	AD29	54	VDD	106	VDD	158	RX_ER
3	VDD	55	AD4	107	ERD26	159	RX_CLK
4	AD28	56	AD3	108	VSSB	160	RX_DV
5	AD27	57	VSSB	109	ERCLK	161	RXD0
6	AD26	58	AD2	110	ERD25	162	VDD
7	VSSB	59	AD1	111	ERD24	163	RXD1
8	AD25	60	AD0	112	ERD23	164	VSSB
9	AD24	61	$\overline{\text{ERCE}}$	113	VDD	165	RXD2
10	C/$\overline{\text{BE3}}$	62	VDD	114	ERD22	166	RXD3
11	VDD	63	$\overline{\text{EROE}}$	115	VSSB	167	MDC
12	IDSEL	64	VSSB	116	ERD21	168	MDIO
13	AD23	65	$\overline{\text{ERWE}}$/$\overline{\text{FLWE}}$	117	ERD20	169	PHY_RST
14	AD22	66	$\overline{\text{ERADV}}$/$\overline{\text{FLOE}}$	118	ERD19	170	VAUX_SENSE
15	VSSB	67	$\overline{\text{ERADSP}}$/$\overline{\text{CEN}}$	119	ERD18	171	TEST
16	AD21	68	VSS	120	VDD	172	CLKSEL0
17	VSS	69	$\overline{\text{FLCS}}$	121	ERD17	173	CLKSEL1
18	AD20	70	VDD	122	VSSB	174	VSSB
19	AD19	71	ERA0/FLA0	123	VSS	175	CLKSEL2
20	VDD	72	VSSB	124	ERD16	176	VSS
21	AD18	73	ERA1/FLA1	125	ERD15	177	XTAL2
22	AD17	74	ERA2/FLA2	126	ERD14	178	XTAL1
23	AD16	75	ERA3/FLA3	127	ERD13	179	AVDD
24	VSSB	76	ERA4/FLA4	128	VDD	180	XCLK
25	C/$\overline{\text{BE2}}$	77	VDD	129	ERD12	181	VDD
26	$\overline{\text{FRAME}}$	78	ERA5/FLA5	130	VSSB	182	$\overline{\text{LED3}}$/EEDO/ RXFRTGD
27	$\overline{\text{IRDY}}$	79	VSSB	131	ERD11/FLA23	183	$\overline{\text{LED2}}$/RXFRTGE
28	VDD	80	ERA6/FLA6	132	ERD10/FLA22	184	$\overline{\text{LED1}}$/EESK
29	$\overline{\text{TRDY}}$	81	ERA7/FLA7	133	ERD9/FLA21	185	$\overline{\text{LED0}}$/EEDI
30	$\overline{\text{DEVSEL}}$	82	ERA8/FLA8	134	ERD8/FLA20	186	EECS
31	$\overline{\text{STOP}}$	83	ERA9/FLA9	135	VDD	187	$\overline{\text{EAR}}$
32	VSSB	84	VDD	136	ERD7/FLD7	188	VSSB
33	$\overline{\text{PERR}}$	85	ERA10/FLA10	137	VSSB	189	VSS
34	VSS	86	VSS	138	VSS	190	FC

Downloaded from Arrow.com.

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
35	$\overline{\text{SERR}}$	87	VSSB	139	ERD6/FLD6	191	VDD
36	PAR	88	ERA11/FLA11	140	ERD5/FLD5	192	$\overline{\text{PME}}$
37	VDD	89	ERA12/FLA12	141	ERD4/FLD4	193	$\overline{\text{WUMI}}$
38	C/$\overline{\text{BE1}}$	90	ERA13/FLA13	142	ERD3/FLD3	194	RWU
39	AD15	91	ERA14/FLA14	143	VDD	195	TCK
40	AD14	92	VDD	144	ERD2/FLD2	196	TMS
41	VSSB	93	ERA15/FLA15	145	VSSB	197	TDO
42	AD13	94	VSSB	146	ERD1/FLD1	198	TDI
43	AD12	95	ERA16/FLA16	147	ERD0/FLD0	199	VSSB
44	AD11	96	ERA17/FLA17	148	CRS	200	PG
45	VDD	97	ERA18/FLA18	149	COL	201	$\overline{\text{INTA}}$
46	AD10	98	ERA19/FLA19	150	VDD	202	$\overline{\text{RST}}$
47	AD9	99	VDD	151	TXD3	203	VDD
48	AD8	100	ERD31	152	TXD2	204	CLK
49	VSSB	101	VSSB	153	VSSB	205	$\overline{\text{GNT}}$
50	C/$\overline{\text{BE0}}$	102	ERD30	154	TXD1	206	$\overline{\text{REQ}}$
51	AD7	103	ERD29	155	TXD0	207	VSSB
52	AD6	104	ERD28	156	TX_EN	208	AD31

Downloaded from Arrow.com.

PIN DESIGNATIONS

Listed By Group

Pin Name	Pin Function	Signal Type[1]	Pin Type[1]	No. of Pins
Clock Interface				
XTAL1	Crystal	I	I	1
XTAL2	Crystal	O	O	1
XCLK	External Clock	I	I	1
CLKSEL0	Clock Select	I	I	1
CLKSEL1	Clock Select	I	I	1
CLKSEL2	Clock Select	I	I	1
TEST	Test Select	I	I	1
PCI Bus Interface				
AD[31:0]	Address/Data Bus	IO	IO	32
C/$\overline{BE}$[3:0]	Bus Command/Byte Enable	IO	IO	4
CLK	Bus Clock	I	I	1
$\overline{DEVSEL}$	Device Select	IO	IO	1
$\overline{FRAME}$	Cycle Frame	IO	IO	1
$\overline{GNT}$	Bus Grant	I	I	1
IDSEL	Initialization Device Select	I	I	1
$\overline{INTA}$	Interrupt	O	TSO	1
$\overline{IRDY}$	Initiator Ready	IO	IO	1
PAR	Parity	IO	IO	1
$\overline{PERR}$	Parity Error	IO	IO	1
$\overline{REQ}$	Bus Request	O	TSO	1
$\overline{RST}$	Reset	I	I	1
$\overline{SERR}$	System Error	IO	IO	1
$\overline{STOP}$	Stop	IO	IO	1
$\overline{TRDY}$	Target Ready	IO	IO	1
Board Interface				
$\overline{LED0}$	LED0	O	TSO	1
$\overline{LED1}$	LED1	O	TSO	1
$\overline{LED2}$	LED2	O	IO	1
$\overline{LED3}$	LED3	O	IO	1
PHY_RST	Reset to PHY	O	O	1
FC	Flow Control	I	I	1
EEPROM Interface				

Downloaded from Arrow.com.

Pin Name	Pin Function	Signal Type[1]	Pin Type[1]	No. of Pins
EECS	Serial EEPROM Chip Select	O	O	1
EEDI	Serial EEPROM Data In	O	TSO	1
EEDO	Serial EEPROM Data Out	I	IO	1
EESK	Serial EEPROM Clock	IO	TSO	1
External Memory Interface				
ERCLK	External Memory Clock	O	O	1
ERA[19:0]/FLA[19:0]	External Memory Address[19:0]	O	O	20
ERD[31:0] / FLA[23:20] / FLD[7:0]	External Memory Data [31:0]/Flash Address[23:20]/Flash Data[7:0]	IO	IO	32
$\overline{\text{ERADV}}$/$\overline{\text{FLOE}}$	External Memory Advance	O	O	1
$\overline{\text{ERADSP}}$/$\overline{\text{CEN}}$	External Memory Address Strobe	O	O	1
$\overline{\text{EROE}}$	External Memory Output Enable	O	O	1
$\overline{\text{ERWE}}$/$\overline{\text{FLWE}}$	External Memory Write Enable	O	O	1
$\overline{\text{ERCE}}$	SSRAM Chip Enable	O	O	1
$\overline{\text{FLCS}}$	Flash Memory Chip Select	O	O	1
Media Independent Interface (MII)				
COL	Collision	I	I	1
CRS	Carrier Sense	I	I	1
FC	Hardware Flow Control	I	I	1
MDC	Management Data Clock	O	O	1
MDIO	Management Data I/O	IO	IO	1
RX_CLK	Receive Clock	I	I	1
RXD[3:0]	Receive Data	I	I	4
RX_DV	Receive Data Valid	I	I	1
RX_ER	Receive Error	I	I	1
TX_CLK	Transmit Clock	I	I	1
TXD[3:0]	Transmit Data	O	O	4
TX_EN	Transmit Data Enable	O	O	1
External Address Detection Interface (EADI)				
$\overline{\text{EAR}}$	External Address Reject	I	I	1
SFBD	Start Frame Byte Delimiter	Note 2	Note 2	1
RXFRTGD	Receive Frame Tag Data	I	IO	1
RXFRTGE	Receive Frame Tag Enable	I	IO	1
Power Management Interface				
RWU	Remote Wake Up	O	TSO	1
$\overline{\text{PME}}$	Power Management Event	O	OD	1
$\overline{\text{WUMI}}$	Wake-Up Mode Indicator	O	OD	1
PG	Power Good	I	I	1

Downloaded from Arrow.com.

Pin Name	Pin Function	Signal Type[1]	Pin Type[1]	No. of Pins
VAUX_SENSE	Vaux Sense	I	I	1
IEEE 1149.1 Test Access Port Interface (JTAG)				
TCK	Test Clock	I	I	1
TDI	Test Data In	I	I	1
TDO	Test Data Out	O	O	1
TMS	Test Mode Select	I	I	1
Power Supplies				
VDD	Digital and I/O Buffer Power	P	P	24
VSS	Digital Ground	P	P	8
AVDD	Analog VDD for PLL and OSC	P	P	1
VSSB	I/O Buffer Ground	P	P	25

Notes:

1. *Since some pins provide more than one signal, the pin type for a signal may differ from the signal type.*
2. *The SFBD signal can be programmed to appear on any of the LED pins.*

Table Legend:

Name	Pin Type
IO	Input/Output
I	Input
O	Output
TSO	Three-State Output
OD	Open Drain

Downloaded from Arrow.com.

PIN DESCRIPTIONS

PCI Interface

AD[31:0]

Address and Data **Input/Output**

Address and data are multiplexed on the same bus interface pins. During the first clock of a transaction, AD[31:0] contain a physical address (32 bits). During the subsequent clocks, AD[31:0] contain data. Byte ordering is Little Endian by default. AD[7:0] are defined as the least significant byte (LSB) and AD[31:24] are defined as the most significant byte (MSB). For FIFO data transfers, the Am79C976 controller can be programmed for Big Endian byte ordering. See Control 0 Register, bit 24 (BSWP) for more details.

During the address phase of the transaction, when the Am79C976 controller is a bus master, AD[31:2] will address the active Double Word (DWord). The Am79C976 controller always drives AD[1:0] to "00" during the address phase indicating linear burst order. When the Am79C976 controller is not a bus master, the AD[31:0] lines are continuously monitored to determine if an address match exists for slave transfers.

During the data phase of the transaction, AD[31:0] are driven by the Am79C976 controller when performing bus master write and slave read operations. Data on AD[31:0] is latched by the Am79C976 controller when performing bus master read and slave write operations.

The Am79C976 device supports Dual Address Cycles (DAC) for systems with 64-bit addressing. As a bus master the Am79C976 device will generate addresses of up to 40 bits in length. If the value of the C/$\overline{BE}$[3:0] bus during the PCI address phase is 1101b, the address phase is extended to two clock cycles. The low order address bits appear on the AD[31:0] bus during the first clock cycle, and the high order bits appear during the second clock cycle. In dual address cycles the PCI bus command (memory read, I/O write, etc.) appears on the C/$\overline{BE}$ pins during the second clock cycle.

C/$\overline{BE}$[3:0]

Bus Command and Byte Enables **Input/Output**

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

CLK

Clock **Input**

This clock is used to drive the system bus interface. All bus signals are sampled on the rising edge of CLK and all parameters are defined with respect to this edge. The Am79C976 controller normally operates over a frequency range of 15 MHz to 33 MHz on the PCI bus due to networking demands. The Am79C976 controller will support a clock frequency of 0 MHz after certain precautions are taken to ensure data integrity. This clock or a derivation is not used to drive any network functions.

$\overline{DEVSEL}$

Device Select **Input/Output**

The Am79C976 controller drives $\overline{DEVSEL}$ when it detects a transaction that selects the device as a target. The device samples $\overline{DEVSEL}$ to detect if a target claims a transaction that the Am79C976 controller has initiated.

$\overline{FRAME}$

Cycle Frame **Input/Output**

$\overline{FRAME}$ is driven by the Am79C976 controller when it is the bus master to indicate the beginning and duration of a transaction. $\overline{FRAME}$ is asserted to indicate a bus transaction is beginning. $\overline{FRAME}$ is asserted while data transfers continue. $\overline{FRAME}$ is deasserted before the final data phase of a transaction. When the Am79C976 controller is in slave mode, it samples $\overline{FRAME}$ to determine the address phase of a transaction.

$\overline{GNT}$

Bus Grant **Input**

This signal indicates that the access to the bus has been granted to the Am79C976 controller.

The Am79C976 controller supports bus parking. When the PCI bus is idle and the system arbiter asserts $\overline{GNT}$ without an active $\overline{REQ}$ from the Am79C976 controller, the device will drive the AD[31:0], C/$\overline{BE}$[3:0], and PAR lines.

IDSEL

Initialization Device Select **Input**

This signal is used as a chip select for the Am79C976 controller during configuration read and write transactions.

$\overline{INTA}$

Interrupt Request **Output**

An attention signal which indicates that one or more enabled interrupt flag bits are set. See the descriptions of the INT and INTEN registers for details.

Downloaded from Arrow.com.

By default INTA is an open-drain output. For applications that need an active-high edge-sensitive interrupt signal, the INTA pin can be configured for this mode by setting INTLEVEL (CMD3, bit 13 or BCR2, bit 7) to 1.

IRDY

Initiator Ready **Input/Output**

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

When the Am79C976 controller is a bus master, it asserts IRDY during all write data phases to indicate that valid data is present on AD[31:0]. During all read data phases, the device asserts IRDY to indicate that it is ready to accept the data.

When the Am79C976 controller is the target of a transaction, it checks IRDY during all write data phases to determine if valid data is present on AD[31:0]. During all read data phases, the device checks IRDY to determine if the initiator is ready to accept the data.

PAR

Parity **Input/Output**

Parity is even parity across AD[31:0] and C/BE[3:0]. When the Am79C976 controller is a bus master, it generates parity during the address and write data phases. It checks parity during read data phases. When the Am79C976 controller operates in slave mode, it checks parity during every address phase. When it is the target of a cycle, it checks parity during write data phases and it generates parity during read data phases.

PERR

Parity Error **Input/Output**

During any slave write transaction and any master read transaction, the Am79C976 controller asserts PERR when it detects a data parity error and reporting of the error is enabled by setting PERREN (PCI Command register, bit 6) to 1. During any master write transaction, the Am79C976 controller monitors PERR to see if the target reports a data parity error.

REQ

Bus Request **Input/Output**

The Am79C976 controller asserts REQ pin as a signal that it wishes to become a bus master. REQ is driven high when the Am79C976 controller does not request the bus.

RST

Reset **Input**

When RST is asserted LOW and the PG pin is HIGH, then the Am79C976 controller performs an internal system reset of the type H_RESET (HARDWARE_RESET, see section on RESET). Immediately after the initial power up, RST must be held low for 26µs. At any other time RST must be held low for a minimum of 30 clock periods to guarantee that the device is properly reset. While in the H_RESET state, the Am79C976 controller will disable or deassert all outputs. RST may be asynchronous to clock when asserted or deasserted.

Asserting RST disables all of the PCI pins except the PME pin.

SERR

System Error **Output**

During any slave transaction, the Am79C976 controller asserts SERR when it detects an address parity error, and reporting of the error is enabled by setting PERREN (PCI Command register, bit 6) and SERREN (PCI Command register, bit 8) to 1.

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

STOP

Stop **Input/Output**

In slave mode, the Am79C976 controller drives the STOP signal to inform the bus master to stop the current transaction. In bus master mode, the Am79C976 controller checks STOP to determine if the target wants to disconnect the current transaction.

TRDY

Target Ready **Input/Output**

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

When the Am79C976 controller is a bus master, it checks TRDY during all read data phases to determine if valid data is present on AD[31:0]. During all write data phases, the device checks TRDY to determine if the target is ready to accept the data.

When the Am79C976 controller is the target of a transaction, it asserts TRDY during all read data phases to indicate that valid data is present on AD[31:0]. During all write data phases, the device asserts TRDY to indicate that it is ready to accept the data.

Downloaded from Arrow.com.

PME

Power Management Event **Output, Open Drain**

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

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

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

Board Interface

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

LED0

LED0 **Output**

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

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

LED1

LED1 **Output**

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

Note: *The LED1 pin is multiplexed with the EESK pin.*

LED2

LED2 **Output**

This output is designed to directly drive an LED. By default, LED2 indicates that the network bit rate is 100 Mb/s. This pin can also be programmed to indicate various network status (see BCR6). The LED2 pin polarity is programmable, but by default it is active LOW. When the LED2 pin polarity is programmed to active LOW, the
output is an open drain driver. When the LED2 pin polarity is programmed to active HIGH, the output is a totem pole driver.

Note: *The LED2 pin is multiplexed with the RXFRTGE pin.*

LED3

LED3 **Output**

This output is designed to directly drive an LED. By default, LED3 indicates that a collision has occurred. This pin can also be programmed to indicate other network status (see BCR7). The LED3 pin polarity is programmable, but by default it is active LOW. When the LED3 pin polarity is programmed to active LOW, the output is an open drain driver. When the LED3 pin polarity is programmed to active HIGH, the output is a totem pole driver.

Special attention must be given to the external circuitry attached to this pin. When this pin is used to drive an LED while an EEPROM is used in the system, then buffering may be required between the LED3 pin and the LED circuit. If an LED circuit were directly attached to this pin, it may create an IOL requirement that could not be met by the serial EEPROM attached to this pin. If no EEPROM is included in the system design or low current LEDs are used, then the LED3 signal may be directly connected to an LED without buffering. In any case, if an EEPROM is present, there must be a pull-up resistor connected to this pin (10 kΩ should be adequate). For more details regarding LED connection, see the section on *LED Support*.

Note: *The LED3 pin is multiplexed with the EEDO and RXFRTGD pins.*

PG

Power Good **Input**

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

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

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

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

RWU

Remote Wake Up **Output**

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

Downloaded from Arrow.com.

Change Detect mode and a Link Change has been detected.

This pin can drive the external system management logic that causes the CPU to get out of a low power mode of operation. This pin is implemented for designs that do not support the PME function.

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

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

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

WUMI

Wake-Up Mode Indicator **Output, Open Drain**

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

VAUX_SENSE

3.3 Vaux Presence Sense **Input**

The signal on this pin is logically anded with bit 15 of the PCI PMC register when the PMC register is read. This pin should normally be connected to the PCI 3.3 Vaux pin. This allows the PMC register to indicate that the device is capable of supporting PME from the $D3_{cold}$ state only when the 3.3 Vaux pin is supplying power.

CLKSEL0

Clock Select 0 **Input**

The Am79C976 system clock can either be driven by an external clock generator connected to the XCLK pin or by an internal clock generator timed by a 25-MHz crystal connected between the XTAL1 and XTAL2 pins. The CLKSEL0 and CLKSEL1 pins select the source of the system clock and the frequency at which the external clock generator must run. In addition, CLKSEL0 and CLKSEL1 determine the frequency of ERCLK, the external SSRAM clock. Table 1 shows the possible combinations.

CLKSEL1

Clock Select 1 **Input**

The Am79C976 system clock can either be driven by an external clock generator connected to the XCLK pin or by an internal clock generator timed by a 25-MHz crystal connected between the XTAL1 and XTAL2 pins. The CLKSEL0 and CLKSEL1 pins select the source of the system clock and the frequency at which the external clock generator must run. In addition CLKSEL0 and CLKSEL1 determine the frequency of ERCLK, the external SSRAM clock. Table 1 shows the possible combinations.

CLKSEL2

Clock Select 2 **Input**

The CLKSEL2 pin must be held low during normal operation.

TEST

Test Reset **Input**

The TEST pin must be held low during normal operation.

XCLK

External Clock Input **Input**

The Am79C976 system clock can either be driven by an external clock generator connected to this pin or by a 25-MHz crystal connected between the XTAL1 and XTAL2 pins, depending on the state of the CLKSEL0 and CLKSEL1 pins. When either CLKSEL0 or CLKSEL1 or both are held high, a 20-, 25-, or $33^1/_3$-MHz clock signal must be applied to XCLK as shown in Table 1. When CLKSEL0 and CLKSEL1 are both held low, the XCLK pin should be connected to either VSS or VDD.

Table 1. System Clock Selections

CLKSEL2	CLKSEL1	CLKSEL0	CLOCK SOURCE	ERCLK (MHz)
1	X	X	Design Factory Test Only.	
0	0	0	25-MHz Crystal, XTAL1,XTAL2	87.5
0	0	1	XCLK, 20 MHz	90
0	1	0	XCLK, 25 MHz	87.5
0	1	1	XCLK, $33^1/_3$ MHz	82.5

XTAL1

Crystal **Input**

If the CLKSEL0 and CLKSEL1 pins are both held low, a 25-MHz crystal should be connected between the XTAL1 pin and the XTAL2 pin. This crystal controls the frequency of the internal clock-generator circuit.

Downloaded from Arrow.com.

If the CLKSEL0 and CLKSEL1 pins are not both held low, a 20-, 25-, or 33 1/3-MHz clock source must be connected to the XCLK pin, and the XTAL1 and XTAL2 pins should be connected to VSS.

XTAL1 and XTAL2 are not 5-volt tolerant pins.

XTAL2

Crystal **Output**

If the CLKSEL0 and CLKSEL1 pins are both held low, a 25 MHz crystal should be connected between the XTAL1 pin and the XTAL2 pin. This crystal controls the frequency of the internal clock generator circuit.

If either the CLKSEL0 or the CLKSEL1 pin or both are held high, a 20-, 25-, or 33 1/3-MHz clock source must be connected to the XCLK pin, and the XTAL1 and XTAL2 pins should be connected to VSS.

XTAL1 and XTAL2 are not 5-volt tolerant pins.

PHY_RST

PHY Reset **Output**

PHY_RST is an output pin that is used to reset the external PHY. This output eliminates the need for a fan-out buffer for the PCI RST signal, provides polarity for the specific PHY used, and prevents the resetting of the PHY when the PG input is LOW. The output polarity is determined by the RST_POL bit (CMD3, bit0), which can be loaded from the EEPROM.

The length of time for which the PHY_RST pin is asserted depends on the number of registers that are loaded from the EEPROM and the order in which the registers are loaded. Immediately after the RST_POL bit is loaded from the EEPROM, the PHY_RST pin is asserted. When the last register has been loaded from the EEPROM, the PHY_RST pin is deasserted. Each register loaded after the RST_POL bit is loaded adds about 240 µs to the time that PHY_RST is asserted. If the PHY_RST pin is used to reset an external PHY, the user should program the EEPROM to make sure that PHY_RST is asserted long enough to meet the requirements of the PHY. The user can insert dummy writes to offset 28h to extend the reset period.

FC

Flow Control **Input**

The Flow Control input signal controls when MAC Control Pause Frames are sent or when half-duplex back pressure is asserted.

EEPROM Interface

EECS

EEPROM Chip Select **Output**

This pin is designed to directly interface to a serial EEPROM that uses the 93Cxx EEPROM interface protocol. EECS is connected to the EEPROM's chip select pin. It is controlled by either the Am79C976 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 2.

EEDI

EEPROM Data In **Output**

This pin is designed to directly interface to a serial EEPROM that uses the 93Cxx EEPROM interface protocol. EEDI is connected to the EEPROM's data input pin. It is controlled by either the Am79C976 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 0.

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

EEDO

EEPROM Data Out **Input**

This pin is designed to directly interface to a serial EEPROM that uses the 93Cxx EEPROM interface protocol. EEDO is connected to the EEPROM's data output pin. It is controlled by either the Am79C976 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by reading from BCR19, bit 0.

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

EESK

EEPROM Serial Clock **Output**

This pin is designed to directly interface to a serial EEPROM that uses the 93Cxx EEPROM interface protocol. EESK is connected to the EEPROM's clock pin. It is controlled by either the Am79C976 controller directly during a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 1.

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

External Memory Interface

ERA[19:0]/FLA[19:0]

External Memory Address [19:0] **Output**

The ERA[19:0] pins provide addresses for both the external SSRAM and the external boot ROM device.

All ERA[19:0] pin outputs are forced to a constant level to conserve power while no access on the External Memory Bus is being performed.

FLA[23:20]

Boot ROM (Flash) Address [23:20] **Output**

The FLA[23:20] pins provide the 4 most significant bits of the address for the external boot ROM device.

Downloaded from Arrow.com.

All FLA[23:20] pin outputs are forced to a constant level to conserve power while no access on the External Memory Bus is being performed.

Note: *The FLA[23:20] pins are multiplexed with the ERD[11:8] pins.*

ERD[31:0]/FLD[7:0]

External Memory Data [31:0] **Input/Output**

The ERD[7:0] pins provide data bits [7:0] for boot ROM accesses. The ERD[31:0] pins provide data bits [31:0] for external SSRAM accesses. The ERD[31:0] signals are forced to a constant level to conserve power while no access on the External Memory Bus is being performed.

Note: *The FLA[23:20] pins are multiplexed with the ERD[11:8] pins.*

$\overline{\text{ERCE}}$

External SSRAM Chip Enable **Output**

$\overline{\text{ERCE}}$ serves as the chip enable for the external SSRAM. It is asserted low when the SSRAM address on the ERA[19:0] pins is valid.

$\overline{\text{FLCS}}$

Boot ROM Chip Select **Output**

$\overline{\text{FLCS}}$ serves as the chip select for the boot device. It is asserted low when the boot ROM address on the FLA[23:20] and ERA[19:0] pins is valid.

$\overline{\text{EROE}}$

External SSRAM Output Enable **Output**

$\overline{\text{EROE}}$ is asserted active LOW during SSRAM device read operations to allow the SSRAM device to drive the ERD[31:0] data bus. It is deasserted at all other times.

$\overline{\text{FLOE}}$

Expansion ROM Output Enable **Output**

$\overline{\text{FLOE}}$ is asserted active LOW during boot ROM read operations to allow the boot ROM to drive the ERD[7:0] data bus. It is deasserted at all other times.

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

$\overline{\text{ERWE}}$/$\overline{\text{FLWE}}$

External Memory Write Enable **Output**

$\overline{\text{ERWE}}$ provides the write enable for write accesses to the external SSRAM and the Flash (boot ROM) device.

$\overline{\text{ERADSP}}$/$\overline{\text{CEN}}$

External Memory Address Strobe **Output**

$\overline{\text{ERADSP}}$ provides the address strobe signal to load the address into the external SSRAM.

$\overline{\text{ERADV}}$

External Memory Address Advance **Output**

$\overline{\text{ERADV}}$ provides the address advance signal to the external SSRAM. This signal is asserted low during a burst access to increment the address counter in the SSRAM.

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

ERCLK

External Memory Clock **Output**

ERCLK is the reference clock for all synchronous SRAM accesses.

Media Independent Interface

TX_CLK

Transmit Clock **Input**

TX_CLK is a continuous clock input that provides the timing reference for the transfer of the TX_EN and TXD[3:0] signals out of the Am79C976 device. TX_CLK must provide a nibble rate clock (25% of the network data rate). Hence, an MII transceiver operating at 10 Mbps must provide a TX_CLK frequency of 2.5 MHz and an MII transceiver operating at 100 Mbps must provide a TX_CLK frequency of 25 MHz.

TXD[3:0]

Transmit Data **Output**

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

TX_EN

Transmit Enable **Output**

TX_EN indicates when the Am79C976 device is presenting valid transmit nibbles on the MII. While TX_EN is asserted, the Am79C976 device generates TXD[3:0] on TX_CLK rising edges. TX_EN is asserted with the first nibble of preamble and remains asserted throughout the duration of a packet until it is deasserted prior to the first TX_CLK following the final nibble of the frame. TX_EN transitions synchronous to TX_CLK rising edges.

COL

Collision **Input**

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

Downloaded from Arrow.com.

CRS

Carrier Sense **Input**

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

RX_CLK

Receive Clock **Input**

RX_CLK is a clock input that provides the timing reference for the transfer of the RX_DV, RXD[3:0], and RX_ER signals into the Am79C976 device. RX_CLK must provide a nibble rate clock (25% of the network data rate). Hence, an MII transceiver operating at 10 Mbps must provide an RX_CLK frequency of 2.5 MHz and an MII transceiver operating at 100 Mbps must provide an RX_CLK frequency of 25 MHz. When the external PHY switches the RX_CLK and TX_CLK, it *must* provide glitch-free clock pulses.

RXD[3:0]

Receive Data **Input**

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

RX_DV

Receive Data Valid **Input**

RX_DV is an input used to indicate that valid, received data is being presented on the RXD[3:0] pins and RX_CLK is synchronous to the receive data. In order for a frame to be fully received by the Am79C976 device on the MII, RX_DV must be asserted prior to the RX_CLK rising edge, when the first nibble of the Start-of-Frame Delimiter is driven on RXD[3:0], and must remain asserted until after the rising edge of RX_CLK, when the last nibble of the CRC is driven on RXD[3:0]. RX_DV must then be deasserted prior to the RX_CLK rising edge which follows this final nibble. RX_DV transitions are synchronous to RX_CLK rising edges.

RX_ER

Receive Error **Input**

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

MDC

Management Data Clock **Output**

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

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

MDIO

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

MDIO is the bidirectional MII management port data pin. MDIO is an output during the header portion of the management frame transfers and during the data portions of write transfers. MDIO is an input during the data portions of read data transfers. When an operation is not in progress on the management port, MDIO is not driven. MDIO transitions from the Am79C976 controller are synchronous to MDC falling edges.

If the PHY is attached through an MII physical connector, then the MDIO pin should be externally pulled down to VSS with a 10-kΩ ±5% resistor. If the PHY is permanently connected, then the MDIO pin should be externally pulled up to VCC with a 10-kΩ ±5% resistor.

External Address Detection Interface

$\overline{\text{EAR}}$

External Address Reject **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 acts as an external address accept function. The pin value is OR'd with the internal address detection result to determine if the current frame should be accepted. If $\overline{\text{EAR}}$ remains high while a frame is being received, the frame will be accepted regardless of the state of the internal address matching logic.

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

SFBD

Start Frame-Byte Delimiter **Output**

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

Downloaded from Arrow.com.

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

Note: *The SFBD signal can be programmed to appear on any of the LED pins.*

RXFRTGD

Receive Frame Tag Data **Input**

When the Receive Frame Tagging is enabled (RXFRTGEN, CMD3, bit 10), the RXFRTGD pin becomes a data input pin for the Receive Frame Tag. See the *Receive Frame Tagging* section for details.

Note: *The RXFRTGD pin is multiplexed with the $\overline{\text{LED3}}$ and EEDO pins.*

RXFRTGE

Receive Frame Tag Enable **Input**

When the Receive Frame Tagging is enabled (RXFRTGEN, CMD3, bit 10), the RXFRTGE pin becomes a data input enable pin for the Receive Frame Tag. See the *Receive Frame Tagging* section for details.

Note: *The RXFRTGE pin is multiplexed with the $\overline{\text{LED2}}$ pin.*

IEEE 1149.1 (1990) Test Access Port Interface

TCK

Test Clock **Input**

TCK is the clock input for the boundary scan test mode operation. It can operate at a frequency of up to 10 MHz. TCK has an internal pull-up resistor.

TDI

Test Data In **Input**

TDI is the test data input path to the Am79C976 controller. The pin has an internal pull-up resistor.

TDO

Test Data Out **Output**

TDO is the test data output path from the Am79C976 controller. The pin is tri-stated when the JTAG port is inactive.

TMS

Test Mode Select **Input**

A serial input bit stream on the TMS pin is used to define the specific boundary scan test to be executed. The pin has an internal pull-up resistor.

Power Supply Pins

AVDD

Analog Power (1 Pin) **Power**

This power supply pin is used for the internal oscillator and phase-locked loop circuits. This pin must be connected to a +3.3-V supply.

VSSB

I/O Buffer Ground (25 Pins) **Power**

There are 25 ground pins that are used by the input/output buffer drivers.

VDD

Digital and I/O Buffer Power (24 Pins)
Power

There are 24 power supply pins that are used by the internal digital circuitry and I/O buffers. All VDD pins must be connected to a +3.3 V supply.

VSS

Digital Ground (8 Pins) **Power**

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

Downloaded from Arrow.com.

BASIC FUNCTIONS

System Bus Interface

The Am79C976 controller is designed to operate as a bus master during normal operations. Some slave I/O accesses to the Am79C976 controller are required in normal operations as well. Initialization of the Am79C976 controller is achieved through a combination of PCI Configuration Space accesses, bus slave accesses, bus master accesses, and an optional read of a serial EEPROM that is performed by the Am79C976 controller. The EEPROM read operation is performed through the 93Cxx EEPROM interface. The ISO 8802-3 (IEEE/ANSI 802.3) Ethernet Address may reside within the serial EEPROM. Some Am79C976 controller configuration registers may also be programmed by the EEPROM read operation.

The Am79C976 controller requires 4 Kbytes of memory address space for access to all the various internal registers as well as access to some setup information stored in an external serial EEPROM. For compatibility with previous PCnet family devices, the lower 32 bytes of the register space are also mapped into I/O space, but some functions of the Am79C976 controller (such as network statistics) are only available in memory space. The location of the memory or I/O address space claimed by this device is programmed through the base address registers in PCI configuration space.

For diskless stations, the Am79C976 controller supports a ROM or Flash-based (both referred to as the *Expansion ROM* throughout this specification) boot device of up to 16 Mbyte in size. The host can map the boot device to any memory address that aligns to a device size boundary by modifying the Expansion ROM Base Address register in the PCI configuration space. The Expansion ROM device size is determined by the value set in the ROM-CFG register.

Software Interface

The software interface to the Am79C976 controller is divided into three parts. One part is the PCI configuration registers used to identify the Am79C976 controller and to setup the configuration of the device. The setup information includes the I/O or memory mapped I/O base address, mapping of the Expansion ROM, and the routing of the Am79C976 controller interrupt channel. This allows for a jumperless implementation.

The second portion of the software interface is the direct access to the I/O resources of the Am79C976 controller. The Am79C976 controller requires 4 Kbytes of memory address space for access to all the various internal registers as well as access to some setup information stored in an external serial EEPROM. For compatibility with previous PCnet family devices, the lower 32 bytes of the register space are also mapped into I/O space, but some functions of the Am79C976 controller (such as network statistics) are only available in memory space.

The third portion of the software interface is the descriptor and buffer areas that are shared between the software and the Am79C976 controller during normal network operations. The descriptor area boundaries are set by the software and do not change during normal network operations. There is one descriptor area for receive activity and there is a separate area for transmit activity. The descriptor space contains relocatable pointers to the network frame data, and it is used to transfer frame status from the Am79C976 controller to the software. The buffer areas are locations that hold frame data for transmission or that accept frame data that has been received.

Network Interface

The Am79C976 controller can be connected to an IEEE 802.3 or proprietary network through the IEEE 802.3-compliant Media Independent Interface (MII). The MII is a nibble-wide interface to an external 100-Mbps and/or 10-Mbps transceiver device.

The Am79C976 controller supports both half-duplex and full-duplex operation on the network interface.

Downloaded from Arrow.com.

DETAILED FUNCTIONS

Slave Bus Interface Unit

The slave bus interface unit (BIU) controls all accesses to the PCI configuration space, the Control and Status Registers (CSR), the Bus Configuration Registers (BCR), the Address PROM (APROM) locations, and the Expansion ROM. Table 2 shows the response of the Am79C976 controller to each of the PCI commands in slave mode.

Table 2. Slave Commands

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

Slave Configuration Transfers

The host can access the Am79C976 PCI configuration space with a configuration read or write command. The Am79C976 controller will assert DEVSEL during the address phase when IDSEL is asserted, AD[1:0] are both 0, and the access is a configuration cycle. AD[7:2] select the DWord location in the configuration space. The Am79C976 controller requires AD[10:8] to be 0, because it is a single function device. AD[31:11] are “don't care.”

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

The active bytes within a DWord are determined by the byte enable signals. Eight-bit, 16-bit, and 32-bit transfers are supported. DEVSEL is asserted two clock cycles after the host has asserted FRAME. All configuration cycles are of fixed length. The Am79C976 controller will assert TRDY on the third or fourth clock of the data phase.

The Am79C976 controller does not support burst transfers for access to configuration space. When the host keeps FRAME asserted for a second data phase, the Am79C976 controller will disconnect the transfer.

When the host tries to access the PCI configuration space while the automatic read of the EEPROM after H_RESET (see section on RESET) is on-going, the Am79C976 controller will terminate the access on the PCI bus with a disconnect/retry response.

The Am79C976 controller supports fast back-to-back transactions to different targets. This is indicated by the Fast Back-To-Back Capable bit (PCI Status register, bit 7), which is hardwired to 1. The Am79C976 controller is capable of detecting a configuration cycle even when its address phase immediately follows the data phase of a transaction to a different target without any idle state in-between. There will be no contention on the DEVSEL, TRDY, and STOP signals, since the Am79C976 controller asserts DEVSEL on the second clock after FRAME is asserted (medium timing).

Slave I/O Transfers

After the Am79C976 controller is configured as an I/O device by setting IOEN (for regular I/O mode) or MEMEN (for memory mapped I/O mode) in the PCI Command register, it starts monitoring the PCI bus for access to its address space. If configured for regular I/O mode, the Am79C976 controller will look for an address that falls within its 32 bytes of I/O address space (starting from the I/O base address). The Am79C976 controller asserts DEVSEL if it detects an address match and the access is an I/O cycle. If configured for memory mapped I/O mode, the Am79C976 controller will look for an address that falls within its 4096 bytes of memory address space (starting from the memory mapped I/O base address). The Am79C976 controller asserts DEVSEL if it detects an address match and the

Downloaded from Arrow.com.

access is a memory cycle. DEVSEL is asserted two clock cycles after the host has asserted FRAME. See Figure 1 and Figure 2.

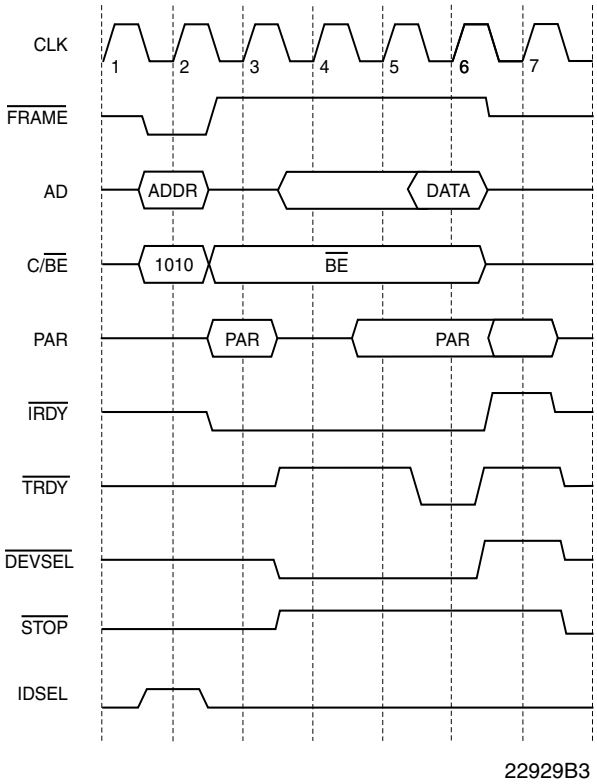


Figure 1. Slave Configuration Read

The Am79C976 controller will not assert DEVSEL if it detects an address match, but the PCI command is not of the correct type. In memory mapped I/O mode, the Am79C976 controller aliases all accesses to the I/O resources of the command types *Memory Read Multiple* and *Memory Read Line* to the basic Memory Read command. All accesses of the type *Memory Write and Invalidate* are aliased to the basic Memory Write command. Eight-bit, 16-bit, and 32-bit transactions are supported. The Am79C976 controller decodes all 32 address lines to determine which I/O resource is accessed.

The number of wait states added to slave transactions varies. Typical values are shown in the table below:

Slave Transactions	Transaction Type	
	Read	Write
Memory-mapped transactions	9	0
I/O-mapped transactions	9	3

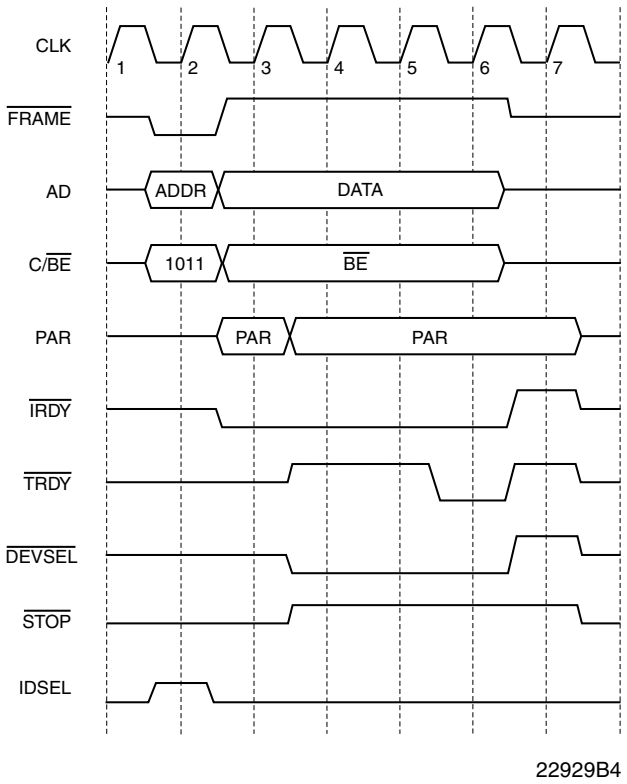


Figure 2. Slave Configuration Write

For compatibility with older members of the PCnet family of controllers, the 32 lowest addresses of the I/O or memory space claimed by the Am79C976 device support indirect addressing of internal registers. The Am79C976 controller does not support burst transfers for access to these locations. When the host keeps FRAME asserted for a second data phase in this address range, the Am79C976 controller will disconnect the transfer. However, the controller does support burst accesses to locations at offsets 32 and above.

Because of the side effects of reading the Reset Register at offset 14h or 18h (depending on the state of DWIO (CMD2, bit 28)), locations at offsets less than 20h cannot be prefetched.

The Am79C976 controller supports fast back-to-back transactions to different targets. This is indicated by the Fast Back-To-Back Capable bit (PCI Status register, bit 7), which is hardwired to 1. The Am79C976 controller is capable of detecting an I/O or a memory-mapped I/O cycle even when its address phase immediately follows the data phase of a transaction to a different target, without any idle state in-between. There will be no contention on the DEVSEL, TRDY, and STOP signals, since the Am79C976 controller asserts DEVSEL on the second clock after FRAME is asserted (medium timing). See Figure 3 and Figure 4.

Downloaded from Arrow.com.

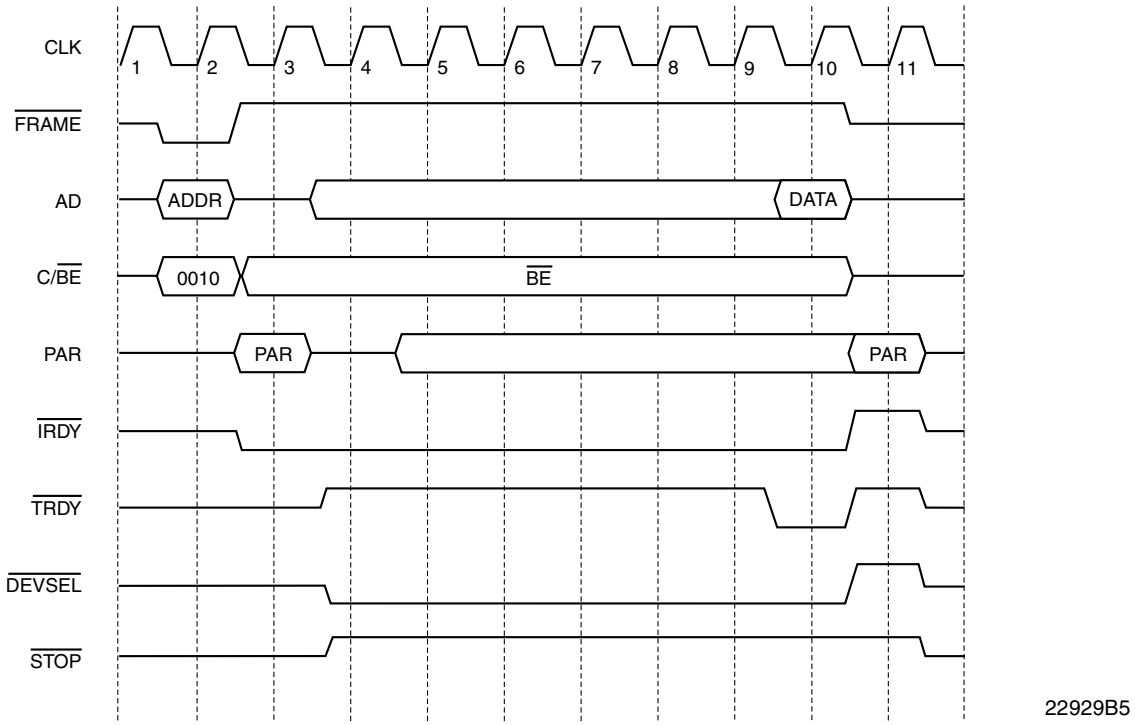


Figure 3. Slave Read Using I/O Command

Downloaded from Arrow.com.

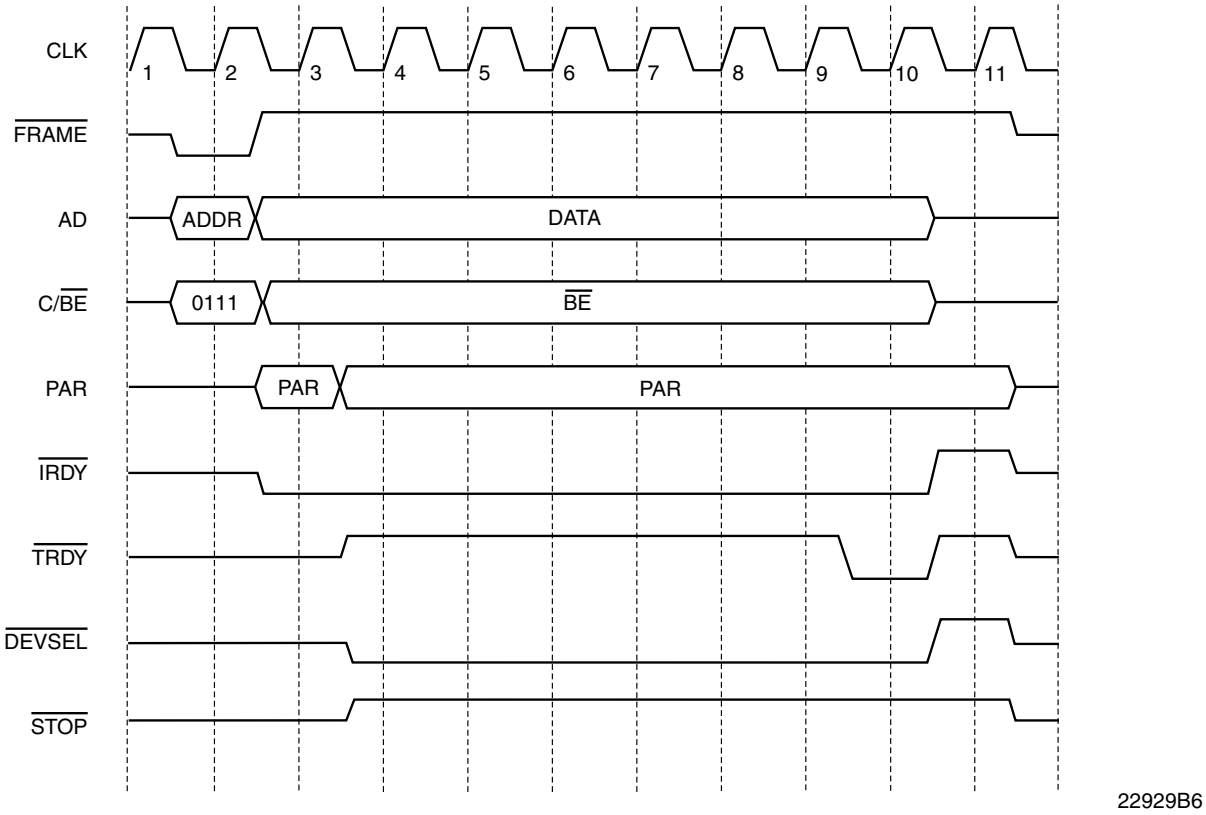


Figure 4. Slave Write Using Memory Command

Downloaded from Arrow.com.

Expansion ROM Transfers

The Am79C976 device includes an interface to an optional expansion ROM. The amount of PCI address space claimed by this ROM is determined by the contents of the ROM Configuration Register, ROM_CFG, which should normally be loaded from the serial EEPROM.

The host must initialize the Expansion ROM Base Address register at offset 30H in the PCI configuration space with a valid address before enabling the access to the device. The Am79C976 controller will not react to any access to the Expansion ROM until both MEMEN (PCI Command register, bit 1) and ROMEN (PCI Expansion ROM Base Address register, bit 0) are set to 1. After the Expansion ROM is enabled, the Am79C976 controller will assert DEVSEL on all memory read accesses to the memory space defined by the contents of the Expansion ROM Base Address register. The Am79C976 controller aliases all accesses to the Expansion ROM of the command types *Memory Read Multiple* and *Memory Read Line* to the basic Memory Read command. Eight-bit, 16-bit, and 32-bit read transfers are supported.

Since setting MEMEN also enables memory mapped access to the I/O resources, attention must be given to the PCI Memory Mapped I/O Base Address register before enabling access to the Expansion ROM. The host must set the PCI Memory Mapped I/O Base Address register to a value that prevents the Am79C976 controller from claiming any memory cycles not intended for it.

The Am79C976 controller will always read four bytes for every host Expansion ROM read access. Since this takes more than 16 PCI clock cycles, the Am79C976 device will assert STOP to force a PCI bus retry. Subsequent accesses will be retried until all four bytes have been read from the ROM and stored in an internal temporary register. The timing of the access to the ROM device is determined by the ROMTMG parameter (CTRL0, bits 11-8).

Note: *The Expansion ROM must not be read when the Am79C976 controller is running (when the RUN bit in CMD0 is set to 1). Any access to the Expansion ROM clears the RUN bit and thereby abruptly stops all network and DMA operations.*

When the host tries to write to the Expansion ROM, the Am79C976 controller will claim the cycle. The write operation will have no effect. Writes to the Expansion ROM are done through the BCR30 Expansion Bus Data Port. See the section on the *Expansion Bus Interface* for more details.

During the boot procedure, the system will try to find an Expansion ROM. A PCI system assumes that an Expansion ROM is present when it reads the ROM signature 55H (byte 0) and AAH (byte 1).

Slave Cycle Termination

In addition to the normal completion of a transaction, there are three scenarios in which the Am79C976 controller terminates a slave access for which it is the target.

Disconnect When Busy

If a slave access to the Am79C976 device takes more than 16 PCI CLK cycles, the Am79C976 device will generate a PCI disconnect/retry cycle by asserting STOP and deasserting TRDY while keeping DEVSEL asserted. This will free up the PCI bus so that it can be used by other bus masters while the Am79C976 device is busy. See Figure 5.

The Am79C976 controller cannot service any slave access while it is reading the contents of the EEPROM. Simultaneous access is not allowed in order to avoid conflicts, since the EEPROM is used to initialize some of the PCI configuration space locations and user-selected BCRs and CSRs. The EEPROM read operation will always happen automatically following H_RESET. (See the *H_RESET* section for more details.) In addition, the host can start the read operation by setting the PREAD bit (BCR19, bit 14). While the EEPROM read is on-going, the Am79C976 controller will disconnect any slave access where it is the target by asserting STOP together with DEVSEL, while driving TRDY high. STOP will stay asserted until the end of the cycle.

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

The Am79C976 device will also generate PCI disconnect/retry cycles when it is executing a blocking read access to an external PHY register.

Downloaded from Arrow.com.

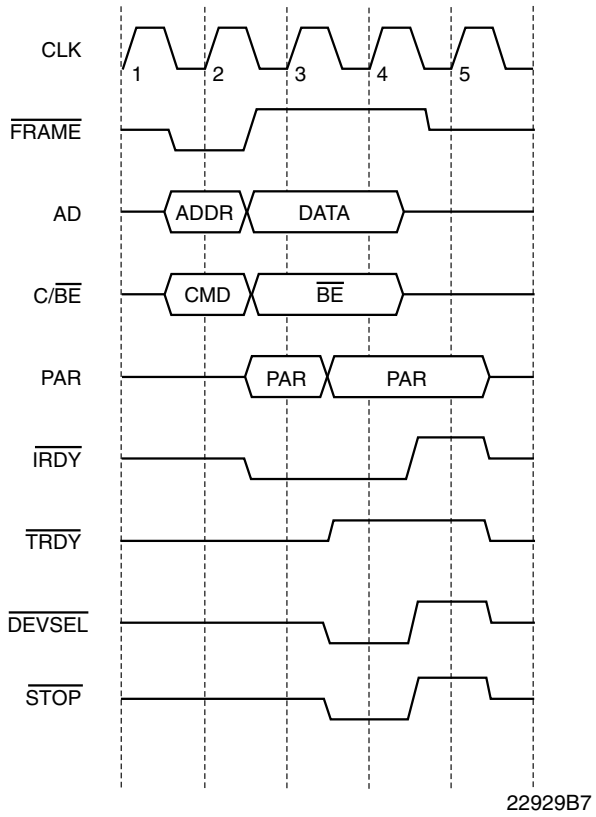


Figure 5. Disconnect Of Slave Cycle When Busy

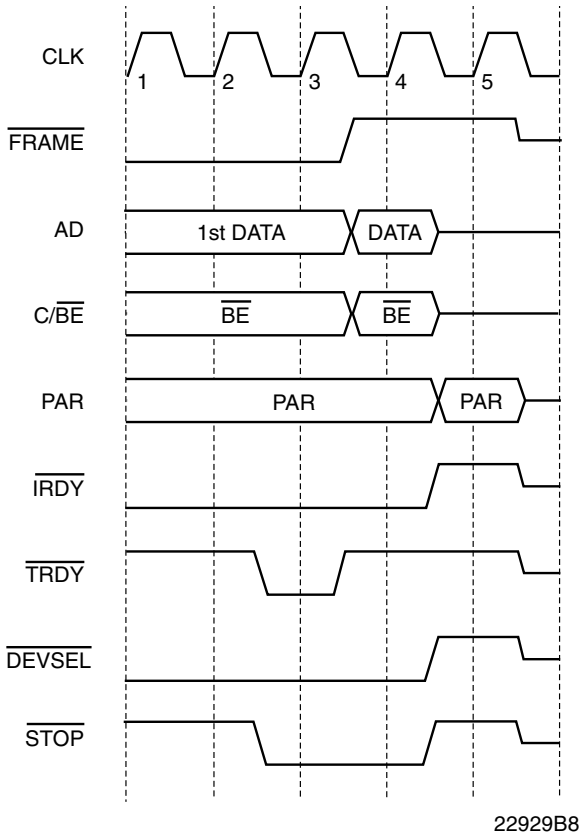


Figure 6. Disconnect Of Slave Burst Transfer - No Host Wait States

Disconnect Of Burst Transfer

The Am79C976 controller does not support burst access to the configuration space, the first 32 bytes of its I/O or memory space, or to the Expansion Bus. The host indicates a burst transaction by keeping FRAME asserted during the data phase. When the Am79C976 controller sees FRAME and IRDY asserted in the clock cycle before it wants to assert TRDY, it also asserts STOP at the same time. The transfer of the first data phase is still successful, since IRDY and TRDY are both asserted. See Figure 6.

If the host is not yet ready when the Am79C976 controller asserts TRDY, the device will wait for the host to assert IRDY. When the host asserts IRDY and FRAME is still asserted, the Am79C976 controller will finish the first data phase by deasserting TRDY one clock later. At the same time, it will assert STOP to signal a disconnect to the host. STOP will stay asserted until the host removes FRAME. See Figure 7.

Downloaded from Arrow.com.

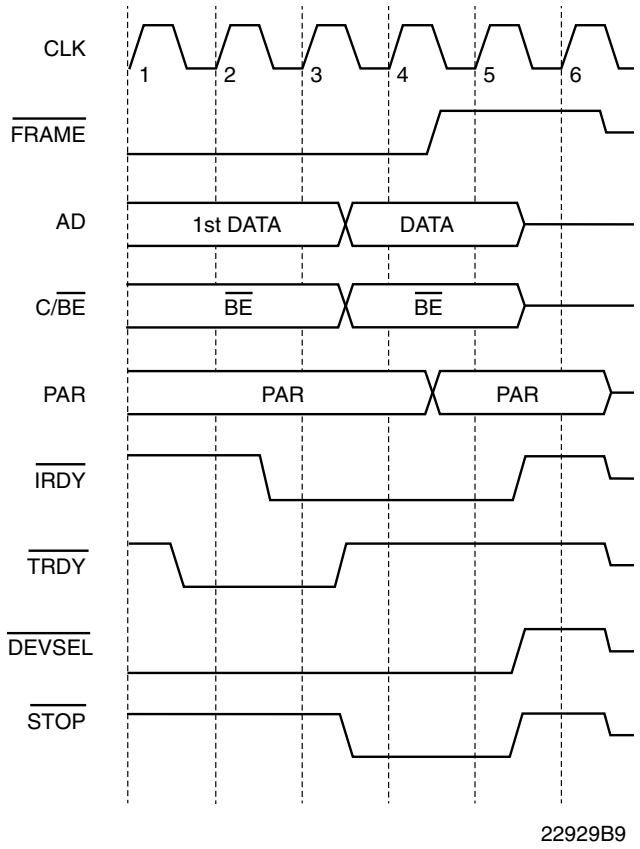


Figure 7. Disconnect Of Slave Burst Transfer - Host Inserts Wait States

Parity Error Response

When the Am79C976 controller is not the current bus master, it samples the AD[31:0], C/BE[3:0], and the PAR lines during the address phase of any PCI command for a parity error. When it detects an address parity error, the controller sets PERR (PCI Status register, bit 15) to 1. When reporting of that error is enabled by setting SERREN (PCI Command register, bit 8) and PERREN (PCI Command register, bit 6) to 1, the Am79C976 controller also drives the SERR signal low for one clock cycle and sets SERR (PCI Status register, bit 14) to 1. The assertion of SERR follows the address phase by two clock cycles. The Am79C976 controller will not assert DEVSEL for a PCI transaction that has an address parity error when PERREN and SERREN are set to 1. See Figure 8.

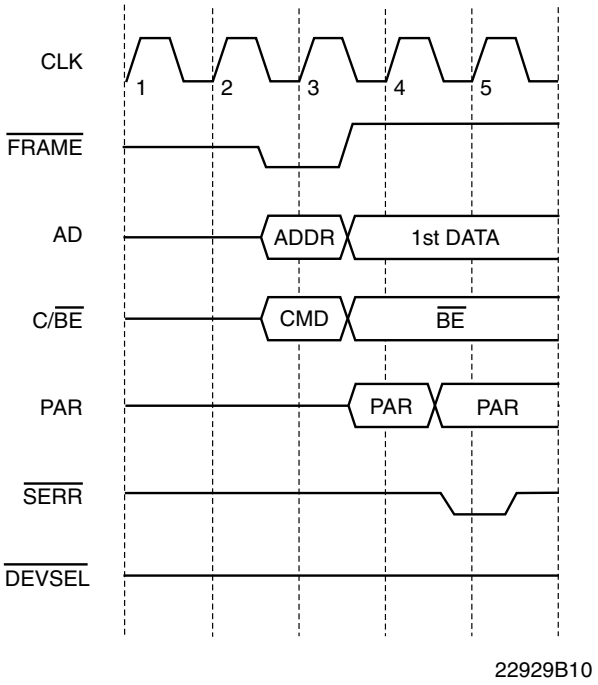


Figure 8. Address Parity Error Response

During the data phase of an I/O write, memory-mapped I/O write, or configuration write command that selects the Am79C976 controller as target, the device samples the AD[31:0] and C/BE[3:0] lines for parity on the clock edge, and data is transferred as indicated by the assertion of IRDY and TRDY. PAR is sampled in the following clock cycle. If a parity error is detected and reporting of that error is enabled by setting PERREN (PCI Command register, bit 6) to 1, PERR is asserted one clock later. The parity error will always set PERR (PCI Status register, bit 15) to 1 even when PERREN is cleared to 0. The Am79C976 controller will finish a transaction that has a data parity error in the normal way by asserting TRDY. The corrupted data will be written to the addressed location.

Figure 9 shows a transaction that suffered a parity error at the time data was transferred (clock 7, IRDY and TRDY are both asserted). PERR is driven high at the beginning of the data phase and then drops low due to the parity error on clock 9, two clock cycles after the data was transferred. After PERR is driven low, the Am79C976 controller drives PERR high for one clock cycle, since PERR is a sustained tri-state signal.

Downloaded from Arrow.com.

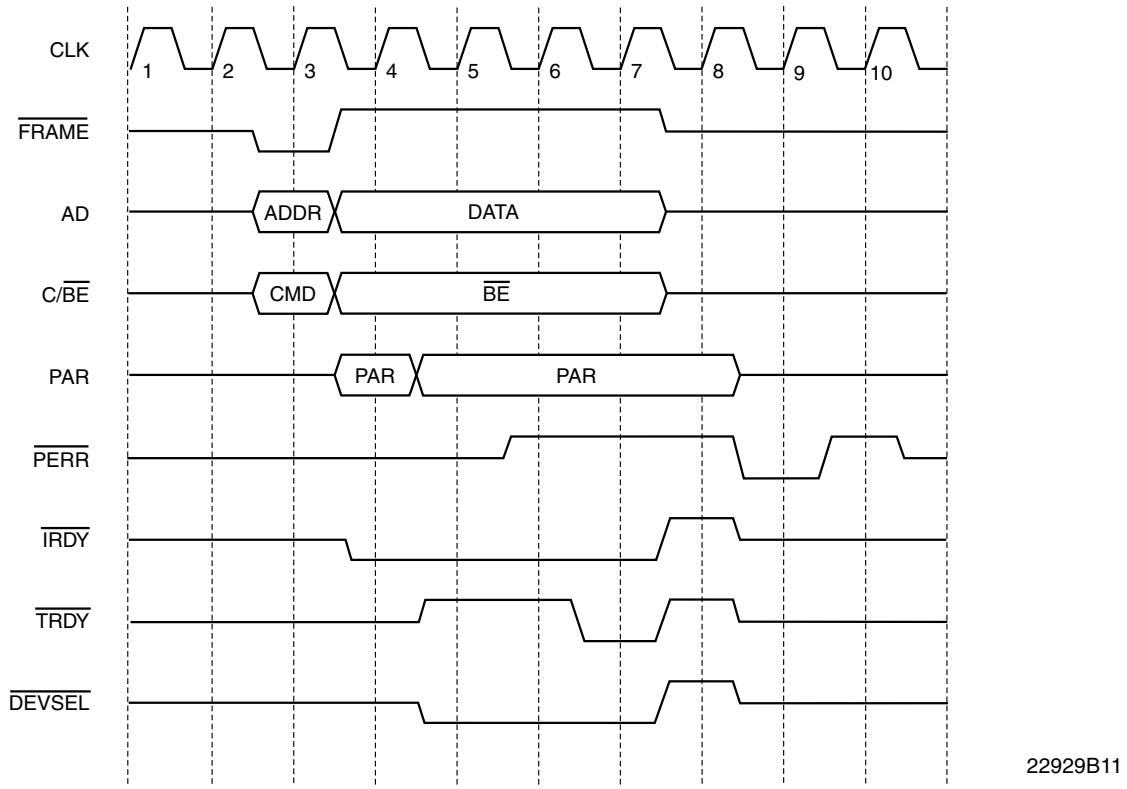


Figure 9. Slave Cycle Data Parity Error Response

Master Bus Interface Unit

The master Bus Interface Unit (BIU) controls the acquisition of the PCI bus and all accesses to the initialization block, descriptor rings, and the receive and transmit buffer memory. Table 3 shows the usage of PCI commands by the Am79C976 controller in master mode.

Bus Acquisition

The Am79C976 logic will determine when a DMA transfer should be initiated. The first step in any Am79C976 bus master transfer is to acquire ownership of the bus. This task is handled by synchronous logic within the BIU. Bus ownership is requested with the REQ signal and ownership is granted by the arbiter through the GNT signal.

Figure 10 shows the Am79C976 controller bus acquisition. REQ is asserted and the arbiter returns GNT while another bus master is transferring data. The Am79C976 controller waits until the bus is idle (FRAME and IRDY deasserted) before it starts driving AD[31:0] and C/BE[3:0] on clock 5. FRAME is asserted at clock 5 indicating a valid address and command on AD[31:0] and C/BE[3:0]. The Am79C976 controller does not use address stepping which is reflected by ADSTEP (bit 7) in the PCI Command register being hardwired to 0.

Bus Master DMA Transfers

There are four primary types of DMA transfers. The Am79C976 controller uses non-burst as well as burst cycles for read and write access to the main memory.

Basic Non-Burst Read Transfer

The Am79C976 controller uses non-burst cycles to access descriptors when SWSTYLE (BCR20, bits 7-0) is 0 or 2. All Am79C976 controller non-burst read accesses are of the PCI command type Memory Read (type 6). Note that during a non-burst read operation, all byte lanes will always be active. The Am79C976 controller will internally discard unneeded bytes.

Downloaded from Arrow.com.

Table 3. PCI Commands

C[3:0]	Command	Use
0000	Interrupt Acknowledge	Not used
0001	Special Cycle	Not used
0010	I/O Read	Not used
0011	I/O Write	Not used
0100	Reserved	
0101	Reserved	
0110	Memory Read	Read of the initialization block and descriptor rings Read of the transmit buffer in non-burst mode
0111	Memory Write	Write to the descriptor rings and to the receive buffer
1000	Reserved	
1001	Reserved	
1010	Configuration Read	Not used
1011	Configuration Write	Not used
1100	Memory Read Multiple	Read of descriptor or transmit buffer in burst mode
1101	Dual Address Cycle	Used when required
1110	Memory Read Line	Read of descriptor or transmit buffer in burst mode
1111	Memory Write Invalidate	Burst write of 1 or more complete cache lines to the receive buffer

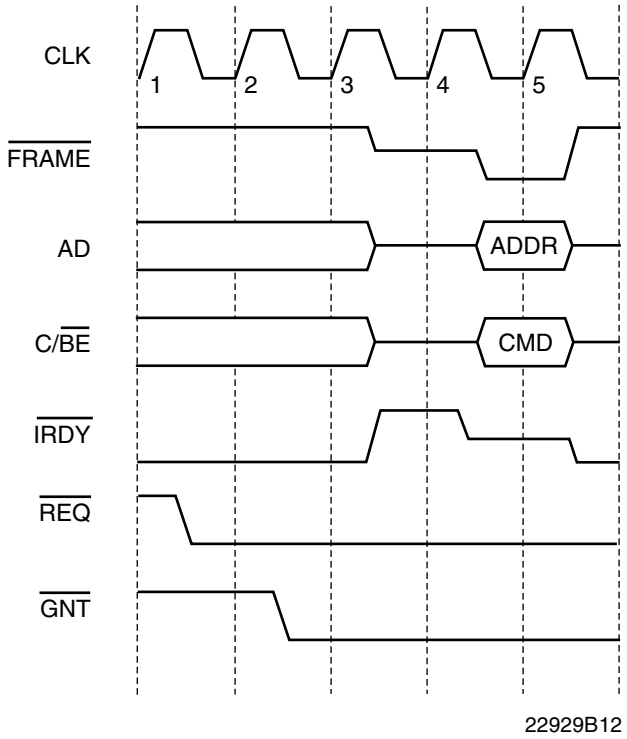


Figure 10. Bus Acquisition

The Am79C976 controller typically performs more than one non-burst read transaction within a single bus mastership period. FRAME is dropped between consecutive non-burst read cycles. REQ, however, stays asserted until FRAME is asserted for the last transaction. The Am79C976 controller supports zero wait-state read cycles. It asserts IRDY immediately after the address phase and at the same time starts sampling DEVSEL. Figure 11 shows two non-burst read transactions. The first transaction has zero wait states. In the second transaction, the target extends the cycle by asserting TRDY one clock later.

Basic Burst Read Transfer

The Am79C976 controller supports burst mode for all bus master read operations. To allow burst transfers in descriptor read operations, the Am79C976 controller must be programmed to use SWSTYLE 3, 4, or 5 (BCR20, bits 7-0).

The BIU chooses which PCI command to use as follows:

- When reading one DWord, use Memory Read.
- When reading a block of more than one DWord that does not cross a cache line, use Memory Read Line.
- When reading a block that crosses a cache line boundary, use Memory Read Multiple.

Downloaded from Arrow.com.

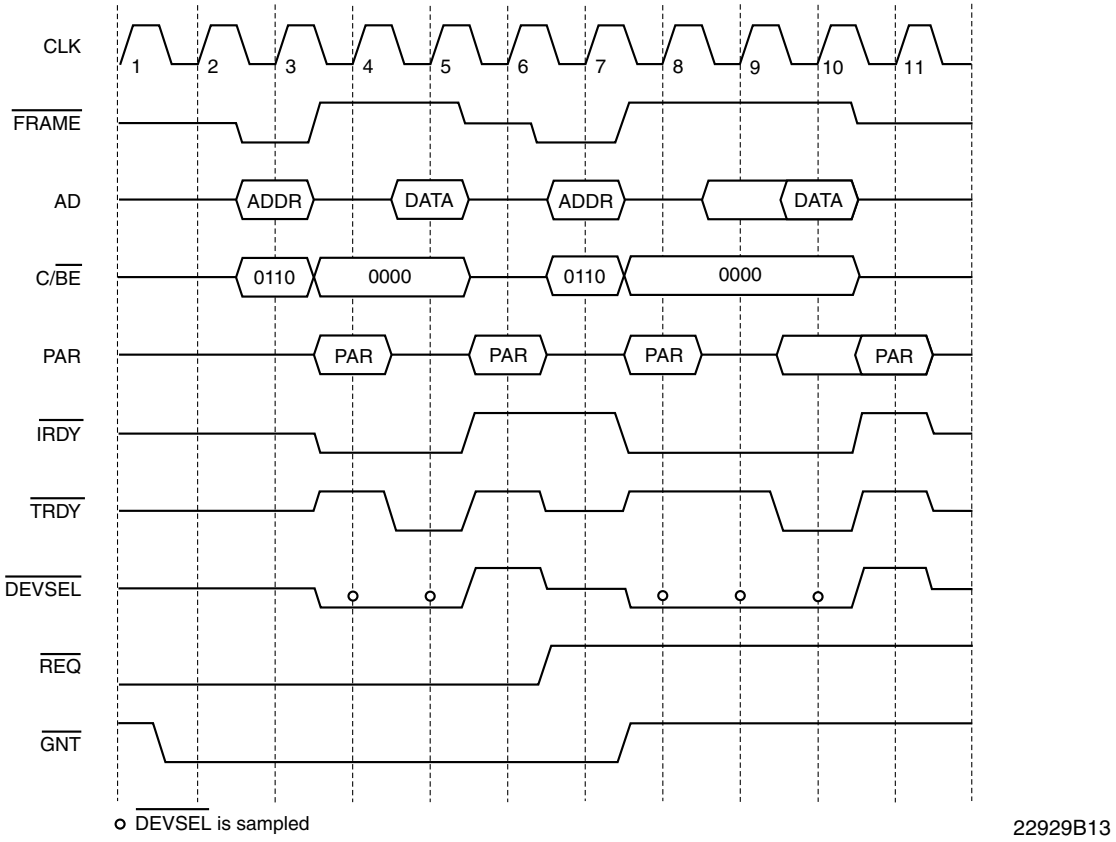


Figure 11. Non-Burst Read Transfer

The FIFO thresholds should be greater than or equal to the cache line size to maximize the use of the MRL and MRM commands. If the PCI bridge stops a transfer, the Am79C976 device waits until the FIFO threshold conditions are met before resuming the transfer.

During the address phase of a burst access, AD[1:0] will both be 0 indicating a linear burst order. Note that during a burst read operation, all byte lanes will always be active. The Am79C976 controller will internally discard unneeded bytes.

The Am79C976 controller will always perform only a single burst read transaction per bus mastership period, where *transaction* is defined as one address phase and one or multiple data phases. The Am79C976 controller supports zero wait state read cycles. It asserts IRDY immediately after the address phase and at the same time starts sampling DEVSEL. FRAME is deasserted when the next-to-last data phase is completed.

The device may insert IRDY wait states in the middle of a burst read transaction.

Figure 12 shows a typical burst read access. The Am79C976 controller arbitrates for the bus, is granted access, reads three 32-bit words (DWord) from the system memory, and then releases the bus. In the example, the memory system extends the data phase of each access by one wait state.

Basic Non-Burst Write Transfer

The Am79C976 controller uses non-burst cycles to write descriptors when SWSTYLE (BCR20, bits 7-0) is 0 or 2. All Am79C976 controller non-burst write accesses are of the PCI command type Memory Write (type 7). The byte enable signals indicate the byte lanes that have valid data.The Am79C976 controller may perform more than one non-burst write transaction within a single bus mastership period. FRAME is dropped between consecutive non-burst write cycles. REQ, however, stays asserted until FRAME is asserted for the last transaction. The Am79C976 supports zero wait state write cycles. (See the section *Descriptor DMA Transfers* for the only exception.) It asserts IRDY immediately after the address phase.

Figure 13 shows two non-burst write transactions. The first transaction has two wait states. The Am79C976 device supports zero wait state non-burst write cycles.

Downloaded from Arrow.com.

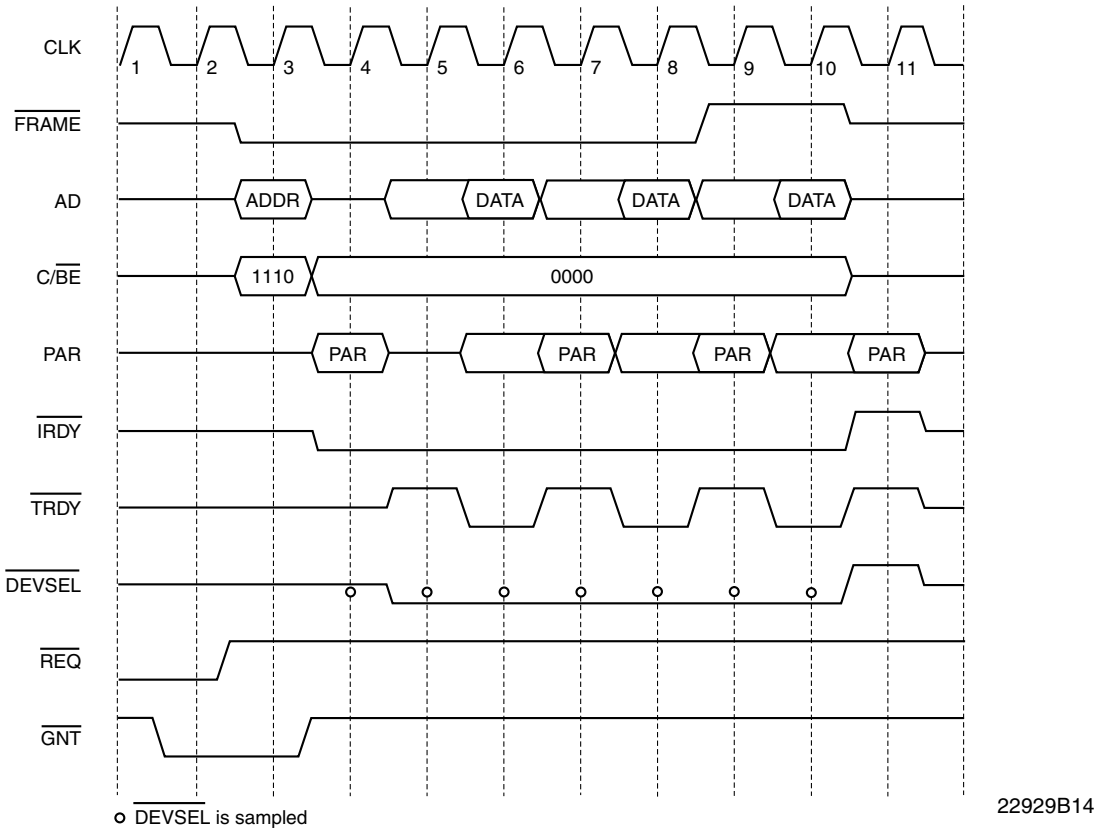


Figure 12. Burst Read Transfer

Downloaded from Arrow.com.

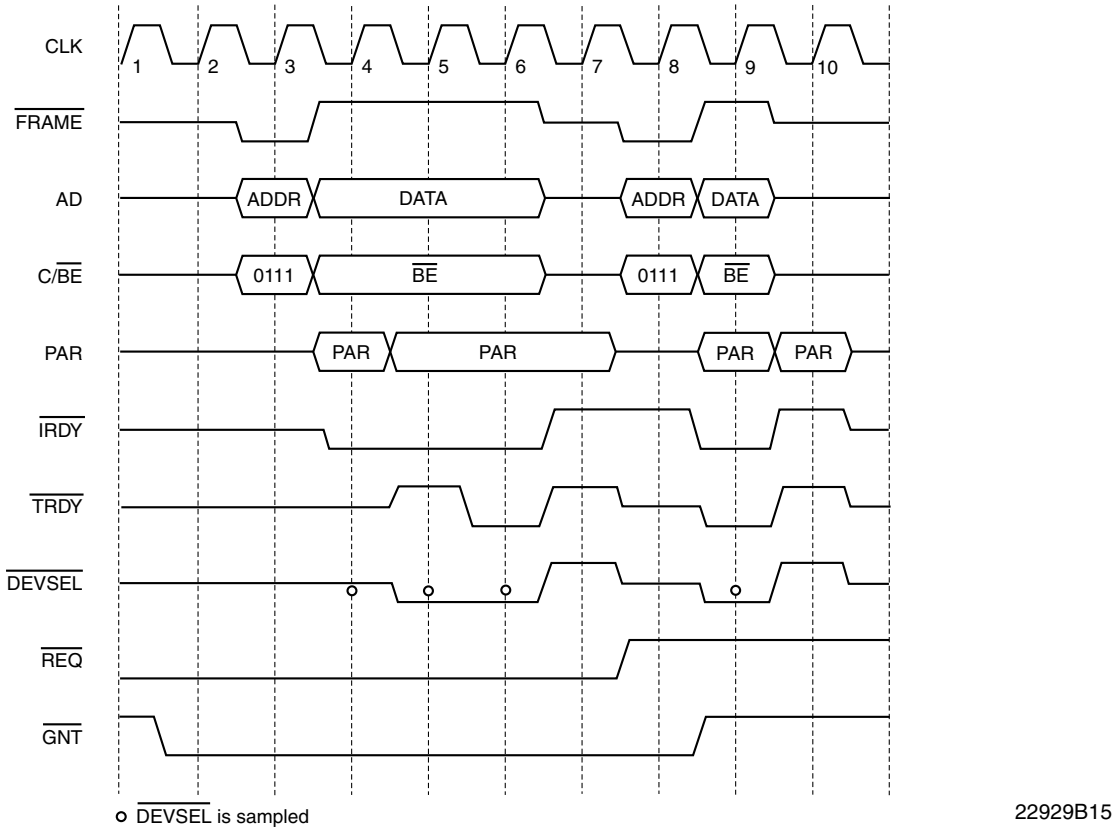


Figure 13. Non-Burst Write Transfer

Basic Burst Write Transfer

The Am79C976 controller supports burst mode for all bus master write operations. To allow burst transfers in descriptor write operations, the Am79C976 controller must be programmed to use SWSTYLE 3, 4, or 5 (BCR20, bits 7-0).

The controller uses the following rules to determine whether to use the PCI Memory Write (MW) command or the Memory Write and Invalidate (MWI) command for burst write transfers.

— When a transfer starts on a cache line boundary, and there is at least a cache line of data to transfer, use MWI.

— When a transfer does not start on a cache line boundary, use MW. (The external PCI bridge should stop the transfer at the cache line boundary if it can make good use of the MWI command.)

— Stop the MWI transfer at a cache line boundary if there is less than 1 cache line of data left to transfer.

The Receive FIFO threshold should be greater than or equal to the cache line size to maximize the use of the MWI command. If the PCI bridge stops a transfer, the Am79C976 device waits until the FIFO threshold conditions are met before resuming the transfer.

During the address phase of a burst write transfer AD[1:0] will both be 0 indicating a linear burst order. The byte enable signals indicate which byte lanes have valid data.

The Am79C976 controller will always perform a single burst write transaction per bus mastership period, where transaction is defined as one address phase and one or multiple data phases. The Am79C976 controller supports zero wait state write cycles when using the Memory Write command. When using Memory Write and Invalidate commands, the device may insert IRDY wait states anywhere in the transaction.

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

Figure 14 shows a typical burst write access. The Am79C976 controller arbitrates for the bus, is granted access, and writes four 32-bit words (DWords) to the system memory and then releases the bus. In this ex-

Downloaded from Arrow.com.

ample, the memory system extends the data phase of the first access by one wait state. The following three data phases take one clock cycle each, which is determined by the timing of TRDY.

DMA Burst Alignment

The BIU has two programmable features that can improve the DMA performance with PCI bridges that do not automatically stop burst transfers to align them with cache line boundaries:

1. The Burst Alignment (BA) bit (CTRL0, bit 4). When this bit is set, if a burst transfer starts in the middle of a cache line, the transfer will stop at the first cache line boundary.
2. The Burst Limit Register (CTRL0, bits 3:0). This 4-bit register limits the maximum length of a burst transfer. If the contents of this register are 0, the burst length is limited by the amount of data available or by the amount of FIFO space available.

   If the contents of this register are not zero, a burst transfer will end when the transfer has crossed the number of cache line boundaries equal to the contents of this register.

Target Initiated Termination

When the Am79C976 controller is a bus master, the cycles it produces on the PCI bus may be terminated by the target in one of three different ways: disconnect with data transfer, disconnect without data transfer, and target abort.

Disconnect With Data Transfer

Figure 15 shows a disconnection in which one last data transfer occurs after the target asserted STOP. STOP is asserted on clock 4 to start the termination sequence. Data is still transferred during this cycle, since both IRDY and TRDY are asserted. The Am79C976 controller terminates the current transfer with the deassertion of FRAME on clock 5 and of IRDY one clock later. It finally releases the bus on clock 7. The Am79C976 controller will again request the bus after two clock cycles, if it wants to transfer more data. The starting address of the new transfer will be the address of the next non-transferred data.

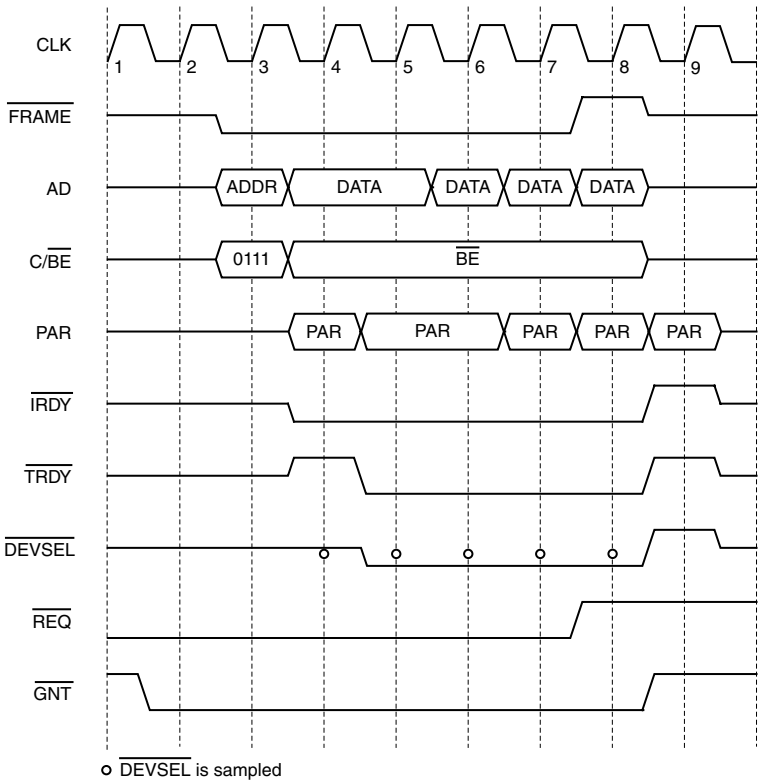


Figure 14. Burst Write Transfer

Downloaded from Arrow.com.

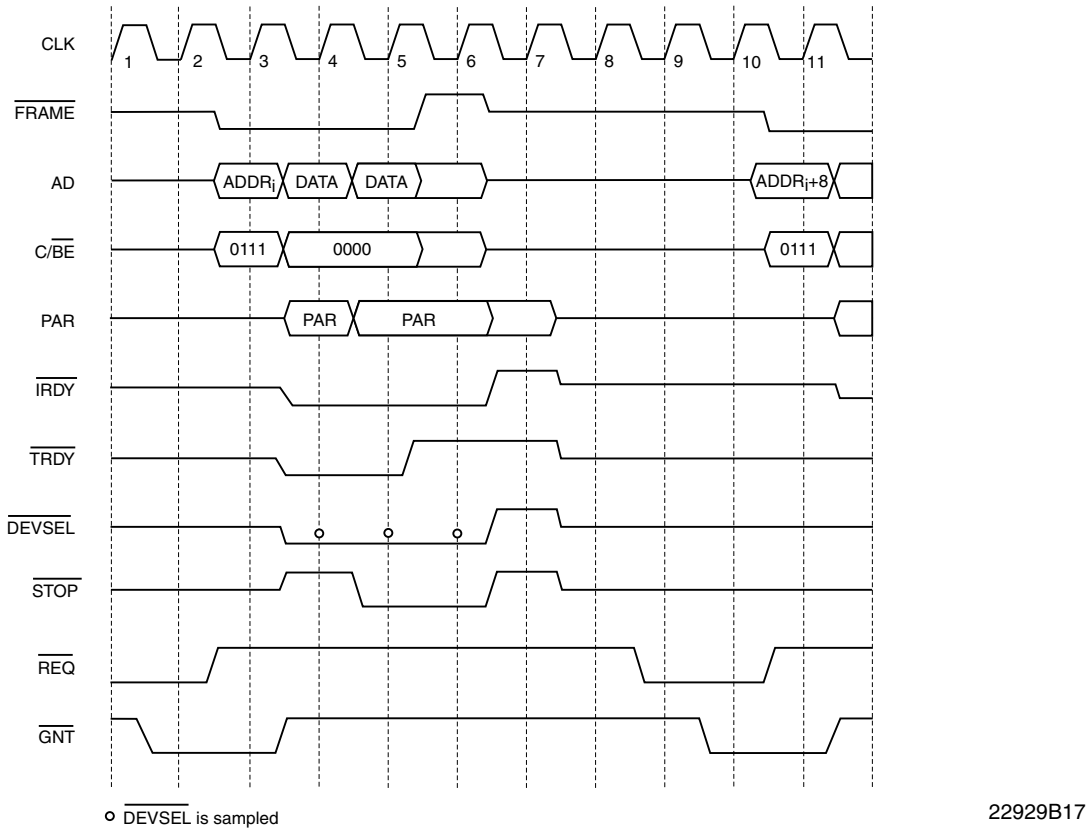


Figure 15. Disconnect With Data Transfer

Disconnect Without Data Transfer

Figure 16 shows a target disconnect sequence during which no data is transferred. STOP is asserted on clock 4 without TRDY being asserted at the same time. The Am79C976 controller terminates the access with the deassertion of FRAME on clock 5 and of IRDY one clock cycle later. It finally releases the bus on clock 7. The Am79C976 controller will again request the bus after two clock cycles to retry the last transfer. The starting address of the new transfer will be the address of the last non-transferred data.

Downloaded from Arrow.com.

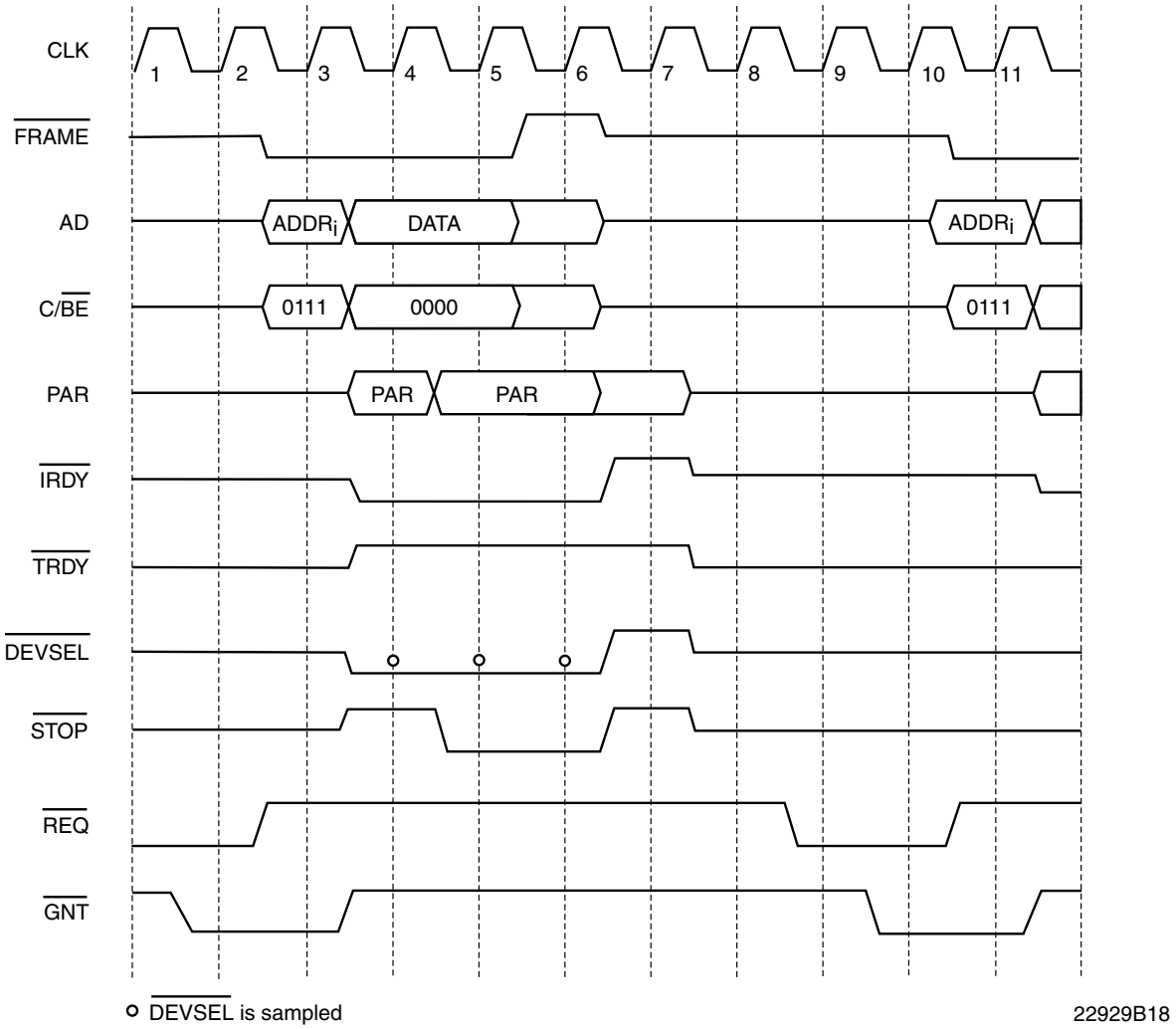


Figure 16. Disconnect Without Data Transfer

Target Abort

Figure 17 shows a target abort sequence. The target asserts DEVSEL for one clock. It then deasserts DEVSEL and asserts STOP on clock 4. A target can use the target abort sequence to indicate that it cannot service the data transfer and that it does not want the transaction to be retried. Additionally, the Am79C976 controller cannot make any assumption about the success of the previous data transfers in the current transaction. The Am79C976 controller terminates the current transfer with the deassertion of FRAME on clock 5 and of IRDY one clock cycle later. It finally releases the bus on clock 7.

Since data integrity is not guaranteed, the Am79C976 controller cannot recover from a target abort event. The Am79C976 controller will reset all CSR locations to their STOP_RESET values. The BCR and PCI configuration registers will not be cleared. Any on-going network transmission is terminated with the current FCS inverted and appended at the next byte boundary. This guarantees that the receiving station will drop the truncated frame.

Downloaded from Arrow.com.

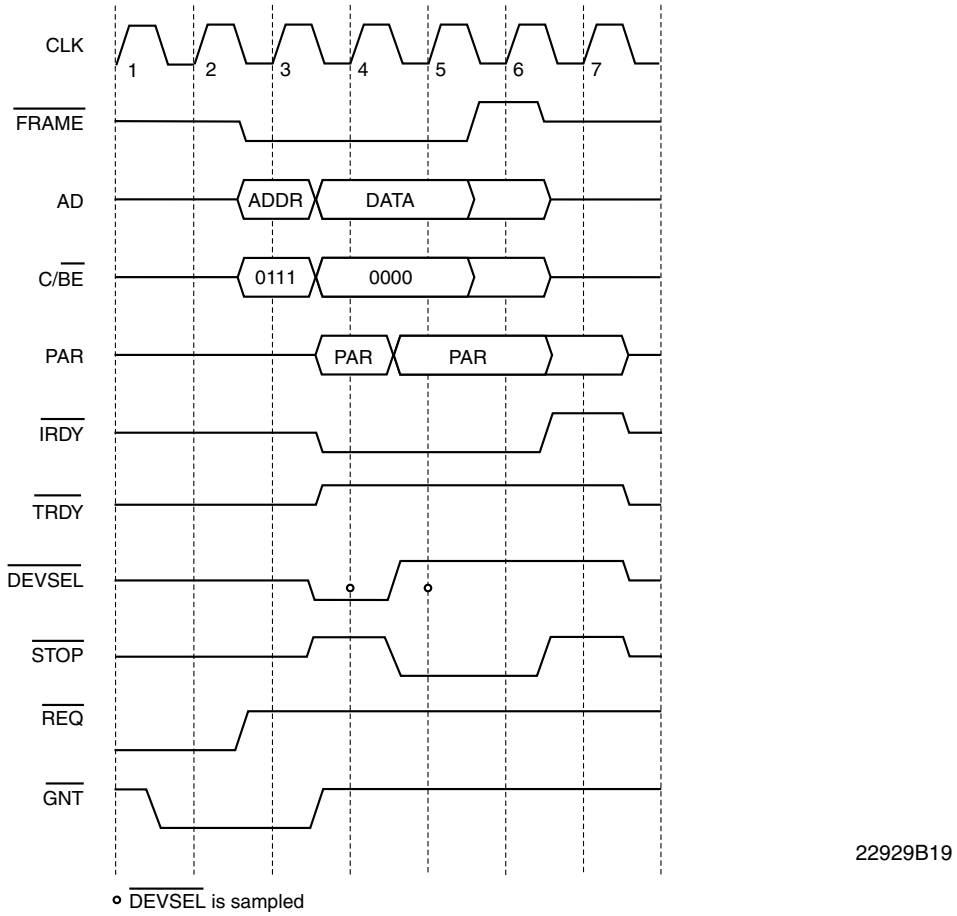


Figure 17. Target Abort

RTABORT (PCI Status register, bit 12) will be set to indicate that the Am79C976 controller has received a target abort. In addition, SINT (CSR5, bit 11) will be set to 1. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to 1. This mechanism can be used to inform the driver of the system error. The host can read the PCI Status register to determine the exact cause of the interrupt.

Master Initiated Termination

There are three scenarios besides normal completion of a transaction where the Am79C976 controller will terminate the cycles it produces on the PCI bus.

Preemption During Non-Burst Transaction

When the Am79C976 controller performs multiple non-burst transactions, it keeps REQ asserted until the assertion of FRAME for the last transaction. When GNT is removed, the Am79C976 controller will finish the current transaction and then release the bus. If it is not the last transaction, REQ will remain asserted to regain bus ownership as soon as possible. See Figure 18.

Preemption During Burst Transaction

When the Am79C976 controller operates in burst mode, it only performs a single transaction per bus mastership period, where *transaction* is defined as one address phase and one or multiple data phases. The central arbiter can remove GNT at any time during the transaction. The Am79C976 controller will ignore the deassertion of GNT and continue with data transfers, as long as the PCI Latency Timer is not expired. When the Latency Timer is 0 and GNT is deasserted, the Am79C976 controller will finish the current data phase, deassert FRAME, finish the last data phase, and release the bus. It will immediately assert REQ to regain bus ownership as soon as possible.

When the preemption occurs after the counter has counted down to 0, the Am79C976 controller will finish the current data phase, deassert FRAME, finish the last data phase, and release the bus. Note that it is important for the host to program the PCI Latency Timer according to the bus bandwidth requirement of the Am79C976 controller. The host can determine this bus bandwidth requirement by reading the PCI MAX_LAT and MIN_GNT registers.

Downloaded from Arrow.com.

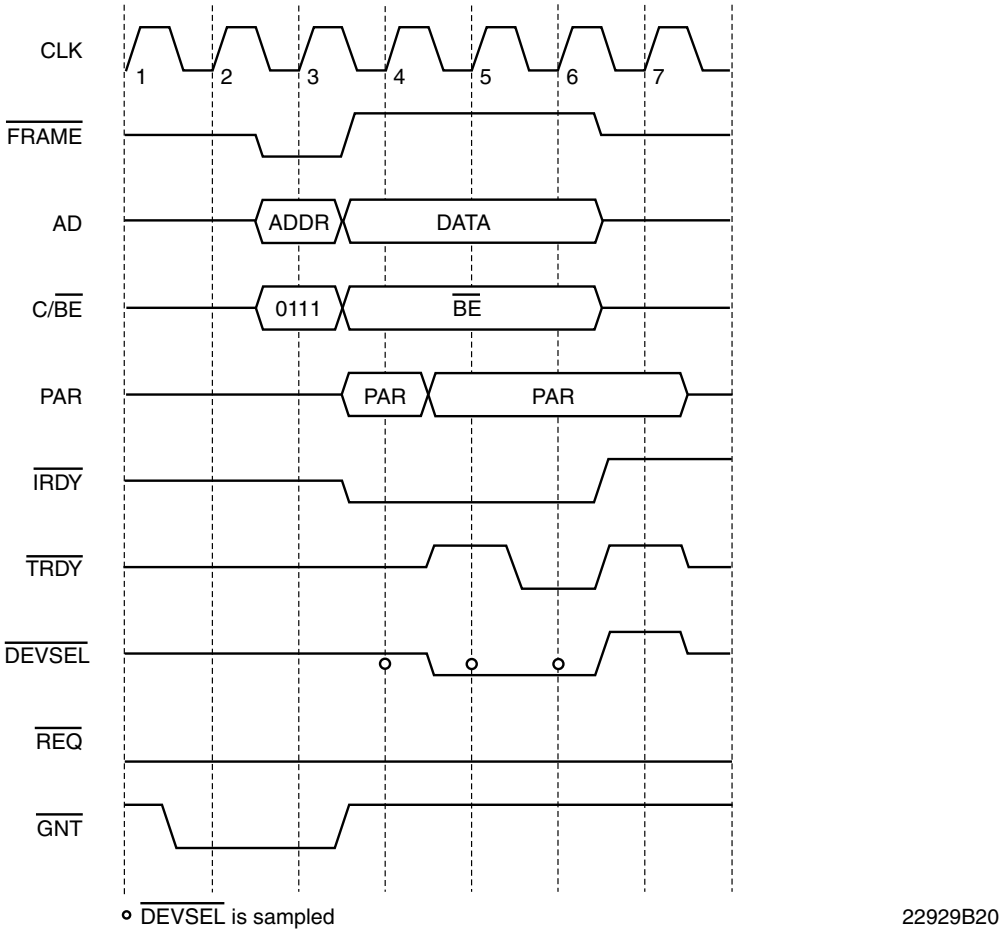


Figure 18. Preemption During Non-Burst Transaction

If the controller is executing a Memory Write and Invalidate instruction when preemption occurs, the controller will finish writing the current cache line before it releases the bus.

Figure 19 assumes that the PCI Latency Timer has counted down to 0 on clock 7.

Master Abort

The Am79C976 controller will terminate its cycle with a Master Abort sequence if DEVSEL is not asserted within 4 clocks after FRAME is asserted. Master Abort is treated as a fatal error by the Am79C976 controller. The Am79C976 controller will reset all CSR locations to their STOP_RESET values. The BCR and PCI configuration registers will not be cleared. Any on-going network transmission is terminated in an orderly sequence. The message will have the current FCS inverted and appended at the next byte boundary to guarantee that the receiving station will treat the transmission either as a runt or as a corrupted frame.

RMABORT (in the PCI Status register, bit 13) will be set to indicate that the Am79C976 controller has terminated its transaction with a master abort. In addition, SINT (CSR5, bit 11) will be set to 1. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to 1. This mechanism can be used to inform the driver of the system error. The host can read the PCI Status register to determine the exact cause of the interrupt. See Figure 20.

Downloaded from Arrow.com.

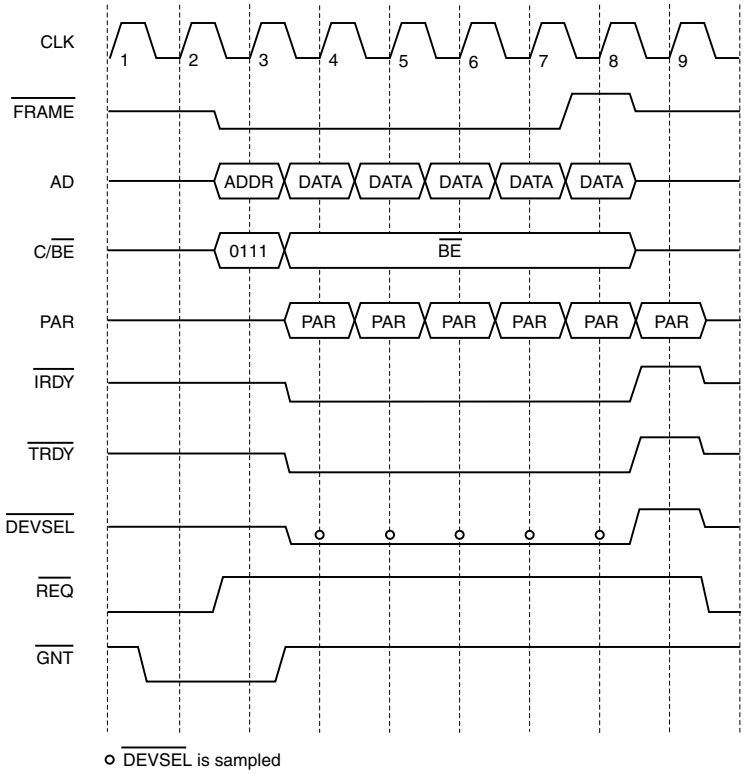


Figure 19. Preemption During Burst Transaction

Downloaded from Arrow.com.

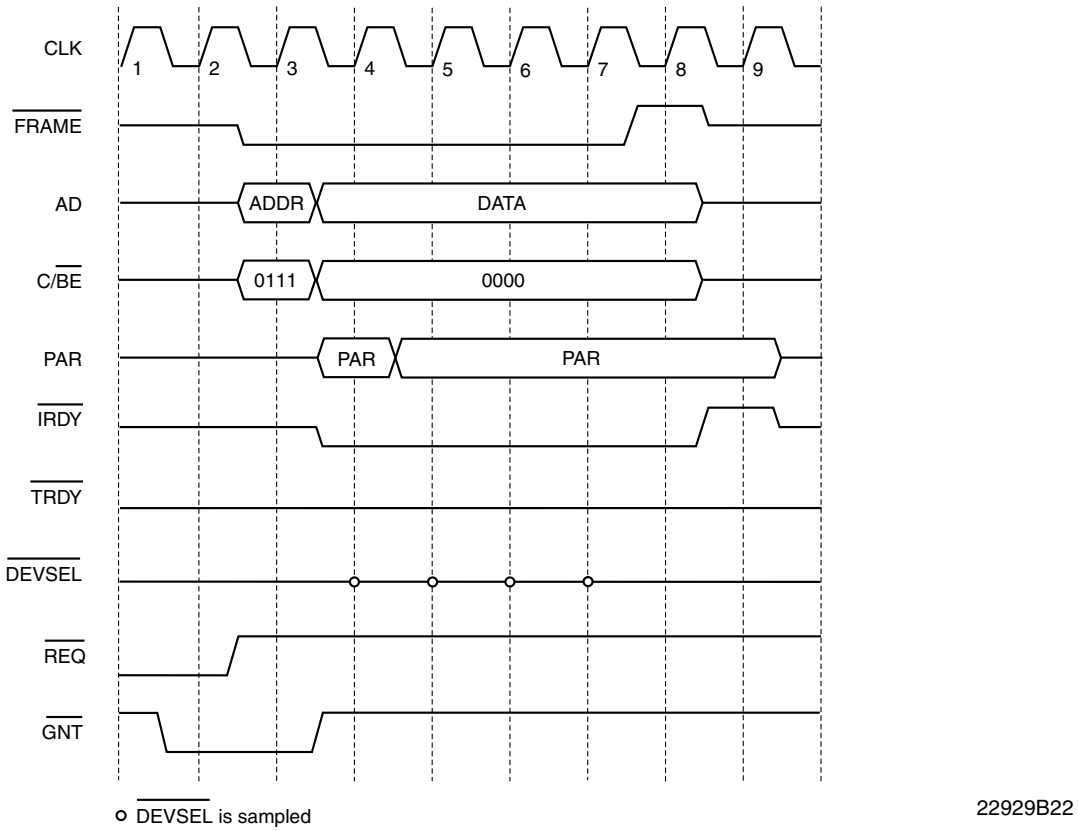


Figure 20. Master Abort

Parity Error Response

During every data phase of a DMA read operation, when the target indicates that the data is valid by asserting TRDY, the Am79C976 controller samples the AD[31:0], C/BE[3:0] and the PAR lines for a data parity error. When it detects a data parity error, the controller sets PERR (PCI Status register, bit 15) to 1. When reporting of that error is enabled by setting PERREN (PCI Command register, bit 6) to 1, the Am79C976 controller also drives the PERR signal low and sets DATAPERR (PCI Status register, bit 8) to 1. The assertion of PERR follows the corrupted data/byte enables by two clock cycles and PAR by one clock cycle.

Figure 21 shows a transaction that has a parity error in the data phase. The Am79C976 controller asserts PERR on clock 8, two clock cycles after data is valid. The data on clock 5 is not checked for parity, since on a read access PAR is only required to be valid one clock after the target has asserted TRDY. The Am79C976 controller then drives PERR high for one clock cycle, since PERR is a sustained tri-state signal.

During every data phase of a DMA write operation, the Am79C976 controller checks the PERR input to see if the target reports a parity error. When it sees the PERR input asserted, the controller sets PERR (PCI Status register, bit 15) to 1. When PERREN (PCI Command register, bit 6) is set to 1, the Am79C976 controller also sets DATAPERR (PCI Status register, bit 8) to 1.

Downloaded from Arrow.com.

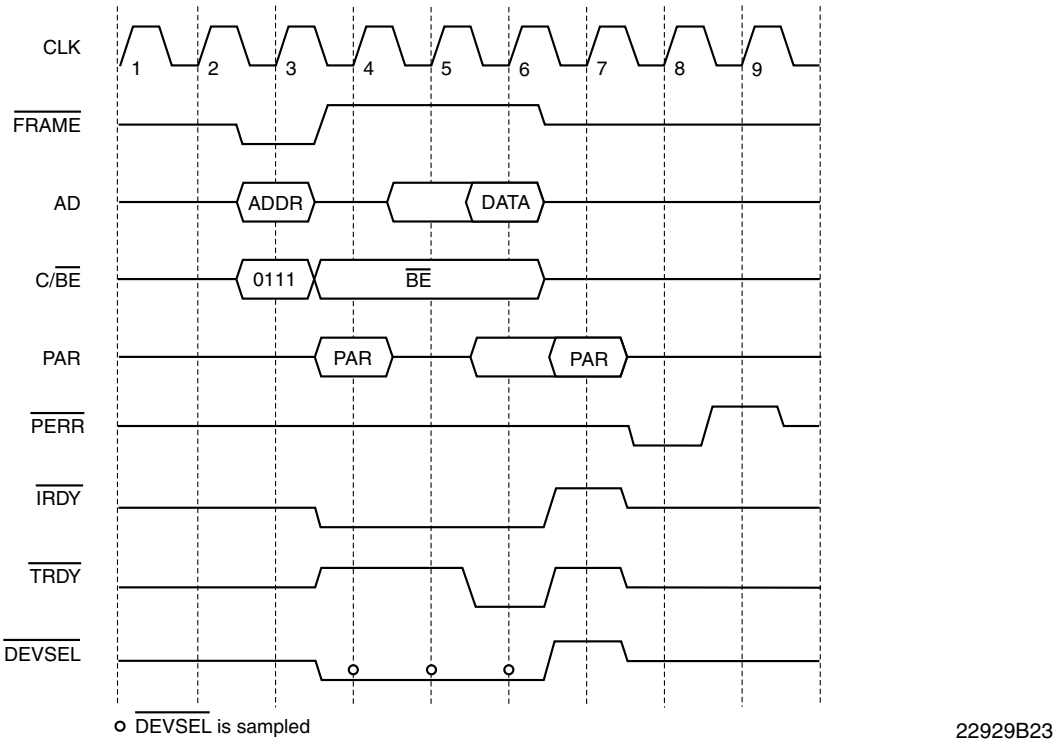


Figure 21. Master Cycle Data Parity Error Response

Whenever the Am79C976 controller is the current bus master and a data parity error occurs, SINT (CSR5, bit 11) will be set to 1. When SINT is set, INTA is asserted if the enable bit SINTE (CSR5, bit 10) is set to 1. This mechanism can be used to inform the driver of the system error. The host can read the PCI Status register to determine the exact cause of the interrupt. The setting of SINT due to a data parity error is not dependent on the setting of PERREN (PCI Command register, bit 6).

By default, a data parity error does not affect the state of the MAC engine. The Am79C976 controller treats the data in all bus master transfers that have a parity error as if nothing has happened. All network activity continues.

Initialization Block DMA Transfers

During execution of the Am79C976 controller bus master initialization procedure, the Am79C976 controller will use a burst transfer of seven Dwords to read the initialization block. AD[1:0] is 0 during the address phase indicating a linear burst order.

Descriptor DMA Transfers

During descriptor read accesses, the byte enable signals will indicate that all byte lanes are active. Should some of the bytes not be needed, then the Am79C976 controller will internally discard the extraneous information that was gathered during such a read.

The settings of SWSTYLE (BCR20, bits 7-0) affect the way the Am79C976 controller performs descriptor read operations.

Because of the order in which the descriptor data must be read or written when SWSTYLE is set to 0 or 2, all descriptor read operations are performed in non-burst mode. See Figure 22.

Downloaded from Arrow.com.

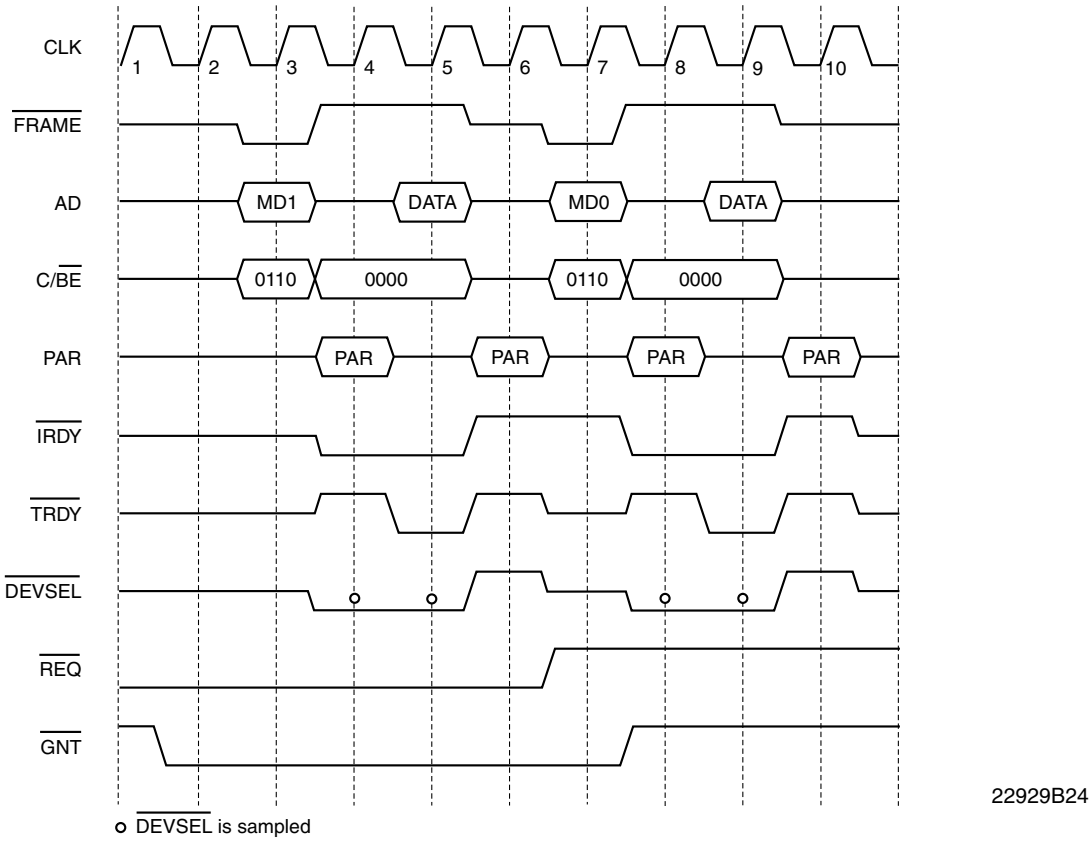


Figure 22. Descriptor Ring Read In Non-Burst Mode

When SWSTYLE is set to 3, 4, or 5 the descriptor entries are ordered to allow burst transfers, and the Am79C976 controller will perform all descriptor read operations in burst mode. The device may read more than one descriptor in a single burst. See Figure 23.

Downloaded from Arrow.com.

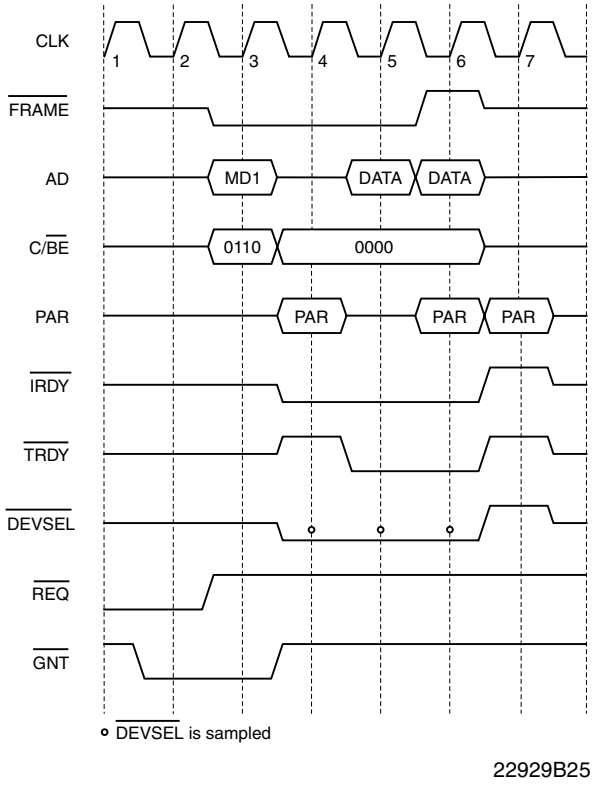


Figure 23. Descriptor Ring Read In Burst Mode

Table 4 shows the descriptor read sequence. During descriptor write accesses, only the byte lanes which need to be written are enabled.

The settings of SWSTYLE (BCR20, bits 7-0) affect the way the Am79C976 controller performs descriptor write operations.

When SWSTYLE is set to 0 or 2, all descriptor write operations are performed in non-burst mode.

When SWSTYLE is set to 3, 4, or 5, the descriptor entries are ordered to allow burst transfers. The Am79C976 controller will perform all descriptor write operations in burst mode. See Table 5 for the descriptor write sequence.

Table 4. Descriptor Read Sequence

SWSTYLE BCR20 [7:0]	AD Bus Sequence for Rx Descriptors	AD Bus Sequence for Tx Descriptors
0	Address = XXXX XX00h Turn around cycle Data Idle Address = XXXX XX04h Turn around cycle Data	Address = XXXX XX00h Turn around cycle Data Idle Address = XXXX XX04h Turn around cycle Data
2	Address = XXXX XX04h Turn around cycle Data Idle Address = XXXX XX00h Turn around cycle Data	Address = XXXX XX04h Turn around cycle Data Idle Address = XXXX XX00h Turn around cycle Data
3	Address = XXXX XX04h Turn around cycle Data Data	Address = XXXX XX04h Turn around cycle Data Data
4	Address = XXXX XX04h Turn around cycle Data Data	Address = XXXX XX00h Turn around cycle Data Data Data
5	Address = XXXX XX08h Turn around cycle Data Data Data	Address = XXXX XX00h Turn around cycle Data Data Data Data

Downloaded from Arrow.com.

Table 5. Descriptor Write Sequence

SWSTYLE BCR20[7:0]	AD Bus Sequence for Rx Descriptor	AD Bus Sequence for Tx Descriptor
0	Address = XXXX XX04h Data Idle Address = XXXX XX00h Data	Address = XXXX XX04h Data Idle Address = XXXX XX00h Data
2	Address = XXXX XX08h Data Idle Address = XXXX XX04h Data	Address = XXXX XX08h Data Idle Address = XXXX XX04h Data
3	Address = XXXX XX00h Data Data	Address = XXXX XX00h Data Data
4	Address = XXXX XX00h Data	Address = XXXX XX00h Data
5	Address = XXXX XX00h Data	Address = XXXX XX00h Data

Note: *Figure 24 assumes that the Am79C976 controller is programmed to use 32-bit software structures (SWSTYLE = 2, 3, 4, or 5). The byte enable signals for the second data transfer would be 0111b, if the device was programmed to use 16-bit software structures (SWSTYLE = 0).*

Downloaded from Arrow.com.

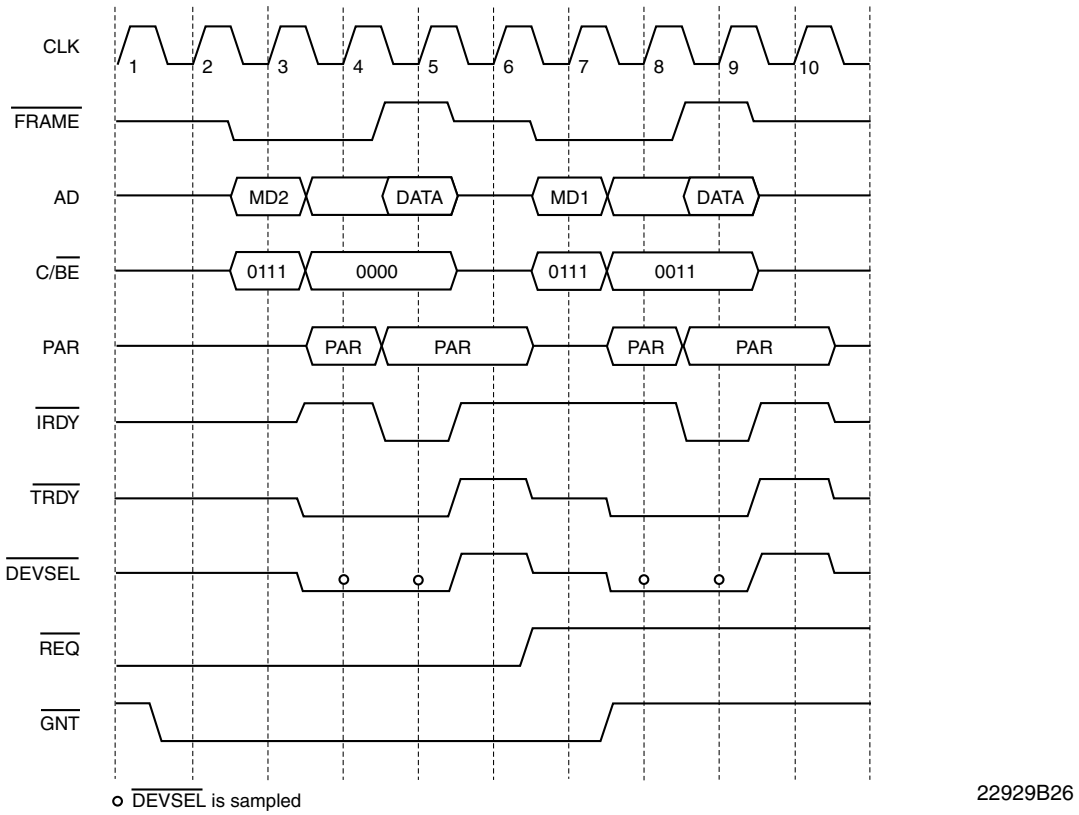


Figure 24. Descriptor Ring Write In Non-Burst Mode

Downloaded from Arrow.com.

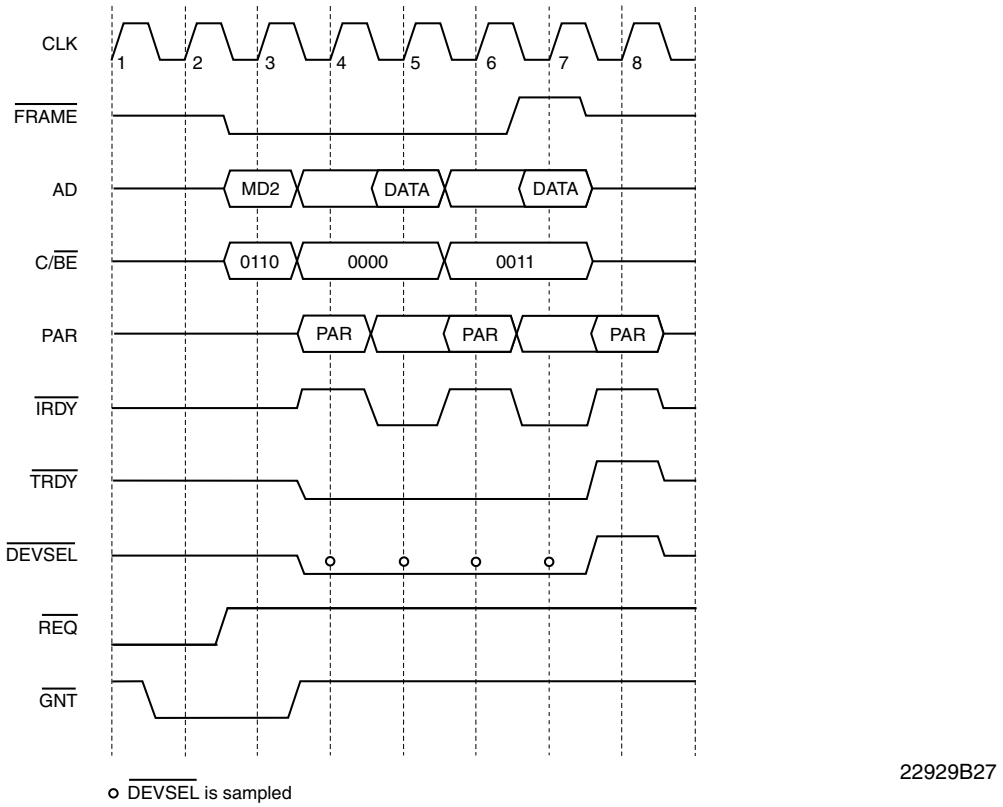


Figure 25. Descriptor Ring Write In Burst Mode

FIFO DMA Transfers

Am79C976 logic will determine when a FIFO DMA transfer is required. This transfer mode will be used for transfers of data to and from the Am79C976 FIFOs. Once the Am79C976 BIU has been granted bus mastership, it will perform a series of consecutive transfer cycles before relinquishing the bus. All transfers within the master cycle will be either read or write cycles, and all transfers will be transferred to contiguous, ascending addresses. Burst cycles are used whenever possible.

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

During FIFO DMA read operations, all byte lanes will always be active. The Am79C976 controller will internally discard unused bytes. During the first and the last data phases of a FIFO DMA burst write operation, one or more of the byte enable signals may be inactive. All other data phases will always write a complete DWord.

Figure 26 shows the beginning of a FIFO DMA write with the beginning of the buffer not aligned to a DWord boundary. The Am79C976 controller starts off by writing only three bytes during the first data phase. This operation aligns the address for all other data transfers to a 32-bit boundary so that the Am79C976 controller can continue bursting full DWords.

Downloaded from Arrow.com.

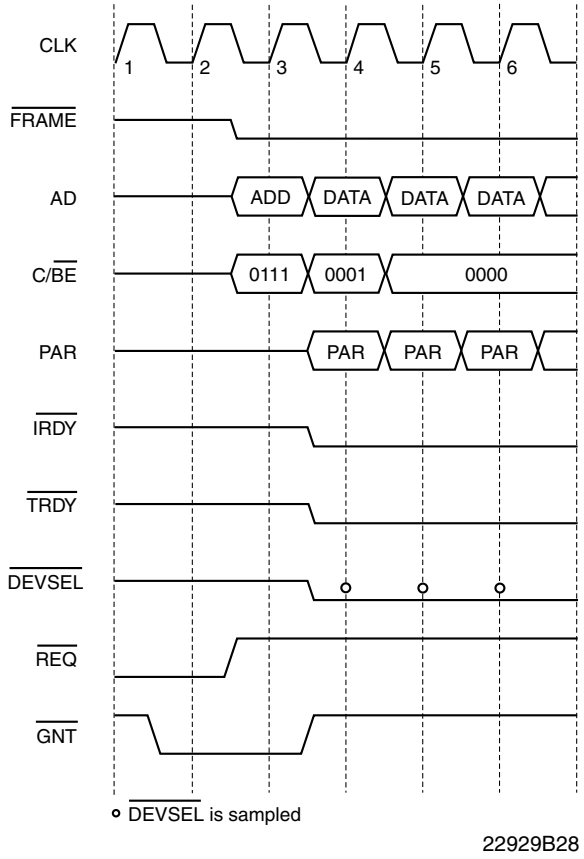


Figure 26. FIFO Burst Write At Start Of Unaligned Buffer

If a receive buffer does not end on a DWord boundary, the Am79C976 controller will perform a non-DWord write on the last transfer to the buffer. Figure 27 shows the final three FIFO DMA transfers to a receive buffer. Since there were only 9 bytes of space left in the receive buffer, the Am79C976 controller bursts three data phases. The first two data phases write a full DWord, the last one only writes a single byte.

Downloaded from Arrow.com.

Downloaded from Arrow.com.

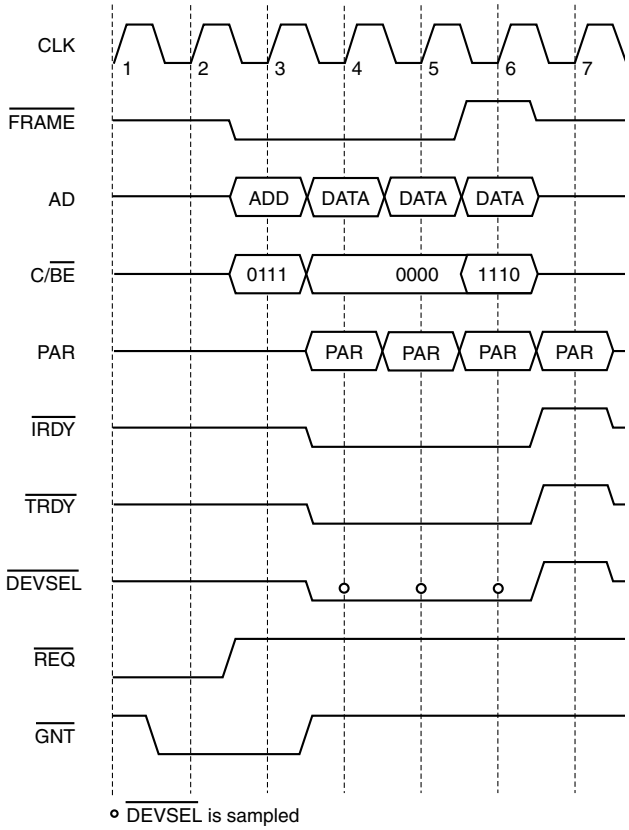


Figure 27. FIFO Burst Write At End Of Unaligned Buffer

Note that the Am79C976 controller will always perform a DWord transfer as long as it owns the buffer space, even when there are less than four bytes to write. For example, if there is only one byte left for the current receive frame, the Am79C976 controller will write a full DWord, containing the last byte of the receive frame in the least significant byte position (BSWP is cleared to 0, CSR3, bit 2). The content of the other three bytes is undefined. The message byte count in the receive descriptor always reflects the exact length of the received frame.

In the normal DMA mode (when the Burst Alignment bit = 0 and the Burst Limit register contents = 0) the Am79C976 controller will continue transferring FIFO data until the transmit FIFO is filled to its high threshold (for read transfers) or the receive FIFO is emptied to its low threshold (for write transfers), or until the Am79C976 controller is preempted and the PCI Latency Timer is expired. The host should use the values in the PCI MIN_GNT and MAX_LAT registers to determine the value for the PCI Latency Timer.

In the burst alignment mode (when the Burst Alignment bit = 1) if a burst transfer starts in the middle of a cache line, the transfer will stop at the first cache line boundary.

If the contents of the Burst Limit register are not zero, a burst transfer will end when the transfer has crossed the number of cache line boundaries equal to the contents of this register.

The exact number of total transfer cycles in the bus mastership period is dependent on all of the following variables: the settings of the FIFO watermarks, the conditions of the FIFOs, the latency of the system bus to the Am79C976 controller's bus request, and the speed of bus operation. The TRDY response time of the memory device will also affect the number of transfers, since the speed of the accesses will affect the state of the FIFO. The general rule is that the longer the Bus Grant latency, the slower the bus transfer operations; the slower the clock speed, the higher the transmit watermark; or the lower the receive watermark, the longer the total burst length will be.

When a FIFO DMA burst operation is preempted, the Am79C976 controller will not relinquish bus ownership until the PCI Latency Timer expires.

Descriptor Management Unit

The Descriptor Management Unit (DMU) implements the automatic initialization procedure and manages the descriptors and buffers.

Initialization

The Am79C976 controller is initialized by a combination of EEPROM register writes, direct register writes from the PCI bus and, for compatibility with older PCnet family products, DMA reads from an initialization block in memory. The registers that must be programmed depend on the features that are required in a particular application. See *USER ACCESSIBLE REGISTERS on page 113* for more details.

The format of the legacy initialization block depends on the programming of the SWSTYLE register, as described in the *Initialization Block* section.

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 ensure correct operation. Once the initialization block has been completely read in and internal registers have been updated, IDON will be set in CSR0, generating an interrupt (if IENA is set).

The Am79C976 controller obtains the start address of the initialization block from the contents of CSR1 (least significant 16 bits of address) and CSR2 (most significant 16 bits of address). The host must write CSR1 and CSR2 before setting the INIT bit. The initialization block contains the user defined conditions for Am79C976 operation, together with the base addresses and length information of the transmit and receive descriptor rings.

Downloaded from Arrow.com.

Re-Initialization

Earlier members of the PCnet family of controllers had to be re-initialized if the transmitter and/or the receiver were not turned on during the original initialization, and it was subsequently required to activate them, or if either section was shut off due to the detection of a memory error, transmitter underflow, or transmit buffer error condition. This restriction does not apply to the Am79C976 device. The memory error and transmit buffer error conditions cannot occur in the Am79C976 controller and the transmit underflow condition does not stop the Am79C976 controller's transmitter.

For compatibility with other PCnet family devices, re-initialization may be done via the initialization block or by setting the STOP bit in CSR0, followed by writing to CSR15, and then setting the STRT bit in CSR0. Note that this form of restart will not perform the same in the Am79C976 controller as in the C-LANCE device. In particular, setting the STRT bit causes the Am79C976 controller to reload the transmit and receive descriptor pointers with their respective base addresses. This means that the software must clear the descriptor OWN bits and reset its descriptor ring pointers before restarting the Am79C976 controller. The reload of descriptor base addresses is performed in the C-LANCE device only after initialization, so that a restart of the C-LANCE without initialization leaves the C-LANCE pointing at the same descriptor locations as before the restart.

Run and Suspend

Following reset, the transmitter and receiver of the Am79C976 controller are disabled, so no descriptor or data DMA activity will occur. The receiver will process incoming frames to detect address matches, which are counted in the RcvMissPkts register. No transmits will occur except that pause frames may be sent (see flow control section).

Setting the RUN bit in CMD0 (equivalent to setting STRT in CSR0) causes the Am79C976 controller to begin descriptor polling and normal transmit and receive activity. Clearing the RUN bit (equivalent to setting STOP in CSR0) causes the Am79C976 controller to halt all transmit, receive, and DMA transfer activities abruptly.

The Am79C976 controller offers suspend modes that allow stopping the device with orderly termination of all network activity. Transmit and receive are controlled separately.

Setting the RX_FAST_SPND bit in CMD0 suspends receiver activity after the current frame being received by the MAC is complete. If no frame is being received when RX_FAST_SPND is set, the receiver is suspended immediately. After the receiver is suspended, the RX_SUSPENDED bit in STAT0 is set and SPNDINT interrupt bit in INT0 is set. Receive data and descriptor DMA activity continues normally while the receiver is fast suspended.

Setting the RX_SPND bit in CMD0 suspends the receiver in the same way as RX_FAST_SPND, but the RX_SUSPENED bit and SPNDINT interrupt bit are only set after any frames in the receive FIFO have been completely transferred into system memory and the corresponding descriptors updated. No receive data or descriptor DMA activity will occur while the receiver is suspended.

When the receiver is suspended, no frames will be received into the receive FIFO, but frames will be checked for address match and the RcvMissPkts counter incremented appropriately, and frames will be checked for Magic Packet match if Magic Packet mode is enabled.

Setting the TX_FAST_SPND bit in CMD0 suspends transmitter activity after the current frame being transmitted by the MAC is complete. If no frame is being transmitted when TX_FAST_SPND is set, the transmitter is suspended immediately. After the transmitter is suspended, the TX_SUSPENDED bit in STAT0 is set and SPNDINT interrupt bit in INT0 is set. Transmit descriptor and data DMA activity continues normally while the transmitter is fast suspended.

Setting the TX_SPND bit in CMD0 suspends the transmitter in the same way as TX_FAST_SPND, but the TX_SUSPENDED bit and SPNDINT interrupt bit are only set after any frames in the transmit FIFO have been completely transmitted. No transmit descriptor or data DMA activity will occur while the transmitter is suspended.

When the transmitter is suspended, no frames will be transmitted except for flow control frames (see Flow Control section).

It is not meaningful to set both TX_SPND and TX_FAST_SPND at the same time, nor is it meaningful to set both RX_SPND and RX_FAST_SPND at the same time. Doing so will cause unpredictable results. However, transmit and receive are independent of each other, so one may be suspended or fast suspended while the other is running, suspended or fast suspended.

For compatibility with other PCnet family devices, setting the SPND bit in CSR5 with FASTSPNDE in CSR7 cleared is equivalent to setting both TX_SPND and RX_SPND and clearing SPND with FASTSPNDE cleared is equivalent to clearing both TX_SPND and RX_SPND. Similarly, setting SPND with FASTSPNDE set is equivalent to setting both TX_FAST_SPND and RX_FAST_SPND and clearing SPND with FASTSPNDE set is equivalent to clearing both TX_FAST_SPND and RX_FAST_SPND. While equivalent, these methods are not identical, so software

Downloaded from Arrow.com.

should not mix the CSR5/CSR7 method with the CMD0 method.

For compatibility with other PCnet family devices, after the SPND bit in CSR5 is set, it will read back a one only after the suspend operation is complete, that is, after both TX_SUSPENDED and RX_SUSPENDED in STAT0 have been set. It is recommended that when software polls this register that a delay be inserted between polls. Continuous polling will reduce the bus bandwidth available to the Am79C976 controller and will delay the completion of the suspend operation.

It is recommended that software use the SPNDINT interrupt to determine when the Am79C976 controller has suspended after one or more suspend bits have been set. This results in the least competition for the PCI bus and thus the shortest time from setting of a suspend bit until completion of the suspend operation.

Clearing the RUN bit in CMD0 will generate a pulse that will clear all the suspend command and status bits (TX_SPND, RX_SPND, TX_FAST_SPND and RX_FAST_SPND in CMD0, TX_SUSPENDED and RX_SUSPENDED in STAT0, SPND in CSR5 and DRX and DTX in CSR15). The RX_SPND or TX_SPND bits may then be set while RUN is cleared. When RUN is subsequently set, the suspend bit will remain set and the corresponding operation (transmit or receive) will be disabled. Since the suspend bit will be cleared when RUN is cleared, this must be done each time RUN is set. Since the suspend bits and RUN are in the same register (CMD0), the suspend bit may be set at the same time that RUN is set.

For compatibility with other PCnet family devices, setting the STOP bit in CSR0 will also clear the SPND bit in CSR5. While STOP is set, the DRX or DTX bits in CSR15 may be set. When the STRT bit in CSR0 is subsequently set, the corresponding operation will be disabled. Since the bits are all cleared when STOP is set, CSR15 must be written (either directly or indirectly via the DMA initialization) each time before STRT is set again.

The suspend bits in CMD0 and STAT0 are equivalent but not identical to the suspend bits in CSR5, CSR7 and CSR15. Software should use one set of bits or the other and not mix them. The SPNDINT bit in INT0 has no equivalent in the CSR registers, so this bit may be used to detect the completion of a suspend operation initiated by the SPND bit in CSR5.

Descriptor Management

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

Descriptor Rings

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

When SWSTYLE is at its default value of 0, the descriptor rings are backwards compatible with the Am79C90 C-LANCE and the Am79C96x PCnet-ISA family. The descriptor ring base addresses must be aligned to 8-byte boundaries. Each ring entry contains a subset of the three 32-bit transmit or receive message descriptors that are organized as four 16-bit structures (SSIZE32 (BCR20, bit 8) is set to 0). Note that even though the Am79C976 controller treats the descriptor entries as 16-bit structures, it will always perform 32-bit bus transfers to access the descriptor entries. The value of CSR2, bits 15-8, is used as the upper 8-bits for all memory addresses during bus master transfers.

When SWSTYLE is set to 2, 3, or 4, the descriptor ring base addresses must be aligned to 16-byte boundaries. Each ring entry is organized as three 32-bit message descriptors (SSIZE32 (BCR20, bit 8) is set to 1). The fourth DWord is reserved for user software purposes. When SWSTYLE is set to 3, 4, or 5, the order of the message descriptors is optimized to allow read and write access in burst mode.

When SWSTYLE is set to 5, the descriptor ring base addresses must be aligned to a 32-byte boundary. Each ring entry is organized as eight 32-bit message descriptors (SSIZE32 (BCR20, bit 8) is set to 1).

Descriptor ring lengths can be set up either by writing directly to the transmit and receive ring length registers (CSR76, CSR78) or by using the initialization block. If the initialization block is used to set up ring lengths, the ring lengths are restricted to powers of two that are less than or equal to 128 if SWSTYLE is 0 or 512 if SWSTYLE is 2 or 3. However, ring lengths of any size up to 65535 descriptors can be set up by writing directly to the transmit and receive ring length registers.

The initialization block can not be used if SWSTYLE is 4 or 5. The descriptor ring lengths must be initialized by writing directly to the appropriate registers.

Each ring entry contains the following information:

- The address of the actual message data buffer in user or host memory

Downloaded from Arrow.com.

- The length of the message buffer
- Status information indicating the condition of the buffer

To permit the queuing and de-queuing of message buffers, ownership of each buffer is allocated to either the Am79C976 controller or the host. The OWN bit within the descriptor status information, either TMD or RMD, is used for this purpose.

Setting the OWN to 1 signifies that the Am79C976 controller currently has ownership of this ring descriptor and its associated buffer. Only the owner is permitted to relinquish ownership or to write to any field in the descriptor entry. A device that is not the current owner of a descriptor entry cannot assume ownership or change any field in the entry. A device may, however, read from a descriptor that it does not currently own. Software should always read descriptor entries in sequential order. When software finds that the current descriptor is owned by the Am79C976 controller, then the software must not read ahead to the next descriptor. The software should wait at a descriptor it does not own until the Am79C976 controller sets OWN to 0 to release ownership to the software. (When LAPPEN (CSR3, bit 5) is set to 1, this rule is modified. See the LAPPEN description.

At initialization, the base address of the receive descriptor ring is written to CSR24 (lower 16 bits) and CSR25 (upper 16 bits), and the base address of the transmit descriptor ring is written to CSR30 and CSR31.

Figure 28 illustrates the relationship between the initialization base address, the initialization block, the receive and transmit descriptor ring base addresses, the receive and transmit descriptors, and the receive and transmit data buffers, when SSIZE32 is cleared to 0.

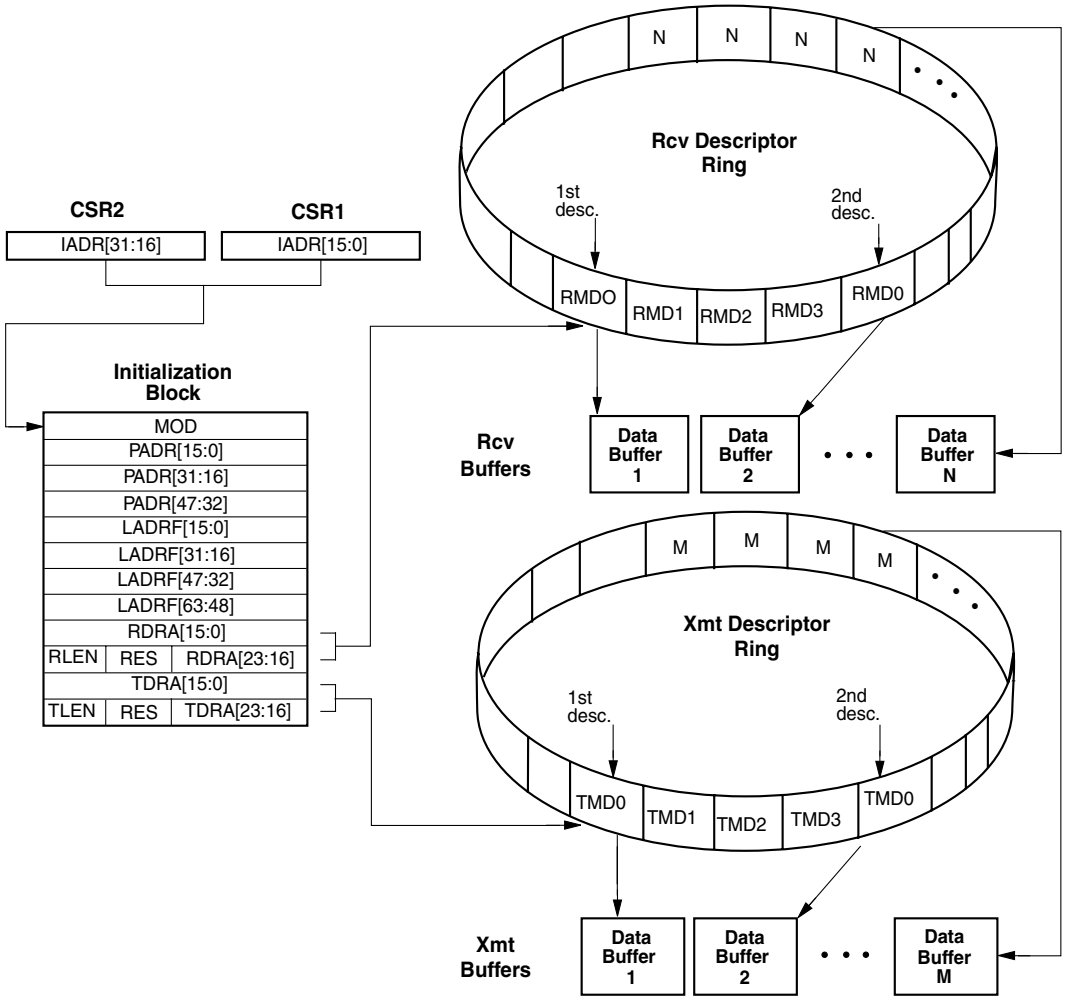


Figure 28. 16-Bit Software Model

Downloaded from Arrow.com.

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

Figure 29 illustrates the relationship between the initialization base address, the initialization block, the receive and transmit descriptor ring base addresses, the receive and transmit descriptors, and the receive and transmit data buffers, when SSIZE32 is set to 1.

.

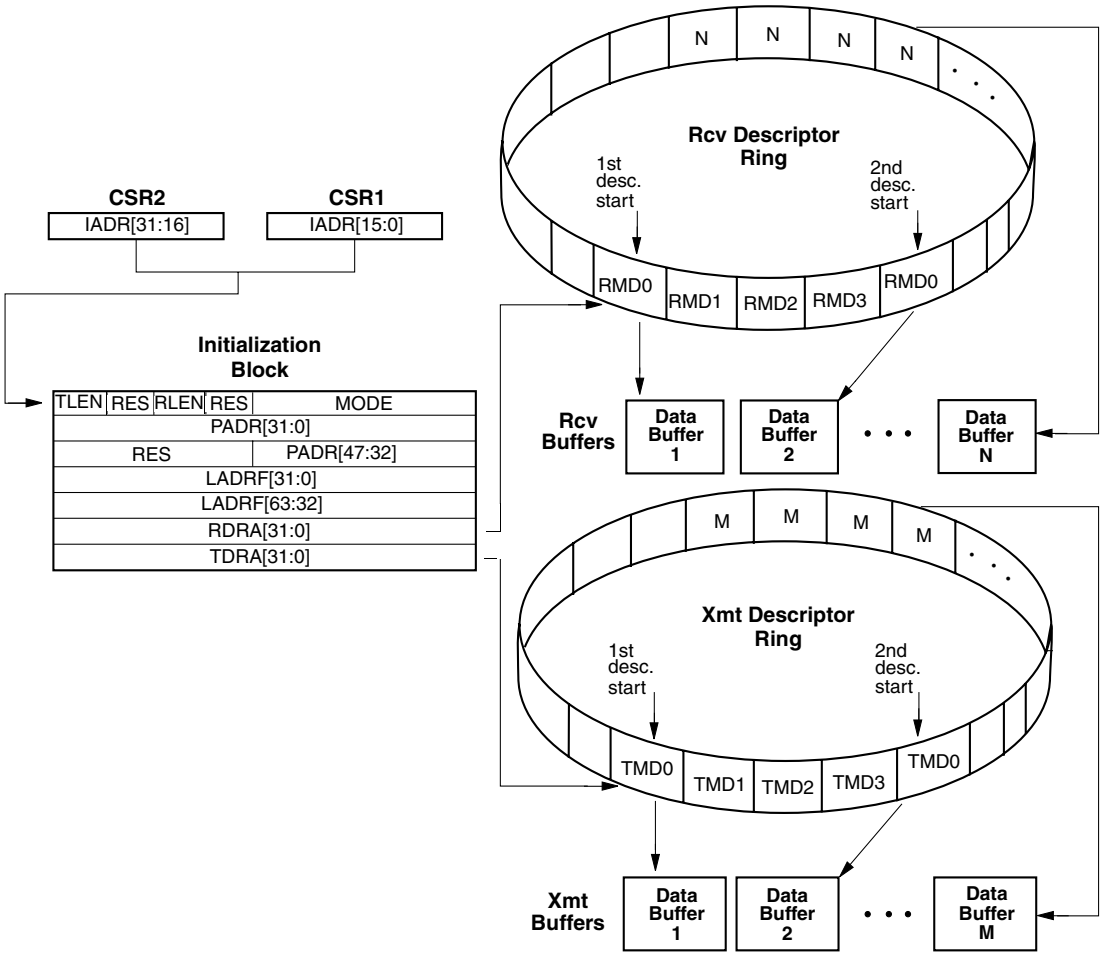


Figure 29. 32-Bit Software Model

Polling

If there is no network channel activity and there is no pre- or post-receive or pre- or post-transmit activity being performed by the Am79C976 controller, then the Am79C976 controller will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership. If the TXDPOLL bit in CSR4 is set, then the transmit polling function is disabled. The Descriptor Management Unit (DMU) is responsible for these operations.

The Am79C976 controller stores internally the information from two or more receive descriptors and two or more transmit descriptors. Polling operations depend on the ownership of the current and next receive and transmit descriptors.

When the poll time has elapsed, if the current receive descriptor is not owned by the Am79C976 controller or if the current receive descriptor is owned and the next receive descriptor is not owned, the unowned descriptor will be polled. Depending on the software style, more than one descriptor may be read in a burst.

If the TXDPOLL bit is not set and the poll time has elapsed, or whenever the TDMD bit is set, if the current transmit descriptor is not owned by the Am79C976 controller, it will be polled. Depending on the software style, more than one descriptor may be read in a burst.

If either transmit or receive or both are suspended or disabled due to the setting of TX_SPND, RX_SPND, SPND, DRX or DTX, the corresponding descriptors will not be polled. Polling is not affected by fast suspend.

Downloaded from Arrow.com.

Receive descriptor polling will continue even if transmit polling is disabled by setting TXDPOLL. If at least two receive descriptors are owned by the Am79C976 controller there will be no descriptor polling if there is no network activity.

The user may change the poll time value from the default value by modifying the value in the Transmit Polling Interval register (CSR47). The default value is 0000h, which corresponds to a polling interval of 65,536 X 3 ERCLK clock periods or 2.185 ms when ERCLK = 90 MHz.

When the Am79C976 controller is in the process of receiving a frame and it does not own the next descriptor or if it is in the process of transmitting a frame that does not end in the current descriptor and it does not own the next descriptor, it switches to the chain polling mode in which the polling interval is determined by the Chain Polling Interval register (CSR49). Thus, the device can be programmed to poll at a faster rate when it is about to run out of buffers.

Transmit Polling

If, after a transmit descriptor access, the Am79C976 controller finds that the OWN bit of that descriptor is not set, the Am79C976 controller resumes the poll time count and re-examines the same descriptor at the next expiration of the poll time count.

If the OWN bit of the descriptor is set, but the Start of Packet (STP) bit is not set, the Am79C976 controller will immediately request the bus in order to clear the OWN bit of this descriptor. After resetting the OWN bit of this descriptor, the Am79C976 controller will again immediately request the bus in order to access the next descriptor in the ring.

If the OWN bit is set and the buffer length is 0, the OWN bit will be cleared. The Am79C976 controller skips buffers with length of 0, which differs from the C-LANCE device, which interprets a buffer length of 0 to mean a 4096-byte buffer. For the Am79C976 device a zero length buffer is acceptable anywhere in the buffer chain.

If the OWN bit and STP are set, the DMA controller will start reading data from the current transmit buffer. If the next transmit descriptor is not already known to be owned, the Am79C976 controller will interleave a read of this descriptor into the sequence of data DMA operations.

If the next transmit descriptor has the OWN bit set, the Am79C976 controller will complete reading the data from the current transmit buffer, clear the OWN bit in the current descriptor and advance the internal ring pointer to make the next transmit descriptor the new current transmit descriptor.

The Am79C976 controller returns ownership of transmit descriptors to the software when the DMA transfer of data from system memory to the Am79C976 controller's memory is complete. This is different from older devices in the PCnet family, which will not return the last transmit descriptor of a frame (the one with ENP=1) until transmission of the frame is complete. The Am79C976 controller does not return any status information in the transmit descriptor, it will only write to the OWN bit to clear it.

Normally, the driver will set all the OWN bits of a frame in reverse order so that the Am79C976 controller will never encounter the situation where the current transmit descriptor has OWN=1 and ENP=0 and the next transmit descriptor has OWN=0. Older devices in the PCnet family treat this condition as a fatal error. The Am79C976 controller allows this mode of operation to permit DMA of the beginning of a frame before processing of the entire frame is complete. The number of bytes in the first buffer(s) should be less than the transmit start point or the REX_UFLO bit in CMD3 should be set.

When the Am79C976 controller encounters the condition of the current transmit descriptor's OWN=1 and ENP=0 and the next transmit descriptor's OWN=0, it enters the chain polling mode. In this mode, polling of the descriptor will occur at intervals determined by the Chain Polling Interval register (CSR49). Setting the TDMD bit will also cause a poll. Chain polling may be disabled by setting the CHDPOLL bit in CSR7 or CMD2. Note that this will also disable chain polling for receive descriptors.

If underflow occurs due to delays in setting the OWN bits or excessive bus latency, the transmitter will append an inverted FCS field to the frame and will increment the XmtUnderrunPkts counter. The frame may be retransmitted (if the REX_UFLO bit in CMD3 is set) or discarded.

If an error occurs in the transmission that causes the frame to be discarded (late collision, underflow or retry failure with the corresponding retry or retransmit option not enabled) before the entire frame has been transferred or if the current transmit descriptor has its KILL bit set, and if current transmit descriptor does not have its ENP bit set, the Am79C976 controller will skip over the rest of the frame which experienced the error. The Am79C976 controller will clear the OWN bit for all descriptors with OWN = 1 and STP = 0 and continue in like manner until a descriptor with OWN = 0 (no more transmit frames in the ring) or OWN = 1 and STP = 1 (the first buffer of a new frame) is reached.

At the end of any transmit operation, whether successful or with errors, the Am79C976 controller will always perform another polling operation, unless the next transmit descriptor is already known to be owned.

Downloaded from Arrow.com.

By default, whenever the DMA controller finishes copying a transmit frame from system memory, it sets the TINT bit of CSR0 to indicate that the buffers are no longer needed. This causes an interrupt signal if the IENA bit of CSR0 has been set and the TINTM bit of CSR3 is cleared.

The Am79C976 controller provides a mode to reduce the number of transmit interrupts. Setting LTINTEN (CSR5, bit14) to 1 suppresses interrupts for transmission of all but the last frame in a sequence.

Receive Polling

If the Am79C976 controller does not own both the current and the next receive descriptor, then the Am79C976 controller will continue to poll according to the polling sequence described in the *Transmit Polling* section. If the receive descriptor ring length is one, then there is no next descriptor to be polled.

If a poll operation has revealed that the current and the next receive descriptors belong to the Am79C976 controller, then additional poll accesses are not necessary. Future poll operations will not include receive descriptor accesses as long as the Am79C976 controller retains ownership of the current and the next receive descriptors.

When receive activity is present on the channel, the Am79C976 controller waits until the number of bytes specified in the RCV_PROTECT register (default 64) have been received. If the frame is accepted based on all active addressing schemes at that time, the DMU is notified that a frame has been received.

As receive buffers become available in system memory, the DMA controller will copy frame data from the receive FIFO into system memory. The Am79C976 controller will set the STP bit in the first descriptor of a frame. If the frame length exceeds the length of the current buffer, the Am79C976 controller will pass ownership back to the system by writing 0s to the OWN and ENP bits of the descriptor when the first buffer is full. This activity continues until the Am79C976 controller recognizes the completion of the frame (the last byte of this receive message has been removed from the FIFO). The Am79C976 controller will subsequently update the current receive descriptor with the frame status (message byte count, VLAN info, frame tag, error flags, etc.) and will set the ENP bit to 1. The Am79C976 controller will then advance the internal ring pointer to make the next receive descriptor the new current receive descriptor.

When the Am79C976 controller has receive data in the FIFO ready to write to system memory, either at the beginning of a new frame or in the middle of a frame that does not fit in the previous buffer, and it does not own the current receive descriptor, it will immediately poll it. If the OWN bit is still zero, polling of this descriptor will continue at a rate determined by the contents of the CHPOLLINT register (CSR49). Polling will occur immediately if the RDMD bit is set.

If the driver does not provide the Am79C976 controller with a descriptor in a timely fashion, the receive FIFO will eventually overflow. Subsequent frames will be discarded and the RcvMissPkts MIB counter will be incremented. Normal receive operation will resume when a descriptor is provided to the Am79C976 controller and sufficient data has been DMA'ed from the Am79C976 controller's receive FIFO into the system memory.

When the receive FIFO is empty and the Am79C976 device does not own two descriptors (current and next), the Receive Descriptor Ring is polled at an interval by the contents of the TXPOLLINT register (CSR47). When the Am79C976 device owns two descriptors, the Receive Descriptor Ring is not polled at all.

Look Ahead Packet Processing

Setting LAPPEN (CMD2, bit 2 or CSR3, bit 5) to a 1 modifies the way the controller processes receive descriptors. The Am79C976 controller will use the STP information to determine where it should begin writing a receive packet's data. Note that while in this mode, the Am79C976 controller can write intermediate packet data to buffers whose descriptors do not contain STP bits set to 1. Following the write to the last descriptor used by a packet, the Am79C976 controller will scan through the next descriptor entries to locate the next STP bit that is set to a 1. The Am79C976 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 containing STP = 1, then some descriptors/buffers may be skipped in the ring. While performing the search for the next STP bit that is set to 1, the Am79C976 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 Am79C976 controller ownership of the descriptor but also indicate STP = 0, then the Am79C976 controller will reset the OWN bit to 0 in these entries. If a scanned entry indicates host ownership with STP = 0, then the Am79C976 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 Am79C976 controller, then the Am79C976 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 Am79C976 controller, then the Am79C976 controller will stop advancing through the

Downloaded from Arrow.com.

ring entries, store the descriptor information that it has just read, and wait for the next receive to arrive.

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

Software Interrupt Timer

The Am79C976 controller is equipped with a software programmable free-running interrupt timer. The timer is constantly running and will generate an interrupt STINT (CSR 7, bit 11) when STINITE (CSR 7, bit 10) is set to 1. After generating the interrupt, the software timer will load the value stored in STVAL and restart. The timer value STVAL (BCR31, bits 15-0) is interpreted as an unsigned number with a resolution of 10.24µs. For instance, a value of 98 (62h) corresponds to 1.0 ms. The default value of STVAL is FFFFh which corresponds to 0.671 seconds. A write to STVAL restarts the timer with the new contents of STVAL.

Media Access Control

The Media Access Control (MAC) engine incorporates the essential protocol requirements for operation of an Ethernet/IEEE 802.3-compliant node and provides the interface between the FIFO subsystem and the MII.

This section describes operation of the MAC engine when operating in half-duplex mode. The operation of the device in full-duplex mode is described in the section titled *Full-Duplex Operation*.

The MAC engine is fully compliant to Section 4 of IEEE Std 802.3, 1998 Edition.

The MAC engine provides programmable enhanced features designed to minimize host supervision, bus utilization, and pre- or post-message processing. These features include the ability to disable retries after a collision, dynamic FCS generation on a frame-by-frame basis, automatic pad field insertion and deletion to enforce minimum frame size attributes, automatic retransmission without reloading the FIFO, and automatic deletion of collision fragments. The MAC also provides a mechanism for automatically inserting, deleting, and modifying IEEE 802.3ac VLAN tags.

The two primary attributes of the MAC engine are:

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

Transmit and Receive Message Data Encapsulation

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

Framing

The MAC engine will autonomously handle the construction of the transmit frame. Once the transmit FIFO has been filled to the predetermined threshold (set by XMTSP in CSR80) and access to the channel is currently permitted, the MAC engine will commence the 7-byte preamble sequence (10101010b, where first bit transmitted is a 1). The MAC engine will subsequently append the Start Frame Delimiter (SFD) byte (10101011b) followed by the serialized data from the transmit FIFO. Once the data has been transmitted, the MAC engine will append the FCS (most significant bit first) which was computed on the entire data portion of the frame. The data portion of the frame consists of destination address, source address, length/type, and frame data. The user is responsible for the correct ordering and content in each of these fields in the frame. The MAC does not use the content in the length/type field unless APAD_XMT (CSR4, bit 11) is set and the data portion of the frame is shorter than 60 bytes.

The receiver section of the MAC engine will detect the incoming preamble sequence when the RX_DV signal is activated by the external PHY. The MAC will discard the preamble and begin searching for the SFD except in the case of 100BASE-T4, for which there is no pre-

Downloaded from Arrow.com.

amble. In that case, the SFD will be the first two nibbles received. Once the SFD is detected, all subsequent nibbles are treated as part of the frame. The MAC engine will inspect the length field to ensure minimum frame size, strip unnecessary pad characters (if automatic pad stripping is enabled), and pass the remaining bytes through the receive FIFO to the host. If pad stripping is performed, the MAC engine will also strip the received FCS bytes, although normal FCS computation and checking will occur. Note that apart from pad stripping, the frame will be passed unmodified to the host. If the length field has a value of 46 or greater, all frame bytes including FCS will be passed unmodified to the receive buffer, regardless of the actual frame length.

If the frame terminates or suffers a collision before 64 bytes of information (after SFD) have been received, the MAC engine will automatically delete the frame from the receive FIFO, without host intervention. The Am79C976 controller has the ability to accept runt packets for diagnostic purposes and proprietary networks.

Destination Address Handling

The first 6 bytes of information after SFD will be interpreted as the destination address field. The MAC engine provides facilities for physical (unicast), logical (multicast), and broadcast address reception.

Error Detection

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

On completion of transmission, the MAC engine updates various counters that are described in the Statistics Counters section. The host CPU can read these counters at any time for network management purposes.

The MAC engine also attempts 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 sent with an invalid FCS, which will cause the receiver to reject the message.

The MAC engine can be programmed to try to transmit the same frame again after a FIFO underflow or excessive collision error.

The status of each receive message is available in the appropriate Receive Message Descriptor (RMD). All received frames are passed to the host regardless of any error.

During the reception, the FCS is generated on every nibble (including the dribbling bits) coming from the MII, although the internally saved FCS value is only updated on each byte boundary. The MAC engine will ignore an extra nibble at the end of a message, which corresponds to dribbling bits on the network medium. A framing or alignment error is reported to the user if an FCS error is detected and there is an extra nibble in the message. If there is an extra nibble but no FCS error, no framing error is reported.

Media Access Management

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

Medium Allocation

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

The ISO 8802-3 (IEEE/ANSI 802.3) standard allows an optional two-part deferral after a receive message.

See ANSI/IEEE Std 802.3-1993 Edition, 4.2.3.2.1:

Note: *It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the inter-Frame gap based on this indication, it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness, the following optional measures, as specified in 4.2.8, are recommended when InterFrame-SpacingPart1 is other than 0:*

1. *Upon completing a transmission, start timing the interrupted gap, as soon as transmitting and carrier sense are both false.*
2. *When timing an inter-frame gap following reception, reset the inter-frame gap timing if carrier sense becomes true during the first 2/3 of the inter-frame gap timing interval. During the final 1/3 of the interval, the timer shall not be reset to ensure fair access to the medium. An initial period shorter than 2/3 of the interval is permissible including 0.*

The MAC engine implements the optional receive two part deferral algorithm, with an InterFrameSpacing-

Downloaded from Arrow.com.

Part1 (IFS1) time of 60 bit times and an InterFrameSpacingPart 2 time of 36 bit times.

The Am79C976 controller will perform the two-part deferral algorithm as specified in Clause 4.2.8 of IEEE Std 802.3 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 96-bit InterFrameSpacing after the receive carrier is deasserted.

During the first part deferral (InterFrameSpacingPart1 - IFS1), the Am79C976 controller will defer any pending transmit frame and respond to the receive message. If carrier sense or collision is detected during the first part of the gap, the IPG counter will be cleared to 0 continuously until carrier sense and collision are both deasserted, at which point the IPG counter will resume the 96-bit time count once again. Once the IPG counter reaches the IFS1 count (60-bit times), the Am79C976 controller will not defer to a receive frame if a transmit frame is pending. Instead, when the IPG count reaches 96-bit times, the transmitter will start transmitting, which will cause a collision. The Am79C976 controller will complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

The Am79C976 controller allows the user to program both the IPG and the first part deferral (InterFrameSpacingPart1 - IFS1) through CSR125. The user can change the IPG value from its default of 96-bit times to compensate for delays through the external PHY device. Changing IFS1 will alter the period for which the Am79C976 MAC engine will defer to incoming receive frames.

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

This transmit two-part deferral algorithm is implemented as an option which can be disabled using the DXMT2PD bit in CSR3. When DXMT2PD is set to 1, the IFS1 register is ignored, and the value 0 is used for the Inter FrameSpacingPart1 parameter. However, the IPG value is still valid.

When the Am79C976 device operates in full-duplex mode, the IPG timer starts counting when TX_EN is de-asserted. CRS is ignored in full-duplex mode.

Signal Quality Error (SQE) Test

During the time period immediately after a transmission has been completed, an external transceiver operating in the 10 Mb/s half-duplex mode should generate an SQE Test signal on the COL pin within 0.6 µs to 1.6 µs after the transmission ceases. Therefore, when the Am79C976 controller is operating in half-duplex mode, the IPG counter ignores the COL signal during the first 40-bit times of the inter-packet gap. This 40-bit times is the time period in which the SQE Test message is expected.

The SQE Test was originally designed to check the integrity of the Collision Detection mechanism independently of the Transmit and Receive capabilities of the Physical Layer. However, MII-based PHY devices detect collisions by sensing receptions that occur during transmissions, a process that does not require a separate level-sensing collision detection mechanism. Collision detection is therefore dependent on the health of the receive channel. Since the Link Monitor function checks the health of the receive channel, the SQE test is not very useful for MII-based devices. Therefore, the Am79C976 device does not report or count SQE Test failures.

Collision Handling

Collision detection is performed and reported to the MAC engine via the COL input pin. Since the COL signal is not required to be synchronized with TX_CLK, the COL signal must be asserted for at least three TX_CLK cycles in order to be detected reliably.

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

The MAC engine will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of collision will cause the transmission to be rescheduled to a time determined by the random backoff algorithm. If a single retry was required, the XmtOneCollision counter will be incremented. If more than one retry was required, the XmtMultipleCollision counter will be incremented. If all 16 attempts experienced collisions, the XmtExcessiveCollision counter will be incremented. After an excessive collision error, if REX_RTRY (CMD3, bit 18) is cleared to 0, the transmit message will be flushed from the FIFO. If the REX_RTRY bit is set to 1, the transmitter will not flush the transmit message from the FIFO. Instead, it will clear the back-off logic and will restart the transmission process, treating the data in the FIFO as a new frame.

If retries have been disabled by setting the DRTY bit in CSR15, the MAC engine will abandon transmission of the frame on detection of the first collision. In this case, XmtExcessiveCollision counter will be incremented, 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 increment the XmtLateCollision counter. If RTRY_LCOL (CMD3, bit 16) is set to 1, the

Downloaded from Arrow.com.

retry logic treats late collisions just like normal collisions. However, if the RTRY_LCOL bit is cleared to 0, no retry attempt will be scheduled on detection of a late collision. In this case, the transmit message will be flushed from the FIFO.

The ISO 8802-3 (IEEE/ANSI 802.3) Standard requires use of a "truncated binary exponential backoff" algorithm, which provides a controlled pseudo random mechanism to enforce the collision backoff interval, before retransmission is attempted.

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

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

The Am79C976 controller provides an alternative algorithm, which suspends the counting of the slot time/IPG during the time that receive carrier sense is detected. This aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. It effectively accelerates the increase in the backoff time in busy networks and allows nodes not involved in the collision to access the channel, while the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time-out their slot time counters as normal.

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

Transmit Operation

The transmit operation and features of the Am79C976 controller are controlled by programmable options. The Am79C976 controller provides a large transmit FIFO to provide frame buffering for increased system latency, automatic retransmission with no FIFO reload, and automatic transmit padding.

Transmit Function Programming

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

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

Automatic pad field insertion is controlled by the APAD_XMT bit in CSR4.

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

REX_RTRY (CMD3, bit 18) and REX_UFLO (CMD3, bit 17) can be programmed to cause the transmitter to automatically restart the transmission process instead of discarding a frame that experiences an excessive collisions or underflow error. In this case the retransmission will not begin until the entire frame has been loaded into the transmit FIFO. The RTRY_LCOL bit (CMD3, bit 16) can be programmed either to drop a frame after a late collision or to treat late collisions just like normal collisions.

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

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

In order to ensure that collisions occurring within 512-bit times from the start of transmission (including preamble) will be automatically retried with no host intervention, the transmit FIFO ensures that data contained within the FIFO will not be overwritten until at least 64 bytes (512 bits) of preamble plus address, length, and data fields have been transmitted onto the network without encountering a collision. If the REX_RTRY bit or the REX_UFLO bit is set, the transmit data will not be overwritten until the frame has been either transmitted or discarded.

Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (excluding preamble). This allows the minimum frame size of 64 bytes (512 bits) for IEEE 802.3/Ethernet to be guaranteed with no software intervention from the host/controlling process. Setting the APAD_XMT bit in CSR4 enables the automatic padding feature. The pad is placed between the LLC data field and FCS field in the IEEE 802.3 frame. FCS is always added if the frame is padded, regardless of the state of DXMTFCS (CSR15, bit 3) or ADD_FCS

Downloaded from Arrow.com.

(TMD1, bit 29). The transmit frame will be padded by bytes with the value of 00H. The default value of APAD_XMT is 0 after H_RESET, which will disable automatic pad generation.

If automatic pad generation is disabled, the software is responsible for insuring that the minimum frame size requirement is met. The hardware can reliably transmit frames ranging in size from 16 to 65536 octets.

It is the responsibility of upper layer software to correctly define the actual length/type field contained in the message to correspond to the total number of LLC Data bytes encapsulated in the frame (length/type field as defined in the IEEE 802.3 standard). The length value contained in the message is not used by the Am79C976 controller to compute the actual number of pad bytes to be inserted. The Am79C976 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 Am79C976 controller will check to ensure that 544 bits have been transmitted. If not, pad bytes are added to extend the frame size to this value, and the FCS is then added. See Figure 30.

.

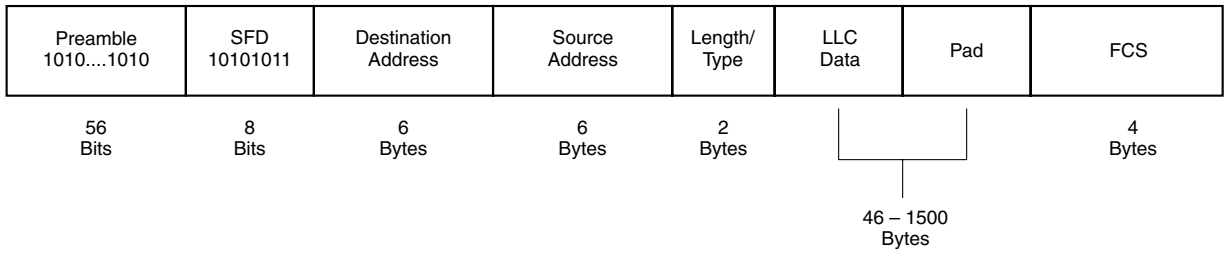


Figure 30. ISO 8802-3 (IEEE/ANSI 802.3) Data Frame

The 544 bit count is derived from the following:

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

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

Preamble/SFD + (Min Frame Size - FCS)

64 + (512-32) = 544 bits

A minimum length transmit frame from the Am79C976 controller, therefore, will be 576 bits, after the FCS is appended.

Transmit FCS Generation

Automatic generation and transmission of FCS for a transmit frame depends on the value of DXMTFCS (CSR15, bit 3). If DXMTFCS is cleared to 0, the transmitter will generate and append the FCS to the transmitted frame. If the transmitter modifies the frame data because of automatic padding or VLAN tag manipulation, the FCS will be appended by the Am79C976 controller regardless of the state of DXMTFCS or ADD_FCS (TMD1, bit 29). Note that the calculated FCS is transmitted most significant bit first. The default value of DXMTFCS is 0 after H_RESET.

When DXMTFCS is set to 1, the ADD_FCS (TMD1, bit 29) allows the automatic generation and transmission of FCS on a frame-by-frame basis. When DXMTFCS is set to 1, a valid FCS field is appended only to those frames whose TX descriptors have their ADD_FCS bits set to 1. If a frame is split into more than one buffer, the ADD_FCS bit is ignored in all descriptors except for the first.

Transmit Error Conditions

The Am79C976 transmitter detects the following error conditions and increments the appropriate error counters when they occur:

- Loss of carrier
- Late collision
- Transmit FIFO Underflow

Late collision errors can only occur when the device is operating in half-duplex mode. Loss of carrier and transmit FIFO underflow errors are possible when the device is operating in half- or full-duplex mode.

When an error occurs in the middle of a multi-buffer frame transmission, the appropriate error counter will be incremented, and the transmission will be aborted with an inverted FCS field appended to the frame. The OWN bit(s) in the current and subsequent descriptor(s) will be cleared until the STP (the next frame) is found.

If REX_UFLO (CMD3, bit 17) is set, the transmitter will not flush the frame data from the transmit FIFO after a

Downloaded from Arrow.com.

transmit FIFO underflow error occurs. Instead, it will wait until the entire frame has been copied into the transmit FIFO, and then it will restart the transmission process.

Loss of Carrier

The XmtLossCarrier counter is incremented if transmit is attempted when the LINK_STAT bit in the STAT0 register is 0.

Late Collision

A late collision will be detected when the device is operating in half-duplex mode and a collision condition occurs after one slot time (512 bit times) after the transmit process was initiated (first bit of preamble commenced). When it detects a late collision, the Am79C976 controller will increment the XmtLateCollision counter. If RTRY_LCOL (CMD3, bit 16) is cleared to 0, the controller will abandon the transmit process for that frame, and process the next transmit frame in the ring. If the RTRY_LCOL bit is set to 1, transmission attempts that incur late collisions will be retried up to a maximum of 16 attempts.

Transmit FIFO Underflow

An underflow error occurs when the transmitter runs out of data from the transmit FIFO in the middle of a transmission. When this happens, an inverted FCS is appended to the frame so that the intended receiver will ignore the frame, and the XmtUnderrunPkts counter is incremented. If REX_UFLO (CMD3, bit 17) is set to 1, the transmitter will then wait until the entire frame has been loaded into the transmit FIFO, and then it will restart the transmission of the same frame. If the REX_UFLO is cleared to 0, the transmitter will not attempt to retransmit the aborted frame.

Receive Operation

The receive operation and features of the Am79C976 controller are controlled by programmable options. The Am79C976 controller uses a large receive FIFO to provide frame buffering for increased system latency, automatic flushing of collision fragments (runt packets), automatic receive pad stripping, and a variety of address match options.

Receive Function Programming

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

The device can be programmed to accept all receive frames regardless of destination address by setting the PROM bit in CSR15. Acceptance of unicast and broadcast frames can be individually turned off by setting the DRCVPA or DRCVBC bits in CSR15. The Physical Address register (CSR12 to CSR14) stores the address that the Am79C976 controller compares to the destination address of the incoming frame for a unicast address match. The Logical Address Filter register (CSR8 to CSR11) serves as a hash filter for multicast address match.

The point at which the controller will start to transfer data from the receive FIFO to buffer memory is controlled by the RCVFW bits in CSR80. The default established during H_RESET is 01b, which sets the watermark flag at 64 bytes filled.

For test purposes, the Am79C976 controller can be programmed to accept runt packets of 12 bytes or larger by setting RPA in CSR124.

Address Matching

The Am79C976 controller supports three types of address matching: unicast, multicast, and broadcast. The normal address matching procedure can be modified by programming three bits in CSR15, the mode register (PROM, DRCVPA, and DRCVBC).

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

When a unicast frame arrives at the Am79C976 controller, the controller will accept the frame if the destination address field of the incoming frame exactly matches the 6-byte station address stored in the Physical Address registers (PADR, CSR12 to CSR14). The byte ordering is such that the first byte received from the network (after the SFD) must match the least significant byte of CSR12 (PADR[7:0]), and the sixth byte received must match the most significant byte of CSR14 (PADR[47:40]).

When DRCVPA (CSR15, bit 13) is set to 1, the Am79C976 controller will not accept unicast frames.

If the incoming frame is multicast, the Am79C976 controller performs a calculation on the contents of the destination address field to determine whether or not to accept the frame. This calculation is explained in the section that describes the Logical Address Filter (LADRF).

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

Although broadcast frames are classified as special multicast frames, they are treated differently by the Am79C976 controller hardware. Broadcast frames are

Downloaded from Arrow.com.

always accepted, except when DRCVBC (CSR15, bit 14) is set. DRCVBC overrides a logical address match. If DRCVBC is set to 1, broadcast frames are not accepted even if the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter.

None of the address filtering described above applies when the Am79C976 controller is operating in the promiscuous mode. In the promiscuous mode, all properly formed packets are received, regardless of the contents of their destination address fields. The promiscuous mode overrides the Disable Receive Broadcast bit (DRCVBC bit Am79C976 in the MODE register) and the Disable Receive Physical Address bit (DRCVPA, CSR15, bit 13).

The Am79C976 controller operates in promiscuous mode when PROM (CSR15, bit 15) is set.

In addition, the Am79C976 controller provides the External Address Detection Interface (EADI) to allow external address filtering. See the *External Address Detection Interface* section for further details.

The receive descriptor entry RMD1 contains three bits that indicate which method of address matching caused the Am79C976 controller to accept the frame. Note that these indicator bits are not available when the Am79C976 controller is programmed to use 16-bit structures for the descriptor entries (BCR20, bit 7-0, SWSTYLE is set to 0).

PAM (RMD1, bit 22) is set by the Am79C976 controller when it accepts the received frame due to a match of the frame's destination address with the content of the physical address register.

LAFM (RMD1, bit 21) is set by the Am79C976 controller when it accepts the received frame based on the value in the logical address filter register.

BAM (RMD1, bit 20) is set by the Am79C976 controller when it accepts the received frame because the frame's destination address is of the type 'Broadcast'. Only BAM, but not LAFM, will be set when a Broadcast frame is received, even if the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter.

When the Am79C976 controller operates in promiscuous mode and none of the three match bits is set, it is an indication that the Am79C976 controller only accepted the frame because it was in promiscuous mode.

When the Am79C976 controller is not programmed to be in promiscuous mode, but the EADI interface is used and when none of the three match bits is set, it is an indication that the Am79C976 controller only accepted the frame because it was not rejected by driving the $\overline{\text{EAR}}$ pin LOW during the receive protect time. The length of receive protect period can be programmed in the Receive Protect Register.

See Table 6 for receive address matches.

Table 6. Receive Address Match

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

Automatic Pad Stripping

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

The number of bytes to be stripped is calculated from the embedded length field (as defined in the ISO 8802-3 (IEEE/ANSI 802.3) definition) contained in the frame. The length indicates the actual number of LLC data bytes contained in the message. Any received frame which contains a length field less than 46 bytes will have the pad field stripped (if ASTRP_RCV is set). Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified.

Figure 31 shows the byte/bit ordering of the received length field for an IEEE 802.3-compatible frame format.

Downloaded from Arrow.com.

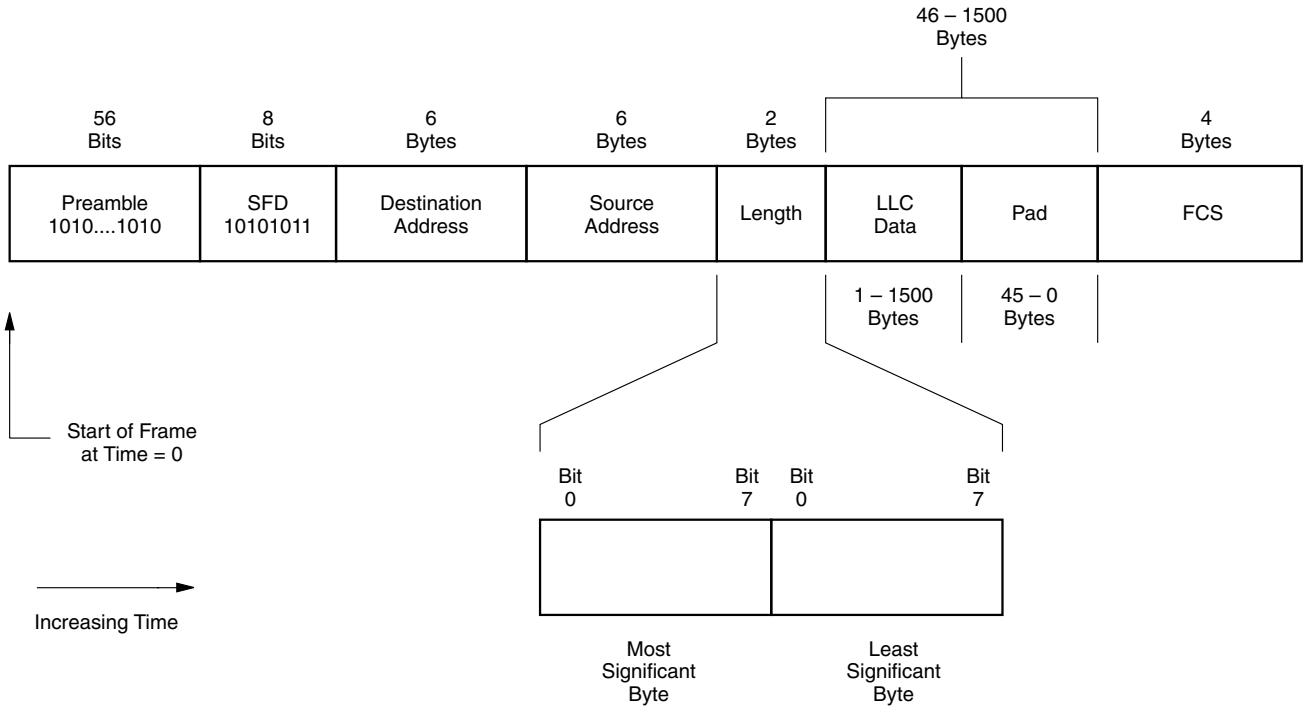


Figure 31. IEEE 802.3 Frame and Length Field Transmission Order

Since any valid Ethernet Type field value will always be greater than a normal IEEE 802.3 Length field (≥46), the Am79C976 controller will not attempt to strip valid Ethernet frames. *Note that for some network protocols, the value passed in the Ethernet Type and/or IEEE 802.3 Length field is not compliant with either standard and may cause problems if pad stripping is enabled*.

Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the Am79C976 controller. Note that if the Automatic Pad Stripping feature is enabled, the FCS for padded frames will be verified against the value computed for the incoming bit stream including pad characters, but the FCS value for a padded frame will not be passed to the host. If an FCS error is detected in any frame, the error will be reported in the CRC bit in the Receive Descriptor.

Receive Exception Conditions

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

Normal exception events are caused by collisions, which can distort and truncate received frames. Frames shorter than 64 bytes will, by default, be discarded. These fragments will be discarded 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.

There are two control bits that can be used to cause the MAC to override normal behavior and accept all frames that pass address match, regardless of the frame length. Setting the Runt Packet Accept (RPA) bit (CMD2, bit 19) causes the MAC to accept runt packets when the device is operating in either half- or full-duplex mode. Setting Full-Duplex Runt Packet Accept (FDRPA, CMD2, bit 20) causes the MAC to accept runt packets when the device is operating in full-duplex mode. (When the value of RPA is 1, runt packets are accepted regardless of the duplex mode or the value of FDRPA.) In either case, there is a minimum frame size of 16 bytes. Frames shorter than this may not be accepted, regardless of the value of RPA or FDRPA.

Abnormal network conditions include:

- FCS errors
- Framing errors
- Receiver overflow events

These error conditions are reported in the corresponding receive descriptors. The RcvFCSErrors, RcvAlignmentErrors, or RcvMissPkts counter is also incremented when one of these events occurs.

Statistics Counters

In order to provide network management information with minimum host CPU overhead, the Am79C976 device automatically maintains a set of 32-bit controller statistics counters. These counters are mapped di-

Downloaded from Arrow.com.

rectly into PCI memory space and can not be accessed indirectly through the RAP and RDP registers.

To simplify the use of software debuggers, the counter logic is designed so that the statistics counters can be accessed one, two, or four bytes at a time. When a portion of a statistics counter is read, the entire 32 bits of the counter is loaded into an internal holding register in a single atomic operation. When the CPU reads one or more bytes from the same counter, the data are read from the holding register rather than from the counter. The holding register is updated when either a read access is made to a different counter or a byte of the same counter is read for a second time.

Write access to statistics counters is provided for debugging purposes only. No holding register is used for write accesses. Writing one or two bytes at a time to a statistics counter while the network is active can cause unpredictable results.

The contents of the entire set of statistics counters can be cleared to zero by setting the INIT_MIB bit (CMD3, bit 25). The counters will be cleared within approximately 55 ERCLK cycles after the INIT_MIB bit is set.

Receive Statistics Counters

The receive statistics counters are defined and the Management Information Base (MIB) objects that they support are listed in Table 7.

For these counters, the definition of a valid frame depends on the state of the JUMBO and VSIZE bits (CMD3, bits 21 and 20) as follows:

If JUMBO = 1, valid frames are frames that are between 64 and 65536 bytes in length and have a correct FCS value. Frames longer than 65536 bytes may not be handled properly.

If JUMBO = 0 and VSIZE = 0, valid frames are frames that are between 64 and 1518 bytes in length and have a correct FCS value.

If JUMBO = 0 and VSIZE = 1, valid frames are frames that are between 64 and 1522 bytes in length and have a correct FCS value.

In Table 7, the Offset column gives the offset with respect to the value stored in the read-only MIB Offset register, which is located at offset 28h in the memory address space allocated to the Am79C976 device. The actual address of a particular counter is the sum of the following quantities:

- The contents of the PCI Memory-Mapped I/O Base Address Register.
- The contents of the MIB Offset Register.
- The offset given in Table 7.

Table 7. Receive Statistics Counters

Offset (hex)	Receive Counter Name	MIB Object Supported	Description of Counter/Comments
00	RcvMissPkts	RMON etherStatsDropEvents RMON etherHistoryDropEvents MIB-II ifInDiscards E-like dot3StatsInternalMacReceiveErrors	The number of times a receive packet was dropped due to lack of resources. This is the number of times a packet was dropped due to receive FIFO overflow. This count does not include undersize, oversize, misaligned or bad FCS packets.
04	RcvOctets	RMON etherStatsOctets RMON etherHistoryOctets MIB-II IfInOctets	The total number of octets of data received including octets from invalid frames. This does not include the preamble but does include the FCS bits. The RcvOctets counter is incremented whenever the receiver receives an octet.
08	RcvBroadCastPkts	RMON etherStatsBroadcastPkts RMON etherHistoryBroadcastPkts EXT-MIB-II ifInBroadcastPkts	The total number of valid frames received that are addressed to a broadcast address. This counter does not include errored broadcast packets or valid multicast packets.
0C	RcvMultiCastPkts	RMON etherStatsMulticastPkts RMON etherHistoryMulticastPkts EXT-MIB-II ifInMulticastPkts	The total number of valid frames received that are addressed to a multicast address. This counter does not include errored multicast packets or valid broadcast packets.

Downloaded from Arrow.com.

Offset (hex)	Receive Counter Name	MIB Object Supported	Description of Counter/Comments
10	RcvUndersizePkts	RMON etherStatsUndersizePkts RMON etherHistoryUndersizePkts	The total number of valid frames received that are less than 64 bytes long (including the FCS) and do not have any error. SFD must be received so that the FCS can be calculated.
14	RcvOversizePkts	RMON etherStatsOversizePkts RMON etherHistoryOversizePkts E-like MIB dot3StatsFrameTooLongs	The total number of packets received that are greater than 1518 (1522 when VLAN set) bytes long (including the FCS) and do not have any error. SFD must be received so that the FCS can be calculated.
18	RcvFragments	RMON etherStatsFragments RMON etherHistoryFragments	The number of packets received that are less than 64 bytes (not including the preamble or SFD) and have either an FCS error or an alignment error.
1C	RcvJabbers	RMON etherStatsJabbers RMON etherHistoryJabbers	The number of packets received that are greater than 1518 (1522 when VLAN set) bytes long and have either an FCS error or an alignment error.
20	RcvUnicastPkts	MIB-II ifInUcastPkts	The number of valid frames received that are not addressed to a multicast address or a broadcast address. This counter does not include errored unicast packets.
24	RcvAlignmentErrors	E-like MIB dot3StatsAlignmentErrors	The number of packets received that are between 64 and 1518 (1522 when VLAN set) bytes (excluding preamble/SFD but including FCS), inclusive, and have a bad FCS with non-integral number of bytes.
28	RcvFCSErrors	E-like MIB dot3StatsFCSErrors	The total number of packets received that are between 64 and 1518 (1522 when VLAN set) bytes (excluding preamble/ SFD but including FCS), inclusive, and have a bad FCS with an integral number of bytes. This counter will also count packets with a correct FCS if RX_ER occurs when valid carrier RX_DV is present.
2C	RcvGoodOctets	RMON hostInOctets RMON hostTimeInOctets	The total number of bytes received by a port. Bytes are 8-bit quantities received after the SFD. This does not include preamble or bytes from erroneous packets, but does include the FCS.
30	RcvMACCtrl	802.3x aMACControlFramesReceived	The total number of valid frames received with a lengthOrType field value equal to 8808h.
34	RcvFlowCtrl	802.3x aPAUSEMACCtrlFramesReceived	The total number of valid frames received with (1) a lengthOrType field value equal to 8808h and (2) an opcode equal to 1.
40	RcvPkts64Octets	RMON etherStatsPkts64Octets	The total number of packets (including error packets) that are 64 bytes long.
44	RcvPkts65to127Octets	RMON etherStatsPkts65to127Octets	The total number of packets (including error packets) that are 65 bytes to 127 bytes long, inclusive.

Downloaded from Arrow.com.

Offset (hex)	Receive Counter Name	MIB Object Supported	Description of Counter/Comments
48	RcvPkts128to255Octets	RMON etherStatsPkts128to255Octets	The total number of packets (including error packets) that are 128 bytes to 255 bytes long, inclusive.
4C	RcvPkts256to511Octets	RMON etherStatsPkts256to511Octets	The total number of packets (including error packets) that are 256 bytes to 511 bytes long, inclusive.
50	RcvPkts512to1023Octets	RMON etherStatsPkts512to1023Octets	The total number of packets (including error packets) that are 512 bytes to 1023 bytes long, inclusive
54	RcvPkts1024to1518Octets	RMON etherStatsPkts1024to1518Octets	The total number of packets (including error packets) that are 1024 bytes to 1518 (1522 when VLAN set) bytes long, inclusive.
58	RcvUnsupportedOpcodes	802.3x an UnsupportedOpcodesReceived	The total number of valid frames received with (1) a lengthOrType field value equal to 8808h and (2) an opcode not equal to 1.
5C	RcvSymbolErrors		The number of times when valid carrier (CRS) was present and there was at least one occurrence of an invalid data symbol (RX_ER). This counter is incremented only once per valid carrier event (once per frame), and if a collision is present, this counter must not be incremented.

Transmit Statistics Counters

Table 8 describes the statistics counters associated with the transmitter and lists the MIB objects that these counters support.

In this table the Offset column gives the offset with respect to the value stored in the read-only MIB Offset register, which is located at offset 28h in the memory address space allocated to the Am79C976 device. The actual address of a particular counter is the sum of the following quantities:

- The contents of the PCI Memory-Mapped I/O Base Address Register
- The contents of the MIB Offset Register
- The offset given in Table 8.

Table 8. Transmit Statistics Counters

Offset (hex)	Transmit Counter Name	MIB Object Supported	Description of Counter/Comments
60	XmtUnderrunPkts	RMON etherStatsDropEvents RMON etherHistoryDropEvents MIB-II ifOutDiscards E-like dot3StatsInternalMacTranmsitErrors	The number of times a packet was dropped due to transmit FIFO underrun.
64	XmtOctets	RMON etherStatsOctets RMON etherHistoryOctets RMON hostOutOctets RMON hostTimeOutOctets MIB-II IfOutOctets	The total number of octets of data transmitted. This does not include the preamble but does include the FCS bits. The XmtOctets counter is incremented whenever the transmitter transmits an octet.

Downloaded from Arrow.com.

Offset (hex)	Transmit Counter Name	MIB Object Supported	Description of Counter/Comments
68	XmtPackets	RMON etherStatsPkts RMON etherHistoryPkts RMON hostOutPkts RMON hostTimeOutPkts BRIDGE-MIB dot1dTpPortOutFrames	The number of packets transmitted. This does not include packets transmitted with errors (i.e., collision fragments and partial packets due to transmit FIFO under runs).
6C	XmtBroadCastPkts	RMON etherStatsBroadcastPkts RMON etherHistoryBroadcastPkts RMON hostOutBroadcastPkts RMON hostTimeOutBroadcastPkts EXT-MIB-II ifOutBroadcastPkts	The number of valid frames transmitted that are addressed to a broadcast address. This counter does not include errored broadcast packets or valid multicast packets.
70	XmtMultiCastPkts	RMON etherStatsMulticastPkts RMON etherHistoryMulticastPkts RMON hostOutMulticastPkts RMON hostTimeOutMulticastPkts EXT-MIB-II ifOutMulticastPkts	The number of valid frames transmitted that are addressed to a multicast address. This counter does not include errored multicast packets or valid broadcast packets.
74	XmtCollisions	RMON etherStatsCollisions RMON etherHistoryCollisions	The number of collisions that occur during transmission attempts. Collisions that occur while the device is not transmitting (i.e., receive collisions) are not identifiable and therefore not counted.
78	XmtUnicastPkts	MIB-II ifOutUcastPkts	The number of valid frames transmitted that are not addressed to a multicast or a broadcast address. This counter does not include errored unicast packets.
7C	XmtOneCollision	E-like dot3StatsSingleCollisionFrames	The number of packets successfully transmitted after experiencing one collision.
80	XmtMultipleCollision	E-like dot3StatsMultipleCollsionFrames	The number of packets successfully transmitted after experiencing more than one collision.
84	XmtDeferredTransmit	E-like dot3StatsDeferredTransmissions	The number of packets for which the first transmission attempt on the network is delayed because the medium is busy.
88	XmtLateCollision	E-like dot3StatsLateCollisions	The number of late collisions that occur. A late collision is defined as a collision that occurs more than 512 bit times after the transmission starts. The 512- bit interval is measured from the start of preamble.
8C	XmtExcessiveDefer		The number of excessive deferrals that occur. An excessive deferral occurs when a transmission is deferred for more than 3036 byte times in normal mode or 3044 byte times in VLAN mode.
90	XmtLossCarrier		The number of transmit attempts made when the LINK_STAT bit in the STAT0 register is 0.
94	XmtExcessiveCollision	E-like dot3StatsExcessiveCollisions	The number of packets that are not transmitted because the packet experienced 16 unsuccessful transmission attempts (the first attempt plus 15 retries).

Downloaded from Arrow.com.

Offset (hex)	Transmit Counter Name	MIB Object Supported	Description of Counter/Comments
98	XmtBackPressure		The total number of back pressure collisions generated.
9C	XmtFlowCtrl	PAUSEMACCtrlFramesTransmitted	The total number of PAUSE packets generated and transmitted by the controller hardware.
A0	XmtPkts64Octets	RMON etherStatsPkts64Octets	The total number of packets (excluding error packets) that are 64 bytes long.
A4	XmtPkts65to127Octets	RMON etherStatsPkts65to127Octets	The total number of packets transmitted (excluding error packets) that are 65 bytes to 127 bytes long, inclusive.
A8	XmtPkts128to255Octets	RMON etherStatsPkts128to255Octets	The total number of packets transmitted (excluding error packets) that are 128 bytes to 255 bytes long, inclusive.
AC	XmtPkts256to511Octets	RMON etherStatsPkts256to511Octets	The total number of packets transmitted (excluding error packets) that are 256 bytes to 511 bytes long, inclusive.
B0	XmtPkts512to1023Octets	RMON etherStatsPkts512to1023Octets	The total number of packets transmitted (excluding error packets) that are 512 bytes to 1023 bytes long, inclusive
B4	XmtPkts1024to1518Octets	RMON etherStatsPkts1024to1518Octets	The total number of packets transmitted (excluding error packets) that are 1024 bytes to 1518 (1522 when VLAN set) bytes long, inclusive.
B8	XmtOversizePkts		The total number of packets transmitted (excluding error packets) that are longer than 1518 (1522 when VLAN set) bytes.

VLAN Support

Virtual Bridged Local Area Network (VLAN) tags are defined in IEEE Std 802.3ac-1998. A VLAN tag is a 4-byte quantity that is inserted between the Source Address field and the Length/Type field of a basic 802.3 MAC frame. The VLAN tag consists of a Length/Type field that contains the value 8100h and a 16-bit Tag Control Information (TCI) field. The TCI field is further divided into a 3-bit User Priority field, a 1-bit Canonical Format Indicator (CFI), and a 12-bit VLAN Identifier.

A frame that has no VLAN tag is said to be untagged. A frame with a VLAN tag whose VLAN Identifier field contains the value 0 is said to be priority-tagged. A frame with a VLAN tag with a non-zero VLAN Identifier field is said to be VLAN-tagged.

The format of a VLAN-tagged frame is shown in Figure 32.

Downloaded from Arrow.com.

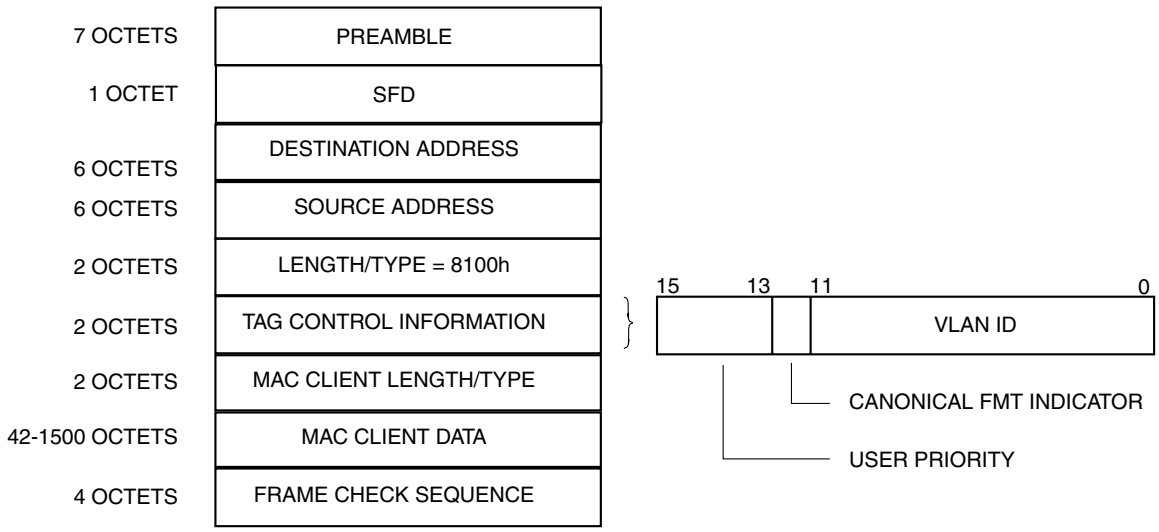


Figure 32. VLAN-Tagged Frame Format

The Am79C976 device includes several features that can simplify the processing of IEEE 802.3ac VLAN-tagged frames.

VLAN Frame Size

While the maximum frame size for IEEE 802.3 frames without VLAN tags is 1518 bytes, the maximum frame size for VLAN-tagged frames is 1522 bytes. The VLAN frame size bit (VSIZE, CMD3, bit 20) determines the maximum frame size. When VSIZE is set to 1 the maximum frame size is 1522 bytes. Otherwise, the maximum frame size is 1518 bytes.

The maximum frame size is used for determining when to increment the XmtOversizePkts, XmtPkts1024to1518Octets, XmtExcessiveDefer, RcvPkts1024to1518Octets, and RcvOversizePkts MIB counters.

Admit Only VLAN Frames Option

The Admit Only VLAN (VLONLY) bit in CMD3, bit19 can be programmed to reject any frame that is not VLAN-tagged. When VLONLY is set, untagged or priority-tagged frames will be flushed from the receive FIFO and will not be copied into system memory. Only frames with a Length/Type field equal to 8100h and a non-zero VLAN ID field will be received. The VLAN ID field consists of bits [11:0] of the 15th and 16th bytes of the frame.

VLAN Tags in Descriptors

When the SWSTYLE field in CSR58 contains the value 4 or 5, VLAN tag information can be passed between the host CPU and the network medium through Transmit or Receive Descriptors. The transmitter can be programmed to insert or delete a VLAN tag or to modify the TCI field of a VLAN tag. This feature allows VLAN software to control the VLAN tag of a frame without modifying data in transmit buffers. The receiver can determine whether a frame is untagged, priority-tagged, or VLAN-tagged, and it can copy the TCI field of the VLAN tag into the Receive Descriptor

The Tag Control Command (TCC) is a 2-bit field in the Transmit Descriptor that determines whether the transmitter will insert, delete, or modify a VLAN tag or transmit the data from the transmit buffers unaltered. The encoding of the TCC field is shown in Table 9.

If the transmitter adds, deletes, or modifies a VLAN tag, it will append a valid FCS field to the frame, regardless of the state of the Disable Transmit FCS (DXMTFCS) bit in CSR15.

When SWSTYLE is 4 or 5, the receiver examines each incoming frame and writes the frame's VLAN classification into the Tag Type (TT) field of the Receive Descriptor. If the frame contains a VLAN tag, the receiver will copy the TCI field of tag into the TCI field of the Receive Descriptor. The encoding of the TT field is shown in Table 10.

Downloaded from Arrow.com.

Table 9. VLAN Tag Control Command

TCC (TMD2[17:16])	Action
00	Transmit data in buffer unaltered
01	Delete Tag Header
10	Insert Tag Header containing TCI field from descriptor.
11	Replace TCI field from buffer with TCI data from descriptor.

Table 10. VLAN Tag Type

TT (RMD1[19:18])	Description
00	Reserved
01	Frame is untagged
10	Frame is priority-tagged
11	Frame is VLAN-tagged

Loopback Operation

Loopback is a mode of operation intended for system diagnostics. In this mode, the transmitter and receiver are both operating at the same time so that the controller receives its own transmissions. The controller provides two basic types of loopback. In internal loopback mode, the transmitted data is looped back to the receiver inside the controller. 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.

The external loopback through the MII requires a two-step operation. The external PHY must be placed into a loop-back mode by writing to the PHY Access Register. Then the Am79C976 controller must be placed into an external loopback mode by setting EXLOOP (CMD2, bit 3).

The internal loopback through the MII is controlled by INLOOP (CMD2, bit 4). When set to 1, this bit will cause the internal portion of the MII data port to loopback on itself. The MII management port (MDC, MDIO) is unaffected by the INLOOP bit.

The internal MII interface is mapped in the following way:

- The TXD[3:0] nibble data path is looped back onto the RXD[3:0] nibble data path
- TX_CLK is looped back as RX_CLK
- TX_EN is looped back as RX_DV
- TX_EN is OR'd with the CRS pin and is looped back as CRS
- The COL input is driven low
- TX_ER is not driven by the Am79C976 and therefore not looped back.

During the internal loopback, the TX_EN and TXD pins will be active. Internal loopback should not be used on a live network because collisions will not be handled correctly. The wire should be disconnected or the PHY isolated before using internal loopback.

Miscellaneous Loopback Features

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

Runt Packet Accept is internally enabled regardless of the state of the RPA bit in CSR124 when any loopback mode is invoked. This is for backwards compatibility with the C-LANCE (Am79C90) software.

The C-LANCE controller and the half-duplex members of the PCnet family of devices place certain restrictions on FCS generation and checking, and on testing multicast address detection. Since the Am79C976 controller has two FCS generators, these restrictions do not apply to the Am79C976 controller. On receive, the Am79C976 controller provides true FCS status. The descriptor for a frame with an FCS error will have the CRC bit (RMD1, bit 27) set to 1. The FCS generator on the transmit side can still be disabled by setting DXMTFCS (CSR15, bit 3) to 1.

In internal loopback operation, the Am79C976 controller provides a special mode to test the collision logic. When FCOLL (CSR15, bit 4) is set to 1, a collision is forced during every transmission attempt. This will result in a Retry error.

Full-Duplex Operation

The Am79C976 controller supports full-duplex operation on both network interfaces. Full-duplex operation allows simultaneous transmit and receive activity on the TXD[3:0] and RXD[3:0] pins of the MII port. Full-duplex operation is enabled by the FDEN bit located in BCR9 for all ports. Full-duplex operation is also enabled through Auto-Negotiation when DISPM (BCR 32, bit 7) is not enabled on the MII port and the ASEL bit is set, and both the external PHY and its link partner are capable of Auto-Negotiation and full-duplex operation.

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

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

- Changes to the Transmit Deferral mechanism:

Downloaded from Arrow.com.

— Transmission is not deferred while receive is active.

— The IPG counter which governs transmit deferral during the IPG between back-to-back transmits is started when transmit activity for the first packet ends, instead of when transmit and carrier activity ends.

- The collision indication input to the MAC engine is ignored.

Full-Duplex Link Status LED Support

The Am79C976 controller provides bits in each of the LED Status registers (BCR4, BCR5, BCR6, BCR7) to display the Full-Duplex Link Status. If the FDLSE bit (bit 8) is set, a value of 1 will be sent to the associated LEDOUT bit when in Full-Duplex.

Media Independent Interface

The Am79C976 controller fully supports the MII according to the IEEE 802.3 standard. This Reconciliation Sublayer interface allows a variety of PHYs (100BASE-TX, 100BASE-FX, 100BASE-T4, 100BASE-T2, 10BASE-T, etc.) to be attached to the Am79C976 MAC engine without future upgrade problems. The MII interface is a 4-bit (nibble) wide data path interface that runs at 25 MHz for 100-Mbps networks or 2.5 MHz for 10-Mbps networks. The interface consists of two independent data paths, receive (RXD(3:0)) and transmit (TXD(3:0)), control signals for each data path (RX_ER, RX_DV, TX_EN), network status signals (COL, CRS), clocks (RX_CLK, TX_CLK) for each data path, and a two-wire management interface (MDC and MDIO). See Figure 33.

The transmit and receive paths in the Am79C976 controller's MAC are independent. The TX_CLK and RX_CLK need not run at the same frequency. TX_CLK can slow down or stop without affecting receive and vice versa. It is only necessary to respect the minimum clock high and low time specifications when switching TX_CLK or RX_CLK. This facilitates operation with PHYs that use MII signaling but do not adhere to 802.3 MII specifications.

MII Transmit Interface

The MII transmit clock is generated by the external PHY and is sent to the Am79C976 controller on the TX_CLK input pin. The clock can run at 25 MHz or 2.5 MHz, depending on the speed of the network to which the external PHY is attached. The data is a nibble-wide (4 bits) data path, TXD(3:0), from the Am79C976 controller to the external PHY and is synchronous with the rising edge of TX_CLK. The transmit process starts when the Am79C976 controller asserts TX_EN, which indicates to the external PHY that the data on TXD(3:0) is valid.

IEEE Std 802.3 provides a mechanism for signaling unrecoverable errors through the MII to the external PHY with the TX_ER output pin. The external PHY will respond to this error by generating a TX coding error on the current transmitted frame. The Am79C976 controller does not use this method of signaling errors on the transmit side. Instead if the Am79C976 controller detects a transmit error, it will invert the FCS to generate an invalid FCS. Since the Am79C976 controller does not implement the TX_ER function, the TX_ER pin on the external PHY device should be connected to VSS.

MII Receive Interface

The MII receive clock is also generated by the external PHY and is sent to the Am79C976 controller on the RX_CLK input pin. The clock will be the same frequency as the TX_CLK but will be out of phase and can run at 25 MHz or 2.5 MHz, depending on the speed of the network to which the external PHY is attached.

Downloaded from Arrow.com.

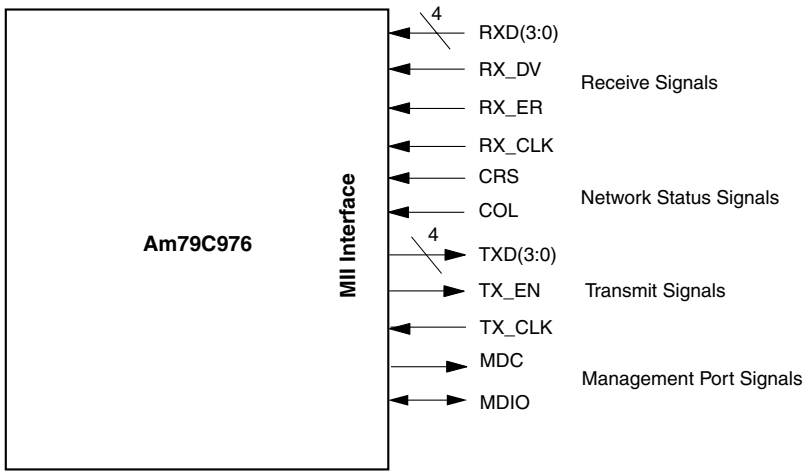


Figure 33. Media Independent Interface

The receive process starts when RX_DV is asserted. RX_DV must remain asserted until the end of the receive frame. If the external PHY device detects errors in the currently received frame, it asserts the RX_ER signal. RX_ER can be used to signal special conditions *out of band* when RX_DV is not asserted. Two defined out-of-band conditions for this are the 100BASE-TX signaling of *bad* Start of Frame Delimiter and the 100BASE-T4 indication of illegal code group before the receiver has synchronized with the incoming data. The Am79C976 controller will not respond to these conditions. All *out of band* conditions are currently treated as NULL events. Certain *in-band* non-IEEE 802.3-compliant flow control sequences may cause erratic behavior for the Am79C976 controller. Consult the switch/bridge/router/hub manual to disable the *in-band* flow control sequences if they are being used.

MII Network Status Interface

The MII also provides the CRS (Carrier Sense) and COL (Collision Sense) signals that are required for IEEE 802.3 operation. Carrier Sense is used to detect non-idle activity on the network for the purpose of interframe spacing timing in half-duplex mode. Collision Sense is used to indicate that simultaneous transmission has occurred in a half-duplex network.

MII Management Interface

The MII provides a two-wire management interface so that the Am79C976 controller can control external PHY devices and receive status from them.

The Am79C976 controller offers direct hardware support of the external PHY device without software intervention. The device automatically uses the MII Management Interface to read auto-negotiation information from the external PHY device and configures the MAC accordingly. The controller also provides the host CPU indirect access to the external PHY through the MII Control, Address, and Data registers (BCR32, 33, 34).

With software support the Am79C976 controller can support up to 31 external PHYs attached to the MII Management Interface.

Two independent state machines use the MII Management Interface to poll external PHY devices: the Network Port Manager and the Auto-poll State Machine. The Network Port Manager coordinates the auto-negotiation process, while the Auto-poll State Machine interrupts the host CPU when it detects changes in user-selected PHY registers.

The Network Port Manager sends a management frame to the default PHY about once every 900 ms to determine auto-negotiation results and the current link status. The Network Port Manager uses the auto-negotiation results to set the MAC's speed, duplex mode, and flow control ability. Changes detected by the Network Port Manager affect the operation of the MAC and MIB counters. For example, if link failure is detected, the transmitter will increment the XmtLossCarrier counter each time it attempts to transmit a frame.

The Auto-poll State Machine periodically sends management frames to poll the status register of the default PHY device plus up to 5 user-selected PHY registers and interrupts the host processor if it detects a change in any of these registers. The Auto-poll State Machine does not change the state of the MAC engine.

MII Management Frames

The format of an MII Management Frame is defined in Clause 22 of IEEE Std 802.3. The start of an MII Man-

Downloaded from Arrow.com.

agement Frame is a preamble of 32 ones that guarantees that all of the external PHYs are synchronized on the same interface. (See Figure 34.) Loss of synchronization is possible due to the *hot-plugging* capability of the exposed MII. The preamble can be suppressed as described below if the external PHY is designed to accept frames with no preamble.

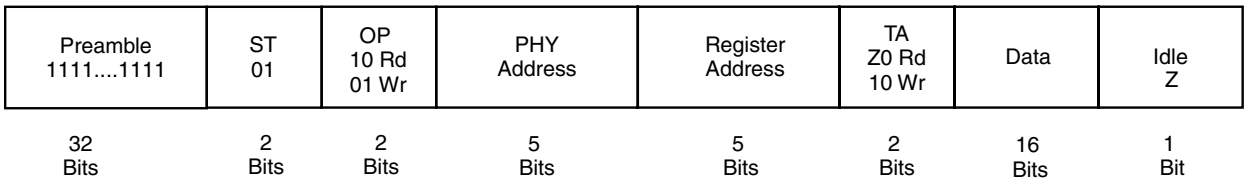


Figure 34. Frame Format at the MII Interface Connection

The preamble (if present) is followed by a start field (ST) and an operation field (OP). The operation field (OP) indicates whether the Am79C976 controller is initiating a read or write operation. This field is followed by the external PHY address (PHYAD) and the register address (REGAD). The PHY address of 1Fh is reserved and should not be used.

The register address field is followed by a bus turnaround field. During a read operation, the bus turnaround field is used to determine if the external PHY is responding correctly to the read request or not. The Am79C976 controller will tri-state the MDIO for both MDC cycles.

During the second cycle of a read operation, if the external PHY is synchronized to the Am79C976 controller, the external PHY will drive a 0. If the external PHY does not drive a 0, the Am79C976 controller will signal a MREINT (CSR7, bit 9) interrupt, if MREINTE (CSR7, bit 8) is set to a 1. This interrupt indicates that the Am79C976 controller had an MII management frame read error and that the data read is not valid.

During a write access the Am79C976 controller drives a 1 for the first bit time of the turnaround field and a 0 for the second bit time.

After the Turn Around field comes the data field. For a write access the Am79C976 controller fills this field with data to be written to the PHY device. For a read access the external PHY device fills this field with data from the selected register.

The last field of the MII Management Frame is an IDLE field that is necessary to give ample time for drivers to turn off before the next access.

MII management frames transmitted through the MDIO pin are synchronized with the rising edge of the Management Data Clock (MDC). The Am79C976 controller will drive the MDC to 0 and tri-state the MDIO anytime the MII Management Port is not active.

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

Host CPU Access to External PHY

The host CPU can indirectly read and write external PHY registers through the PHY Access Register or, for compatibility with other PCnet family devices, through BCR33 and BCR34.

PHY register accesses must be done using either BCR33 and BCR34 registers or the PHY_ACCESS register. Do not use both methods. If BCR33 and BCR34 are used, the PHY_ACCESS register should be at the default value of all zeros.

To write to a PHY register the host CPU puts the register data into the PHY_DATA field of the PHY Access Register, specifies the address of the external PHY device in the PHY_ADDR field and the PHY register number in the PHY_REG_ADDR field, and sets the PHY_WR_CMD bit.

The Am79C976 device provides two types of read access to external PHY registers, blocking and non-blocking. If a blocking read access is used, the device will generate PCI disconnect/retry cycles if the host CPU attempts to read the PHY Access Register while the MII Management Frame is being processed. If a

Downloaded from Arrow.com.

non-blocking read is used, the PHY Access Register can be read at any time, and the PHY_CMD_DONE bit in that register indicates whether or not PHY_DATA field contains valid data.

To generate a non-blocking read from a PHY register the host CPU specifies the address of the external PHY device in the PHY_ADDR field and the PHY register number in the PHY_REG_ADDR field of the PHY Access Register and sets the PHY_NBLK_RD_CMD bit. The host CPU can then poll the register until the PHY_CMD_DONE bit is 1, or it can wait for the MII Management Command Complete Interrupt (MCCINT in the Int0 Register). When the PHY_CMD_DONE bit is 1, the PHY_DATA field contains the data read from the specified external PHY register. If an error occurs in the read operation, the MII Management Read Error Interrupt (MREINT) bit in the Interrupt0 Register is set, and if the corresponding enable bit is set (MREINTE in the Interrupt Enable Register), the host CPU is interrupted.

To generate a blocking read, the host CPU uses the same procedure as it does for a non-blocking read, except that it sets the PHY_BLK_RD_CMD bit rather than the PHY_NBLK_RD_CMD bit, and it can poll the PHY Access Register until the PHY_CMD_DONE bit is set.

When any one of the PHY register access command bits (PHY_WR_CMD, PHY_BLK_RD_CMD or PHY_NBLK_RD_CMD) is set, the PHY_CMD_DONE bit must be cleared by writing a 1 to it in the same PHY_ACCESS register write.

The host CPU must not set both the PHY_BLK_RD_CMD bit and the PHY_NBLK_RD_CMD bit at the same time.

The host CPU must not attempt a second PHY register access until the first access is complete. When the access is complete, the PHY_CMD_DONE bit in the PHY Access Register and the MII Management Command Complete Interrupt (MCCINT) bit in the Interrupt Register will be set to 1, and if the corresponding enable bit is set, the host CPU will be interrupted. The host can either wait for this interrupt, or it can use some other method to guarantee that it waits for a long enough time. Note that with a 2.5 MHz MDC clock it takes about 27 µs to transmit a management frame with a preamble. However, if the Auto-Poll or Port Manager machines are active, there may be a delay in sending a host generated management frame while other frames are sent. Under these conditions, the host should always check for command completion.

For an MII Management Frame transmitted as the result of a host CPU access to the PHY Access Register, preamble suppression is controlled by the Preamble Suppression bit (PRE_SUP) in the PHY Access Register. If this bit is set to 1 the preamble will be suppressed. Otherwise, the frame will include a preamble. The host CPU should only set the Preamble Suppression bit when accessing a register in a PHY device that is known to be able to accept management frames without preambles. For PHY devices that comply with Clause 22 of IEEE Std 802.3, bit 6 of PHY Register 1 is fixed at 1 if the PHY will accept management frames with the preamble suppressed.

The PHY_PRE_SUP bit in the PHY_ACCESS register controls preamble suppression for PHY register accesses initiated either from the PHY_ACCESS register or from the BCR33 and BCR34 registers. When using BCR33 and BCR34 to access PHY registers, PHY_PRE_SUP must be 0 unless all connected PHY's support preamble suppression.

See *Appendix B, MII Management Registers,* for descriptions of the standard registers that are found in IEEE 802.3 compatible devices.

Auto-Poll State Machine

As defined in the IEEE 802.3 standard, the external PHY attached to the Am79C976 controller's MII has no way of communicating important timely status information back to the Am79C976 controller. Unless it polls the external PHY's status register, the Am79C976 controller has no way of knowing that an external PHY has undergone a change in status. Although it is possible for the host CPU to poll registers in external PHY devices, the Am79C976 controller simplifies this process by implementing an automatic polling function that periodically polls up to 6 user-selected PHY registers and interrupts the host CPU if the contents of any of these registers change.

The automatic polling of PHY registers is controlled by six 16-bit Auto-Poll registers, AUTOPOLL0 to AUTOPOLL5. By writing to the Auto-Poll registers, the user can independently define the PHY addresses and register numbers for six external PHY registers. The registers are not restricted to a single PHY device. In the Auto-Poll registers there is an enable bit for each of the selected PHY registers. When the host CPU sets one of these enable bits, the Auto-Poll logic reads the corresponding PHY register and stores the result in the corresponding Auto-Poll Data Register. (There is one Auto-Poll Data register for each of the six PHY registers.) Thereafter, at each polling interval, the Auto-Poll logic compares the current contents of the selected PHY register with the corresponding Auto-Poll Data Register. If it detects a change, it sets the MII Management Auto-Poll Interrupt (MAPINT) in the Interrupt Register, which causes an interrupt to the host CPU (if that interrupt is enabled).

Note that when the host CPU writes to one of the Auto-Poll Registers the contents of the associated Auto-Poll Data Register are considered to be invalid during the next polling cycle so that the next polling cycle updates

Downloaded from Arrow.com.

the appropriate Auto-Poll Data Register without causing an interrupt.

When the contents of one of the selected PHY registers changes, the corresponding Auto-Poll Data Register is updated so that another interrupt will occur when the data changes again.

Auto-Poll Register 0 differs from the other Auto-Poll Registers in several ways. The PHY address (AP_PHY0_ADDR) field of this register defines the default PHY address that is used by both the Auto-Poll State Machine and the Network Port Manager. The register number field is fixed at 1 (which corresponds to the external PHY status register), and the register is always enabled. This means that if the Auto-Poll State Machine is enabled, it will always poll register 1 of the default PHY and will interrupt the host CPU when it detects a change in that register.

In addition to the PHY address, register number, and enable bit, the Auto-Poll Registers contain two other control bits for each of the 5 user-selected registers. These bits are the Preamble Suppression (AP_PRE_SUP) and Default PHY (AP_DFLT_PHY) bits.

If the Preamble Suppression bit is set, the Auto-Poll sends management frames to the corresponding register with no preamble field. The host CPU should only set the Preamble Suppression bit for registers in PHY devices that are known to be able to accept management frames without preambles. For PHY devices that comply with Clause 22 of IEEE Std 802.3, bit 6 of PHY register 1 is fixed at 1 if the PHY will accept management frames with the preamble suppressed.

If the Default PHY bit (AP_DFLT_PHY) is set, the corresponding Preamble Suppression bit and PHY address field are ignored. In this case the Auto-Poll State Machine uses the default PHY address from the AP_PHY0_ADDR field, and suppresses the preamble if the Network Port Manager logic has determined that the default PHY device accepts management frames with no preamble. If the Network Port Manager logic has not determined that the default PHY device accepts management frames with no preamble, the Auto-Poll State Machine does not suppress the preamble when accessing the selected register.

The Auto-Poll State Machine is enabled when the Auto-Poll External PHY (APEP) bit (CMD3, bit 24) is set to 1. If APEP is cleared to 0, the Auto-Poll machine does not poll any PHY registers regardless of the state of the enable bits in the Auto-Poll registers. The APEP bit has no effect on the Network Port Manager, which may poll the default PHY even when the state of the APEP bit is 0.

The Auto-Poll's frequency of generating MII management frames can be adjusted by setting of the APDW bits (BCR32, bits 10-8). The delay can be adjusted from 0 MDC periods to 2048 MDC periods.

Network Port Manager

The Am79C976 controller is unique in that it does not require software intervention to control and configure an external PHY attached to the MII. This feature was included to ensure backwards compatibility with existing software drivers. The Am79C976 controller will operate with existing PCnet drivers from revision 2.5 upward (although older drivers will report incorrect statistics for the Am79C976 device). The heart of this automatic configuration system is the Network Port Manager.

The Network Port Manager initiates auto-negotiation in the external PHY when necessary and monitors the results. When auto-negotiation is complete, the Network Port Manager sets up the MAC to be consistent with the negotiated configuration. The Network Port Manager auto-negotiation sequence requires that the external PHY respond to the auto-negotiation request within 4 seconds. Otherwise, system software will be required to properly control and configure the external PHY attached to the MII. After auto negotiation is complete, the Network Port Manager generates MII management frames about once every 900 ms to monitor the status of the external PHY.

The Network Port Manager is enabled when the Disable Port Manager (DISPM) bit (CMD3, bit 14) is cleared to 0.

Auto-Negotiation

The external PHY and its link partner may have one or more of the following capabilities: 100BASE-T4, 100BASE-TX Full-/Half-Duplex, 10BASE-T Full-/Half-Duplex, and MAC Control PAUSE frame processing. During the auto-negotiation process the two PHY devices exchange information about their capabilities and then select the best mode of operation that is common to both devices. The modes of operation are prioritized according to the order shown in Table 11 (with the highest priority shown at the top of the table).

Table 11. Auto-Negotiation Capabilities

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

Downloaded from Arrow.com.

Auto-Negotiation goes further by providing a message-based communication scheme called, *Next Pages*, before connecting to the Link Partner. *The Network Port Manager does not support this feature. However, the host CPU can disable the Network Port Manager and manage Next Pages by accessing the PHY device through the PHY Access Register.* The host CPU can disable the Network Port Manager by setting the Disable Port Manager (DISPM) bit (CMD3, bit 14) to 1. (The DISPM bit corresponds to the Disable Auto-Negotiation Auto Setup (DANAS) bit in BCR32 of older PCnet family devices.)

To control the auto-negotiation process, the Network Port Manager generates MII Management Frames to execute the procedure described below. (See *Appendix B, MII Management Registers,* for the MII register bit descriptions.)

The Network Port Manager is held in the IDLE state while H_RESET is asserted, while the EEPROM is being read and while the DISPM bit is set. When none of these conditions are true, the Network Port Manager proceeds through the following steps:

1. If XPHYRST is set, write to the PHY's Control Register (R0) to set the Soft Reset bit and cause the PHY to reset. The Network Port Manager then periodically reads the PHY's Control Register (R0) until the reset is complete.
2. If XPHYRST is not set or after the PHY reset is complete, the PHY's Status Register (R1) is read.
3. If the PHY's Auto-Negotiation Ability bit (R1, bit 3) is 0 or if the XPHYANE bit in the Control2 Register is 0, write to the PHY's Control Register (R0) to disable auto-negotiation and set the speed and duplex mode to the values specified by the XPHYSP and XPHYFD bits in the Control2 Register "and"ed with the appropriate bits from the PHY's Technology Ability Field. Then proceed to step 8.
4. Otherwise write to the Auto-Negotiation Advertisement Register (R4). Bits A0 to A5 of Technology Ability field of R4 are taken from bits 15 to 11 in R1. Bit A6 of the Technology Ability field indicates the MAC's ability to respond to MAC Control Pause frames. This bit is set equal to the value of the Negotiate Pause Ability (NPA) bit in the Flow Control Register. The Next Page, Acknowledge, and Remote Fault bits are set to 0, and the Selector Field is set to 00001 to indicate IEEE Std 802.3.
5. Write to the Control Register (R0) to restart Auto-negotiation.
6. Poll R1 until the Auto-Negotiation Complete bit is set to 1.
7. Read the Auto-Negotiation Link Partner Ability Register (R5). Set the MAC's speed, duplex mode, and pause ability to the highest priority mode that is common to both PHY devices.
8. Poll R1 until the Link Status bit is 1. If Link Status is not found to be 1 after two polls at 900 ms intervals, go back to step 1.
9. Poll R1 at intervals of about 900 ms until the Link Status bit is 0. Go to step 8.

When Auto-Negotiation is complete, the Network Port Manager examines the MF Preamble Suppression bit in PHY register 1. If this bit is set, the Network Port Manager suppresses preambles on all frames that it sends until one of the following events occurs:

- A Software or hardware reset occurs.
- The DISPM bit (in CMD3.bit 14 Register) is set.
- Management frame read error occurs.
- The external PHY is disconnected.

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

The Network Port Manager is not disabled when the MDIO pin is held low when the MII Management Interface is idle. If no PHY is connected, reads of the external PHY's registers will result in read errors, causing the MREINT interrupt to be asserted.

Auto-Negotiation With Multiple PHY Devices

The MII Management Interface (MDC and MDIO) can be used to manage more than one external PHY device. The external PHY devices may or may not be connected to the Am79C976 controller's MII bus. For example, two PHY devices can be connected to the Am79C976 controller's MII bus so that the MAC can communicate over either a twisted-pair cable or a fiber-optic link. Conversely, several Am79C976 controllers may share a single integrated circuit that contains several PHY devices with separate MII busses but with only one MII Management bus. In this case, the MII Management Interface of one Am79C976 controller could be used to manage PHY devices connected to different Am79C976 controllers.

If more than one PHY device is connected to the MII bus, only one PHY device is allowed to be enabled at any one time. Since the Network Port Manager can not detect the presence of more than one PHY on the MII bus, the host CPU is responsible for making sure that only one PHY is enabled. The host CPU can use the PHY Access Register to set the Isolate bit in the Control Register (Register 0, bit 10) of any PHY that needs to be disabled.

Operation Without MMI Management Interface

The Port Manager normally sets up the speed, duplex mode, and flow control (pause) ability of the MAC based on the results of auto-negotiation. However, it is possible to operate the Am79C976 device with no MII

Downloaded from Arrow.com.

Management Interface connection, in which case the Port Manager is not able to start the auto-negotiation process or set up the MAC-based on auto-negotiation results. This may happen if the Am79C976 controller is connected to a multi-PHY device that has only one MII Management Interface that is shared among several PHYs.

If the Am79C976 controller is operating without a MII Management Interface connection to its external PHY, the host CPU can force the MAC into the desired state by setting the DISPM bit in CMD3 Register to 1 to disable the Port Manager, then writing to the following bits:

1. FORCE_FD (CMD3, bit 12),
2. FORCE_SPEED (CTRL2, bits 18-16),
3. FORCE_LINK_STAT (CMD3, bit 11), and
4. Force Pause Ability (FPA, FLOW_CONTROL, bit 20).

These bits set up the duplex mode, speed, and flow control ability in the MAC and put the MAC into the Link Pass state.

Regulating Network Traffic

The Am79C976 device provides two hardware mechanisms for regulating network traffic: 802.3x Flow Control and collision-based back pressure. 802.3x Flow Control applies to full-duplex operation only, while back pressure applies to half-duplex operation only. 802.3x Flow Control works by sending and receiving MAC Control PAUSE frames, which cause the receiving station to postpone transmissions for a time determined by the contents of the PAUSE frame. Back pressure forces collisions to occur when other nodes attempt to transmit, thereby preventing other nodes from transmitting for periods of times determined by the back-off algorithm.

MAC Control Pause Frames

The format of a MAC Control Pause frame is shown in Table 12.

Table 12. MAC Control Pause Frame Format

Octet Numbers	Field Name	Value
1-6	Destination Address	01-80-C2-00-00-01
7-12	Source Address	Sender's physical address
13-14	Length/Type	88-08
15-16	MAC Control Opcode	00-01
17-18	Request_operand	Pause time measured in slot times
19-60	Pad	Zeros
61-64	FCS	FCS

When a network station that supports IEEE 802.3x Flow Control receives a pause frame, it must suspend transmissions after the end of any frame that was being transmitted when the pause frame arrived. The length of time for which the station must suspend transmissions is given in the request_operand field of the pause frame. This pause time is given in units of slot times. For 10-Mbps and 100-Mbps 802.3 networks, one slot time is 512 bit times. The request_operand field is interpreted as Big-Endian data. Octet 17 is the most significant byte and octet 18 is the least significant byte.

Back Pressure

The Am79C976 device supports collision-based back pressure for congestion control when the device is operating in half-duplex mode. Back pressure is enabled when the device is operating in half duplex mode and either the Flow Control Command bit (FCCMD, FLOW_CONTROL, bit 16) is set or the FC Pin Enable bit (FCPEN, FLOW_CONTROL, bit 17) is set and the FC pin is asserted.

When the MAC begins receiving a frame that passes the address matching criteria and if back pressure is enabled, the MAC will intentionally cause a collision by transmitting a "phantom" frame consisting of a continuous stream of alternating 1s and 0s. The length of the phantom frame is 568 bits so that it will be interpreted as a runt frame.

Back pressure does not affect the transmission of a frame. The MAC will only force a collision when it begins to receive a new frame.

The generation of a Back-Pressure collision causes the XmtBackPressure MIB Counter to increment.

Enabling Traffic Regulation

Traffic regulation can be controlled either by external hardware or by CPU commands. Traffic regulation is affected by the following:

- Duplex mode
- Flow Control (FC) pin
- FC Pin Enable (FCPEN) bit
- Flow Control Command (FCCMD) bit
- Fixed Length Pause (FIXP) bit
- 16-bit PAUSE Length Register

Downloaded from Arrow.com.

- Negotiate Pause Ability (NPA) bit
- Force Pause Ability (FPA) bit.

The duplex mode affects the type of traffic regulation that is used. In full-duplex mode the FC pin and the FCPEN, FCCMD, and FIXP bits control the transmission of pause frames. In half-duplex mode the same pin and bits control the assertion of back pressure. Also, in half-duplex mode the Am79C976 device does not respond to received pause frames.

The Am79C976 device includes support for two styles of full-duplex flow control. In one style, which is similar to an XON-XOFF protocol, a pause frame whose request_operand field (bytes 17 and 18) contains 0FFFFh is sent to prevent the link partner from transmitting. Later, a pause frame whose request_operand field contains 0 is sent to allow the link partner to resume transmissions. This style of flow control is selected by clearing the Fixed Length Pause bit (FIXP) to 0.

For the other style of flow control, a single pause frame is sent to halt transmissions for a predetermined period of time. The contents of the request_operand field of this frame are taken from the Pause Length register. This style of flow control is selected by setting the Fixed Length Pause bit (FIXP) to 1.

Hardware Control of Traffic Regulation

The Flow Control pin (FC) allows external hardware to cause pause frames to be transmitted or back pressure to be asserted. The use of the FC pin for traffic regulation is enabled by the FC Pin Enable bit (ENFC). When FCPEN is cleared to 0, the signal on the FC pin is ignored. Otherwise, back pressure is enabled when FC is high and the device is operating in half-duplex mode, and pause frames are sent at FC pin signal transitions when the device is operating in full-duplex mode.

In full-duplex mode with the FC Pin Enable bit (FCPEN) =1, the actions that occur at low-to-high and high-to-low transitions of the FC pin depend on the value of the Fixed Length Pause bit (FIXP). If FIXP is 1, a low-to-high transition causes a pause frame to be sent with its request_operand field contents taken from the Pause Length register. In this case high-to-low transitions of the FC pin are ignored.

If FIXP is 0, a low-to-high transition sends a pause frame whose request_operand field contains 0FFFFh, while a high-to-low transition sends a pause frame whose request_operand field contains 0.

The effects of the FC pin are summarized in Table 13.

Table 13. FC Pin Functions

FC Pin Transition	FCPEN	FIXP	Duplex Mode	Action
X	0	X	X	No Action
0 to 1	1	X	Half	Enable back pressure
1 to 0	1	X	Half	Disable back pressure
0 to 1	1	1	Full	Send pause frame with request operand equal to the contents of the Pause Length register
1 to 0	1	1	Full	No action
0 to 1	1	0	Full	Send pause frame with request operand equal to 0FFFFh.
1 to 0	1	0	Full	Send pause frame with request operand equal to 0000h.

Software Control of Traffic Regulation

For software control of traffic regulation the Flow Control Command bit (FCCMD) mimics the FC pin.

In half-duplex mode, back pressure is enabled when FCCMD is set to 1, and it is disabled when FCCMD is cleared to 0.

In full-duplex mode, the act of setting FCCMD to 1 causes a pause frame to be sent. The contents of the request_operand field of the frame depend on the state of the FIXP bit. If FIXP is 1, the contents of the request_operand field are copied from the Pause Length register. If FIXP is 0, the contents of the request_operand field are set to 0FFFFh.

In full-duplex mode, if FIXP is 0, the act of clearing FCCMD to 0 causes a pause frame to be sent with its request_operand field cleared to 0.

If FIXP is set to 1, the FCCMD bit is self-clearing--the CPU does not have to write to the Am79C976 device to clear the FCCMD bit. This allows the CPU to use a single write access to cause a pause frame to be sent with a predetermined request_operand field.

Downloaded from Arrow.com.

The effects of the FCCMD bit are summarized in Table 14.

Table 14. FCCMD Bit Functions

FCCMD Transition	FIXP	Duplex Mode	Action
0 to 1	X	Half	Enable back pressure
1 to 0	X	Half	Disable back pressure
0 to 1	1	Full	Send pause frame with request operand equal to the contents of the Pause Length register. Automatically clear FCCMD to 0.
1 to 0	1	Full	No action. (FCCMD is cleared automatically when FIXP = 1.)
0 to 1	0	Full	Send pause frame with request operand equal to 0FFFFh.
1 to 0	0	Full	Send pause frame with request operand equal to 0000h.

Programming the Pause Length Register

When the host CPU changes the contents of the Pause Length Register, it must make sure that no Pause frame is transmitted while the register is being updated. If the host CPU can not control the state of the FC pin, it can clear the FCPEN pin so that the FC pin will be ignored. It can then poll the PAUSE_PENDING bit in the Status0 Register until that bit is 0. When FCPEN and PAUSE_PENDING are both 0, it is safe to write to the Pause Length Register.

PAUSE Frame Reception

The ability to respond to received pause frames, or pause ability, is controlled independently from the transmission of pause frames. When pause ability is enabled, the receipt of a pause frame causes the device to stop transmitting for a time period that is determined by the contents of the pause frame.

Pause ability is enabled by Negotiate Pause Ability (NPA, FLOW_CONTROL, bit 19) and Force Pause Ability (FPA, FLOW_CONTROL, bit 20). If the FPA bit is set, pause ability is enabled regardless of the Pause Ability state of the link partner. If the NPA bit is set and the FPA bit is not set, pause ability is enabled only if the auto-negotiation process determines that the link partner also supports 802.3x flow control.

The auto-polling state machine is extended to read the external PHY Status registers at register locations 1, 4, and 5. (The contents of these registers are defined in the IEEE P802.3u specification.) From Register 1 the state machine obtains the link status and auto-negotiation status as well as Jabber and Remote Fault indications. If auto negotiation is complete, the logic uses the Technology Ability fields of the Auto-Negotiation Advertisement register (Register 4) and the Auto-Negotiation Link Partner Ability register (Register 5) to determine the network speed and duplex mode and the flow control status. The MAC device will be put into the speed and duplex mode for the highest common ability that the PHY and its link partner share. If full-duplex mode is selected and the PAUSE bits are set in both Register 4 and Register 5, pause ability will be enabled so that the MAC will be able to respond to MAC Control PAUSE frames as described in the IEEE P802.3x specification.

A MAC Control PAUSE frame is any valid frame with the following:

- A destination address field value equal to the MAC's physical address or equal to the reserved multicast address 01-80-C2-00-00-01,
- A Length/Type field value equal to 88-08, and
- A MAC Control Opcode field value equal to 0001.

If such a frame is received while pause ability is enabled, the MAC device will wait until the end of the frame currently being transmitted (if any) and then stop transmitting for a time equal to the value of the request_operand field (octets 17 and 18) multiplied by 512-bit times.

If another MAC Control PAUSE frame is received before the Pause timer has timed out, the Pause timer will be reloaded from the request_operand field of the new frame so that the new frame overrides the earlier one.

Received MAC Control PAUSE frames are handled completely by the Am79C976 hardware. They are not passed on to the host computer. However, MAC Control frames with opcodes not equal to 0001h are treated as normal frames, except that their reception causes the Unsupported Opcodes counter to be incremented.

Since the host computer does not receive MAC Control PAUSE frames, 32-bit MIB counters have been added to record the following:

- MAC Control Frames received
- Unsupported Opcodes received
- PAUSE frames received

Downloaded from Arrow.com.

External Address Detection Interface

The EADI is provided to allow external address filtering and to provide a Receive Frame Tag word for proprietary routing information. This feature is typically utilized by terminal servers, switches and/or router products. The EADI interface can be used in conjunction with external logic to capture the packet destination address from the MII input data stream as it arrives at the Am79C976 controller, to compare the captured address with a table of stored addresses or identifiers, and then to determine whether or not the Am79C976 controller should accept the packet.

The EADI consists of the External Address Reject ($\overline{EAR}$), Start Frame-Byte Delimiter (SFBD), Receive Frame Tag Data (RXFRTGD), and Receive Frame Tag Enable (RXFRTGE) pins.

The SFBD pin indicates two types of information to the external logic--the start of the frame and byte boundaries. The first low-to-high transition on the SFBD pin after the assertion of the RX_DV signal indicates that the first nibble of the Destination Address field of the incoming frame is available on the RXD[3:0] pins. Thereafter, SFBD toggles with each RX_CLK pulse so that SFBD is high when the least significant nibble of frame date is present on the RXD[3:0] lines and low when the most significant nibble is present. SFBD stays low when RX_DV is not asserted (which indicates that the receiver is idle).

Note that the SFBD signal is available on any LED pin. To direct the SFBD signal to one of the LED pins, the SFBDE and LEDPOL bits should be set to 1 and the PSE bit should be cleared to 0 in the appropriate LED register. The SFBDE bit directs the SFBD signal to the pin, the LEDPOL bit sets the polarity to active high and enables the totem-pole driver, and the PSE bit disables the LED pulse stretcher logic.

If the system needs all four LEDs as well as the EADI function, the Am79C976 controller can be programmed to use the shared pin for the LED function, and the external logic can be designed to generate the SFBD signal by searching for the 11010101b Start Frame Delimiter (SFD) pattern in the RCD[3:0] data.

The external address detection logic can use the $\overline{EAR}$ input to indicate whether or not the incoming frame should be accepted. If the $\overline{EAR}$ signal remains high during the receive protect time, the frame will be accepted and copied into host system memory. The receive protect time is a period of time measured from the receipt of the SFD field of a frame. The length of the receive protect time is programmable through the Receive Protect Register.

A frame is accepted if it passes either the internal address match criteria or the external address match criteria. If the internal address logic is disabled, the acceptance of a frame depends entirely on the external address match logic. If the external address match logic is disabled, the acceptance of a frame depends entirely on the internal address match logic.

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

External address matching can be disabled by holding the $\overline{EAR}$ pin low. There is no programmable bit that causes the Am79C976 device to ignore the state of the $\overline{EAR}$ pin.

The EADI logic only samples $\overline{EAR}$ from 2 nibble times after SFD until the end of the receive protect time. (See the *Receive Protect Register* section.) The frame will be accepted if $\overline{EAR}$ has not been asserted during this window. $\overline{EAR}$ must have a pulse width of at least two bit times plus 10 ns.

External Address Detection Interface: Receive Frame Tagging

Receive Frame Tagging is a feature that allows the external address detection logic to pass an identification code or tag to the Am79C976 controller to be placed in the RX descriptor corresponding to a received frame. The external logic can shift this tag in as a serial bit stream on the Receive Frame Tag Data (RXFRTGD) pin. It uses the Receive Frame Tag Enable (RXFRTGE) pin to indicate when the tag data is valid. The clock signal for shifting in the tag data is RX_CLK. See Figure 35.

If the Software Style (SWSTYLE) field in BCR20 contains the value 2 or 3, the Receive Frame Tag can be up to 15 bits long. In this case the tag data is sampled on the low-to-high transition of RX_CLK whenever RXFRTGE is high. If SWSTLYE = 5, the Receive Frame Tag can be up to 32 bits long. In this case, the tag data is sampled on both edges of RX_CLK so that the entire tag can be shifted in 16 RX_CLK cycles or less, depending on the length of the tag. If SWSTYLE is 0 or 4, Receive Frame Tagging is not supported. In those cases the descriptor space is allocated to other functions.

If SWSTYLE = 5, tag bits are shifted in the order B31, B15, B30, B14, … , B0, where B0 is the least significant bit of the tag. This sequence allows the external logic to be simplified slightly if the system design requires a frame tag of fewer than 17 bits. In this case the external logic can use only one clock edge to shift in the data. Since the Am79C976 device samples the RXFRTGD pin on both edges of RX_CLK, the same data will appear in the upper and lower halves of the frame tag field in the descriptor.

Downloaded from Arrow.com.

If SWSTYLE = 2 or 3, tag bits are shifted in the order B14, B13, ... , B0.

Because of the order in which frame tag bits are shifted in, if the tag is shorter than 15 bits, the tag data will be placed in the least significant portion of the Receive Frame Tag field of the RX descriptor, and the most significant bits of the field will be cleared to zeros.

RXFRTGE need not be a continuous signal. It can toggle on and off so that the tag data can be shifted in at a slower rate than the frequency of RX_CLK. The length of the frame tag is determined by the number of RX_CLK cycles during which RXFRTGE is asserted before the end of the frame arrives (with a maximum of 15 bits for SWSTYLE 2 or 3 or a maximum of 32 bits for SWSTYLE 5). The last bit of the Receive Frame Tag must be shifted into the RXFRTGD input at least one RX_CLK cycle before RX_DV is de-asserted.

The Receive Frame Tagging feature is enabled by the RXFRTGEN bit in CMD3 Register. When this bit is cleared to 0, the Receive Frame Tag field of the RX descriptor will be filled with zeros.

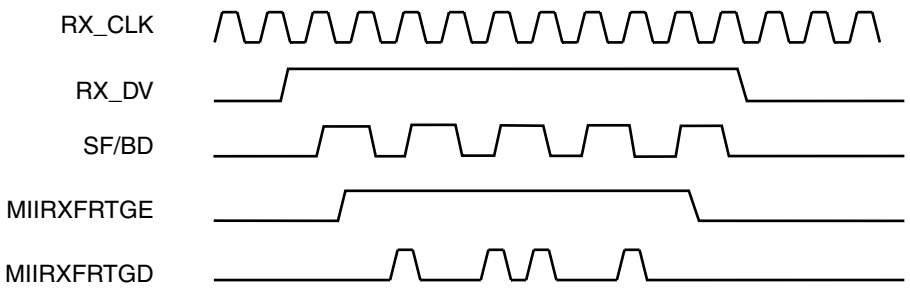


Figure 35. MII Receive Frame Tagging

External Memory Interface

The Am79C976 controller contains an External Memory Interface that supports Flash (or EPROM) devices as boot devices, as well as SSRAM for frame data storage. The controller provides read and write access to Flash or EPROM. No glue logic is required for the memory interface.

The Am79C976 device contains a built-in self test system (MBIST) that can be programmed to run a diagnostics test on the external SSRAM.

The external SSRAM is organized around a 32-bit data bus. The memory can be as large as 1M X 32 bits. The memory devices can be either JEDEC standard Pipeline Burst Synchronous Static RAM devices (PB-SSRAM) or ZBT™ Synchronous Static RAM (ZBT-SSRAM) with pipelined outputs. The SRAM_TYPE field of the CTRL0 Register must be initialized to indicate which type of SSRAM is actually used.

The contents of the SRAM_TYPE field are defined in Table 15.

Table 15. SRAM_TYPE Field Encoding

SRAM_TYPE[1:0]	External Memory Type
00	Reserved
01	ZBT
10	Reserved
11	Pipelined Burst

The width of the Flash memory (or EPROM) is 8 bits. The memory can be as large as 16M X 8 bits.

The external memory bus uses the same address, data, and control pins to access both Flash and SSRAM memory, but it has separate chip select (or chip enable) pins so that only one device can be selected at a time. $\overline{\text{FLCS}}$ selects the Flash memory, while $\overline{\text{ERCE}}$ selects the SSRAM. The Flash memory must not be accessed when the Am79C976 controller is running (when the RUN bit in CMD0 is set to 1). Any access to the Flash memory clears the RUN bit and thereby abruptly stops all network and DMA operations.

ERA[19:0] provides 20 bits of address for the SSRAM and the lower 20 bits of address for the Flash memory.

Downloaded from Arrow.com.

The higher 4 bits of address for the Flash memory are shared with bits [11:8] of the SSRAM data bus (ERD[11:8]). The lower 8 bits of the external memory data bus ERD[7:0] are used by both the SSRAM and the Flash. The high order 20 bits of the external memory data bus ERD[31:12] are used only by the SSRAM.

The output enable signal for the Flash ($\overline{\text{FLOE}}$) shares a pin with the SSRAM Address Advance signal (ERADV).

Figure 36 shows how the SSRAM and Flash can be connected to the Am79C976 controller.

Expansion ROM - Boot Device Access

The Am79C976 controller supports EPROM or Flash as an Expansion ROM boot device. Both are configured using the same methods and operate the same. See Figure 36. See the previous section on Expansion ROM transfers for the PCI timing and functional description of the transfer method.

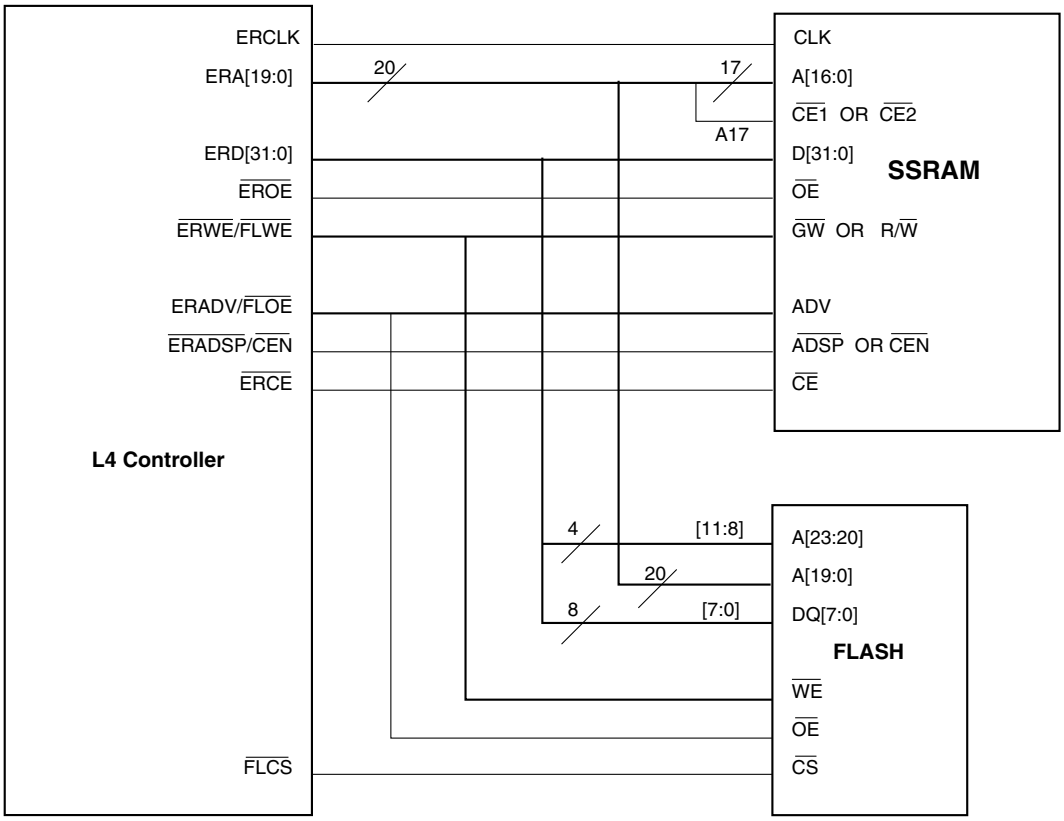


Figure 36. External SSRAM and Flash Configuration

The Am79C976 controller will always read four bytes for every host Expansion ROM read access. The interface to the Expansion Bus is timed by an internal signal called ROMCLK, which runs at one fourth of the frequency of the external memory interface clock (ERCLK). Thus, when the clock select pins are configured so that ERCLK runs at 90 MHz; ROMCLK runs at 22.5 MHz.

The time that the Am79C976 controller waits for data to be valid is programmable. ROMTMG (CTRL0, bits 8-11 or BCR18, bits 15-12) defines the time from when the Am79C976 controller drives ERA[19:0] with the Expansion ROM address to when the Am79C976 controller latches in the data on the ERD[7:0] inputs. The register value specifies the time in number of ROMCLK cycles. When ROMTMG is set to nine (the default value), ERD[7:0] is sampled with the next rising edge of ROMCLK ten cycles after ERA[19:0] was driven with a new address value. The clock edge that is used to sample the data is also the clock edge that generates the next Expansion ROM address. All four bytes of Expansion ROM data are stored in holding registers.

Because Expansion ROM accesses take longer than 16 PCI bus clock cycles, the PCI access will be disconnected with no data transfer after 15 clocks. Subsequent accesses will be retried until all four bytes have been read from the Expansion ROM.

Downloaded from Arrow.com.

The timing diagram in Figure 37 assumes the default programming of ROMTMG (1001b = 9 CLK). After reading the first byte, the Am79C976 controller reads in three more bytes by incrementing the lower portion of the ROM address. The PCI bus logic generates disconnect/retry cycles until all 32 bits are ready to be transferred over the PCI bus. When the host tries to perform a burst read of the Expansion ROM, the Am79C976 controller will disconnect the access at the second data phase.

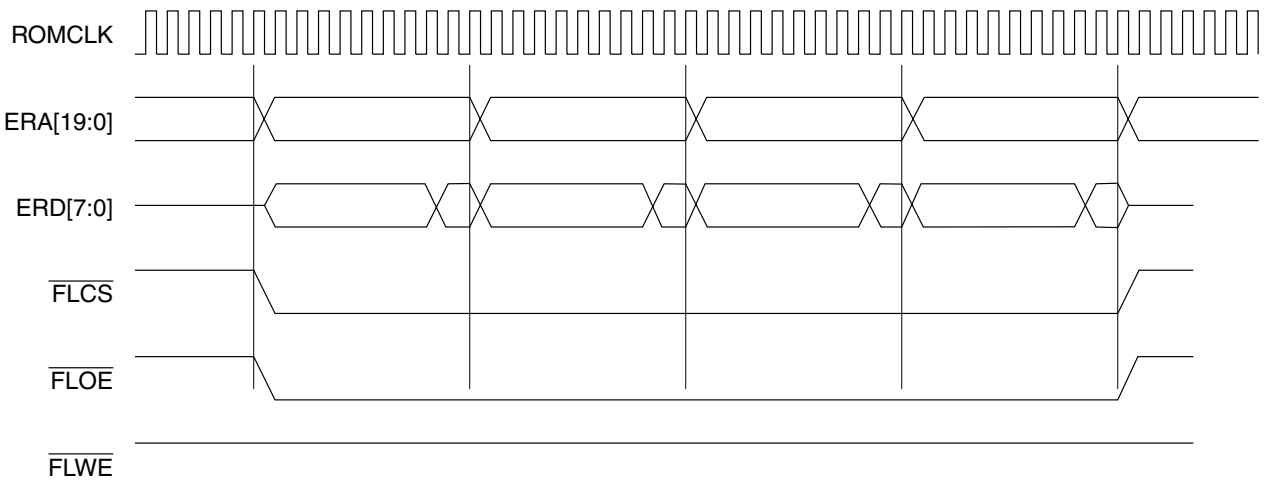


Figure 37. Expansion ROM Bus Read Sequence

The host must program the Expansion ROM Base Address register (ROMBASE) in the PCI configuration space before the first access to the Expansion ROM. The Am79C976 controller will not react to any access to the Expansion ROM until both MEMEN (PCI Command register, bit 1) and ROMEN (PCI Expansion ROM Base Address register, bit 0) are set to 1.

The amount of memory space that the Am79C976 device will claim for the Expansion ROM depends on the contents of the Expansion ROM Configuration Register (ROM_CFG), which should be loaded from the EEPROM. This register is included in the Am79C976 device so that the controller can accommodate ROMs of different sizes without wasting memory space. The ROM occupies a block of memory space that is some power of two between 2K and 16M in size. If the ROM requires 2^n bytes of address space, bits 1 through n-1 of the Expansion ROM Base Address Register in PCI configuration space (ROMBASE) should appear to be wired to 0. The contents of the Expansion ROM Configuration Register (ROM_CFG) determine how many bits of the configuration space register are forced to 0.

Bits [15:1] of ROM_CFG correspond to bits [23:9] of ROMBASE and bit 0 of ROM_CFG corresponds to bit 0 of ROMBASE. If a bit in ROM_CFG is set to 0, the corresponding bit in ROMBASE is fixed at zero. If a bit in ROM_CFG is set to 1, the corresponding bit in ROMBASE can be programmed to 0 or 1 through PCI configuration space accesses to ROMBASE.

Bit 0 of ROM_CFG controls bit 0 of ROMBASE. If bit 0 of ROM_CFG is 0, the host CPU cannot write to bit 0 of ROMBASE. This bit is the address decode enable bit. When this bit is fixed at 0, it will appear to the host CPU that the ROM Base Address Register and, therefore, the expansion ROM does not exist.

If bit 0 of ROM_CFG is set to 1, the host CPU is able to read and write bit 0 of ROMBASE.

As an example, if the Expansion ROM occupies 2^{16} (65536) bytes, bits 15:9 of ROMBASE should be fixed at 0. Since bits 15:9 of ROMBASE are controlled by bits 7:1 of ROM_CFG, bits 7:1 of ROM_CFG should be cleared to 0 and bits 15:8 should be set to 1. To make ROMBASE accessible to the host CPU, bit 0 of ROM_CFG should be set to 1. Therefore, ROM_CFG should be set to FF01h. If the host CPU writes all 1s to the ROMBASE register and then reads back the contents of ROMBASE, the result would be FFFF0001h.

After the host CPU has written to the Expansion ROM Base Address Register in PCI configuration space to map the ROM into PCI memory space and to enable accesses to the ROM, the address output to the Expansion ROM will be the offset from the address on the PCI bus to ROMBASE.

Downloaded from Arrow.com.

The Am79C976 controller aliases all accesses to the Expansion ROM of the command types *Memory Read Multiple* and *Memory Read Line* to the basic Memory Read command.

Since setting MEMEN also enables memory mapped access to the I/O resources, attention must be given to the PCI Memory Mapped I/O Base Address register, before enabling access to the Expansion ROM. The host must set the PCI Memory Mapped I/O Base Address register to a value that prevents the Am79C976 controller from claiming any memory cycles not intended for it.

During the boot procedure, the system will try to find an Expansion ROM. A PCI system assumes that an Expansion ROM is present when it reads the ROM signature 55h (byte 0) and AAh (byte 1).

Direct Flash Access

In addition to mapping the Flash memory into PCI address space, the Am79C976 controller provides an indirect read/write data path for programming the Flash memory. The Flash is accessed by first writing the memory address to the Flash Address Register, and then reading or writing the Flash Data Register.

For software compatibility with older PCnet devices, the Flash device can also be accessed by a read or write to the Expansion Bus Data port (BCR30). The user must load the upper address EPADDRU (BCR 29, bits 3-0). EPADDRU is not needed if the Flash size is 64K or less, but still must be programmed. The user will then load the lower 16 bits of address, EPADDRL (BCR 28, bits 15-0).

Flash/EPROM Read

A read to the Flash Data Register will start a read cycle on the External Memory Interface. The Am79C976 controller will drive ERD[11:8] with the 4 most significant address bits at the same time that it drives ERA[19:0] with the 20 least significant bits.

The $\overline{\text{FLCS}}$ pin is driven low for the value ROMTMG + 1. Figure 38 assumes that ROMTMG is set to nine. ERD[7:0] is sampled with the next rising edge of CLK ten clock cycles after ERA[19:0] was driven with a new address value. This PCI slave access to the Flash/EPROM will result in a retry for the very first access. Subsequent accesses may give a retry or not, depending on whether or not the data is present and valid. The access time is dependent on the ROMTMG bits (CTRL0, bits 11-8, or BCR18, bits 15-12) and can be tuned for the particular memory device used.

This access mechanism using BCR28, 29, and 30 differs from the Expansion ROM access mechanism since only one byte is read in this manner, instead of the 4 bytes in an Expansion ROM access.

If the Lower Address Auto Increment (LAAINC) bit (FLASH_ADDR, bit 31 or BCR29, bit 14) is set, the EBADDRL address will be incremented and a continuous series of reads from the Expansion Data Port (FLASH_DATA or EBDATA, BCR30) is possible. The upper address field, EBADDRU, is not automatically incremented when the lower address field, EBADDRL rolls over.

The Flash write procedure is almost identical to the read access, except that the Am79C976 controller will not drive $\overline{\text{FLOE}}$ low. The $\overline{\text{FLCS}}$ and $\overline{\text{FLWE}}$ signals are driven low for the value ROMTMG again. The write to the FLASH port is a posted write and will not result in a retry to the PCI, unless the host tries to write a new value before the previous write is complete. Then the host will experience a retry. See Figure 39.

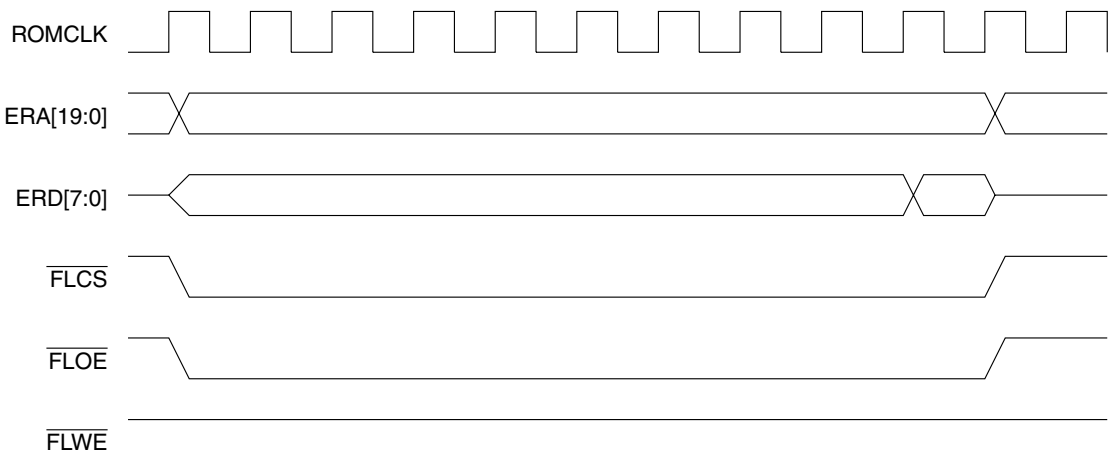


Figure 38. Flash Read from Expansion Bus Data Port

Downloaded from Arrow.com.

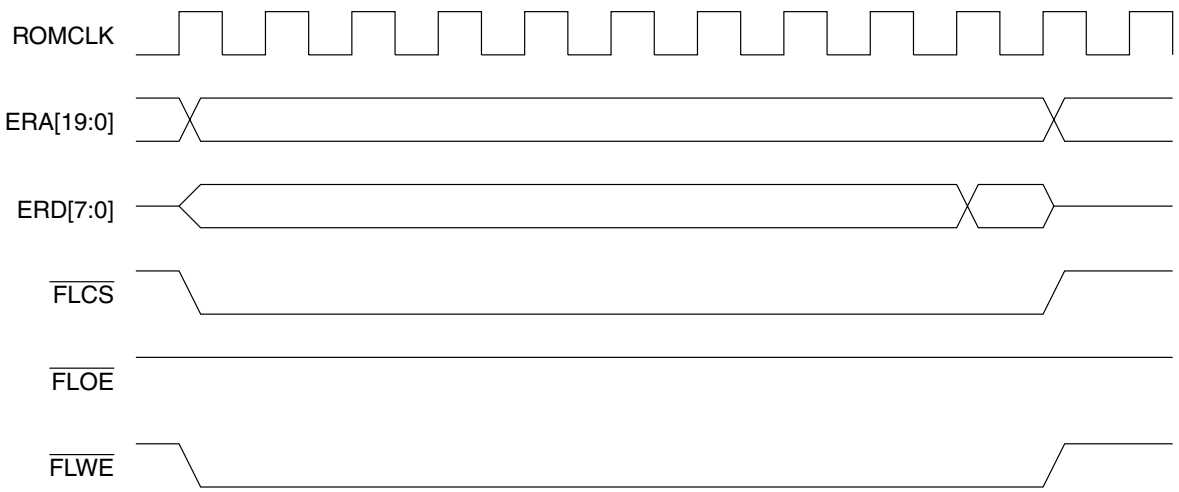


Figure 39. Flash Write Sequence

SRAM Configuration

The Am79C976 controller uses external SSRAM for receive and transmit FIFOs. The size of the SSRAM can be up to 4 Mbytes, organized as 1M X 32 bits. The size of the SSRAM is indicated by the contents of the SSRAM Size Register (or BCR25). SRAM_SIZE should be loaded from the EEPROM.

The SSRAM is programmed in units of 512-byte pages. To specify how much of the SSRAM is allocated to transmit and how much is allocated to receive, the user should program SRAM_BND Register (or BCR26, bits 15-0) with the page boundary where the receive buffer begins. The SRAM_BND is also programmed in units of 512-byte pages. The transmit buffer space starts at 0000h. It is up to the user or the software driver to split up the memory for transmit or receive; there is no defaulted value. The minimum SSRAM size required is four 512-byte pages for each transmit and receive queue, which limits the SSRAM size to be at least 4 Kbytes.

The SRAM_BND upon H_RESET will be reset to 0000h. SRAM_BND must be programmed to a non-zero value if the transmitter is enabled. SRAM_BND should be programmed to a value larger than the maximum frame size to use the automatic retransmission options, REX_UFLO, REX_RTRY, and RTRY_LCOL, or if the transmit FIFO start point, XMTSP, is set to Full Frame. (XMTSP is CTRL1, bits 17-16, or CSR80, bits 11-10.)

The Am79C976 controller does not allow software diagnostic access to the SRAM as do older devices in the PCnet family. The Am79C976 controller provides software access to an internal memory built-in self-test (MBIST) controller which runs extensive, at-speed tests on the external SRAM, internal SRAM access logic, and the PC board interconnect.

The MBIST controller can determine the size of the external SRAM and verify its operation using the following procedure:

1. Program SRAM_SIZE to the minimum allowed value of 4.
2. Write DM_START and DM_FAIL_STOP (write DATAMBIST bits 63:56 with 0x28). The remainder of the DATAMBIST register ignores writes so it may be written with arbitrary data or not written at all.
3. Read DM_DONE (DATAMBIST bit 63) and DM_ERROR (DATAMBIST bit 62) until DM_DONE is set.
4. If DM_ERROR is set, the memory is defective; report the error and exit.
5. Program SRAM_SIZE to the maximum value of 0x8000 and repeat steps 2 and 3.
6. If DM_ERROR is zero, report the current value of SRAM_SIZE as the SSRAM size.
7. If DM_ERROR is set, program SRAM_SIZE to one-half the maximum (0x4000) and repeat steps 2 and 3.
8. Repeat, using the binary search algorithm, until the SRAM size has been determined.

EEPROM Interface

The Am79C976 device includes an interface to an optional 16-bit word-oriented 93Cxx-compatible serial EEPROM that supports automatic address incrementing (sequential read). This EEPROM can be used for storing initial values for Am79C976 registers. The contents of this EEPROM are automatically loaded into the

Downloaded from Arrow.com.

selected registers after a reset operation or whenever the host CPU requests an EEPROM read operation.

Note that if the EEPROM is not included in the system, the MAC address (and Magic Packet information, if needed) must be initialized by the host CPU.

The Am79C976 device automatically detects the size of the EEPROM. When the EEPROM decodes a read command, it drives its DO pin low when the A0 address bit is written to the DI pin. The Am79C976 device uses this fact to detect the number of bits in the EEPROM address and from this determines the EEPROM size.

Data in the EEPROM are interpreted as three-byte entries that contain register address and register data so that the system designer can choose which registers will automatically be loaded. In a typical system, the EEPROM would be used to initialize the device's IEEE 802 physical address, the PCI Subsystem Vendor ID, LED configuration, SSRAM configuration, and other hardware configuration information. For compatibility with older PCnet family software the Address PROM Space should be loaded from the EEPROM. See the *Address PROM Space* section for details.

Only the memory-mapped registers can be loaded from the EEPROM. While the CSRs and BCRs are not memory-mapped, all useful bits in the CSRs and BCRs are aliased into memory-mapped registers so that all useful bits can be loaded from the EEPROM.

Most of the memory-mapped registers are 32 bits wide and occupy 4 bytes of memory space each. For example, the CMD2 Register is located at offset 50h from the memory base address. Its least significant 16 bits can be accessed at offset 50h, and its most significant 16 bits can be accessed at offset 52h. Register data are loaded from the EEPROM 16 bits at a time, so that the high order bits of a register are loaded independently from the low order bits.

The EEPROM Access Register gives the host CPU direct access to the interface pins so that it can read from or write to the EEPROM.

Automatic EEPROM Read Operation

After the trailing edge of the $\overline{\text{RESET}}$ signal or after the PREAD bit in BCR19 is set, the Am79C976 device begins to read data from the EEPROM. Data from the EEPROM are interpreted as a string of 3-byte entries. Each entry contains a 1-byte register address and a 2-byte register data field. The register address field contains the offset of the target register divided by 2. The initialization logic writes the contents of the register data field into the register selected by the register address byte.

Since EEPROM data are loaded two bytes at a time, the least significant bit of the target register offset is omitted from the address field. Only bits 8:1 are included. Therefore, the register address byte contains the offset of the target register divided by two. For example, the Control2 Register (CTRL2) is a 32-bit register located at offset 70h (relative to the contents of the Memory-Mapped I/O Base Address Register). Therefore, the byte stream 38h, 02h, 05h would cause the value 0205h to be loaded into bits [15:0] of CTRL2, and 39h, 00h, 03h would cause the value 0003h to be loaded into bits [31:16] of the same register.

If the value of the address byte is 0FFh, the following 2-byte field is interpreted as a 16-bit CRC code rather than as register data. The CRC code covers all EEPROM data up to and including the address byte of the entry containing the CRC. All EEPROM data after the CRC code word are ignored.

The CRC code used is CRC-16, which is based on the generator polynomial $x^{16} + x^{15} + x^2 + 1$.

The EEPROM must contain data for an odd number of registers so that the CRC is aligned on a 16-bit word boundary in the EEPROM. If an even number of registers need to be loaded from the EEPROM, two duplicate entries for the same register can be included so that the CRC is aligned properly.

For full compatibility with legacy Magic Packet software, the EEPROM should initialize both the APROM area (offset 0-0Fh) and the PADR Register.

Data are shifted into or out of the EEPROM most significant bit first.

Figure 41 shows the mapping of the 3-byte entries into the 16-bit word-oriented EEPROM.

If the Am79C976 device detects a CRC error in the EEPROM or fails to detect the presence of an EEPROM, it restores all registers to their default values and clears the PVALID bit in BCR19 to indicate the error.

Downloaded from Arrow.com.

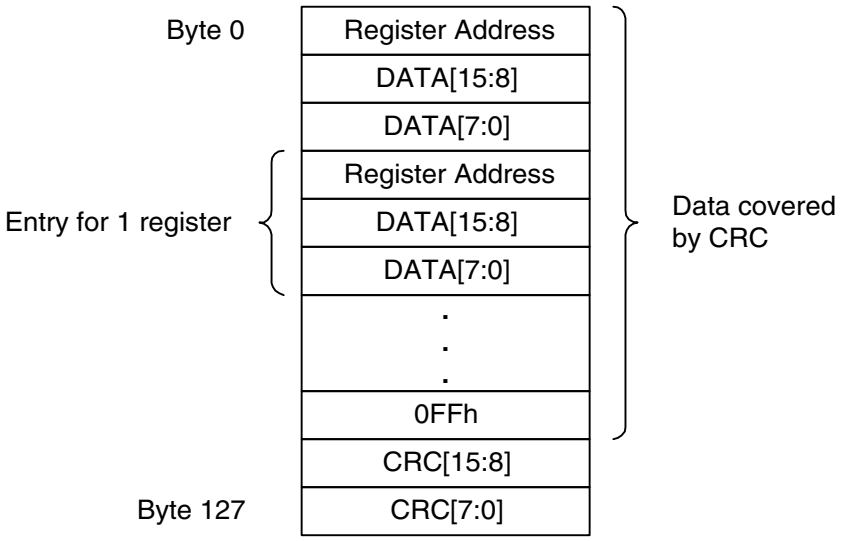


Figure 40. EEPROM Data Format

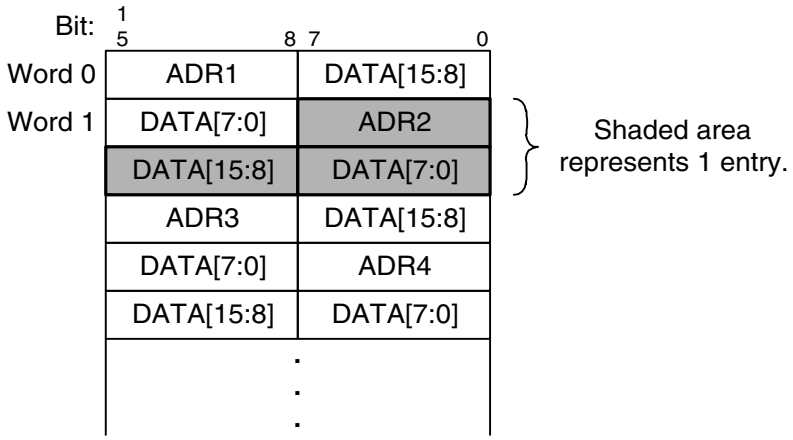


Figure 41. EEPROM Entry Positions

Note: *All registers are restored to their default values, not just those registers that were altered by the EEPROM read operation.*

If the Am79C976 device detects a correct CRC code, it sets the PVALID bit to 1 to indicate that the registers have been successfully initialized.

The CPU can initiate an automatic EEPROM read operation at any time by setting the PREAD bit in BCR19 to 1.

The CPU cannot access any Am79C976 register while an automatic EEPROM read operation is in progress. If the CPU attempts to access a register during this time, the Am79C976 controller will terminate the access attempt by asserting $\overline{\text{DEVSEL}}$ and $\overline{\text{STOP}}$ while $\overline{\text{TRDY}}$ is not asserted, a combination that indicates that the initi-

Downloaded from Arrow.com.

ator must disconnect and retry the access at a later time. The automatic read operation takes about 180 µs for each 16-bit register that is initialized plus 180 µs for the CRC code word.

EEPROM Auto-Detection

When the address field of an EEPROM instruction is shifted in through the DI pin of the EEPROM, the EEPROM drives its DO pin low when the A0 bit appears on the DI pin. The Am79C976 controller makes use of this feature to detect the presence of an EEPROM. When the device attempts to read the first word from the EEPROM and if the EEDO pin is not driven low before the 15th EESK clock cycle, the device assumes that there is no EEPROM present.

Direct Access to the Interface

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

EEPROM CRC Calculation

The EEPROM interface logic first shifts each 16-bit word from the EEPROM most significant bit first into an internal holding register. Then it shifts the word through the CRC logic least significant bit first, effectively swapping the bytes. Therefore, the data shown in Figure 42 are processed by the CRC logic in the following order: DATA[15:8], ADR1, ADR2, DATA[7:0], DATA[7:0], DATA[15:8], ...

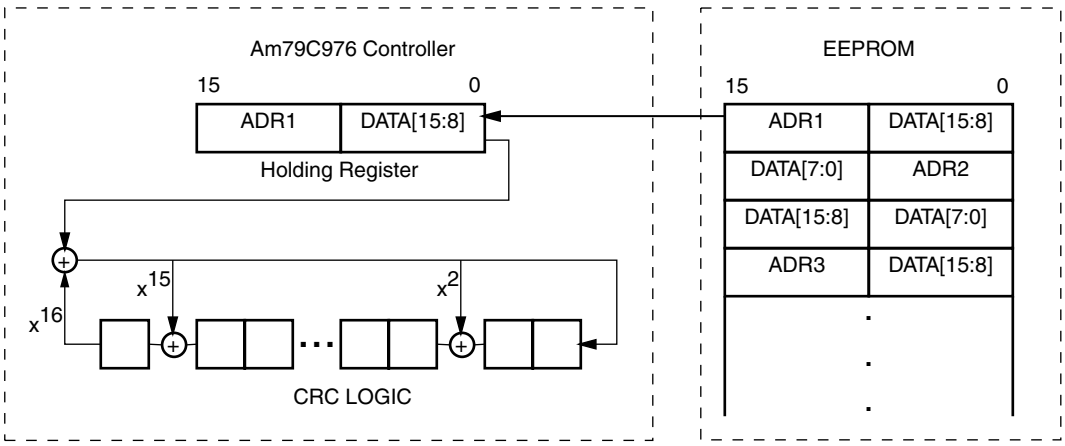


Figure 42. CRC Flow

LED Support

The Am79C976 controller can support up to four LEDs. LED outputs LED0, LED1, and LED2 allow for direct connection of an LED and its supporting pull-up device.

In applications that want to use the pin to drive an LED and also have an EEPROM, it might be necessary to buffer the LED3 circuit from the EEPROM connection. When an LED circuit is directly connected to the EEDO/LED3/RXFRTGD pin, then it is not possible for most EEPROM devices to sink enough I_{OL} to maintain a valid low level on the EEDO input to the Am79C976 controller. Use of buffering can be avoided if a low power LED is used.

Each LED can be programmed through a BCR register to indicate one or more of the following network statuses or activities: Collision Status, Full-Duplex Link Status, Receive Match, Receive Status, Magic Packet, Transmit Status, and Start Frame/Byte Delimiter.

The LED pins can be configured to operate in either open-drain mode (active low) or in totem-pole mode (active high). The output can be stretched to allow the human eye to recognize even short events that last only several microseconds. After H_RESET, the four LED outputs are configured as shown in Table 16.

For each LED register, each of the status signals is AND'd with its enable signal, and these signals are all OR'd together to form a combined status signal. Each LED pin combined status signal can be programmed to run to a pulse stretcher, which consists of a 3-bit shift register clocked at 38 Hz (26 ms). The data input of each shift register is normally at logic 0. The OR gate

Downloaded from Arrow.com.

output for each LED register asynchronously sets all three bits of its shift register when the output becomes asserted. The inverted output of each shift register is used to control an LED pin. Thus, the pulse stretcher provides 2 to 3 clocks of stretched LED output, or 52 ms to 78 ms. See Figure 43.

Table 16. LED Default Configuration

LED Output	Indication	Driver Mode	Pulse Stretch
LED0	Link Status	Open Drain - Active Low	Enabled
LED1	Activity	Open Drain - Active Low	Enabled
LED2	Speed	Open Drain - Active Low	Enabled
LED3	Coll	Open Drain - Active Low	Enabled

B

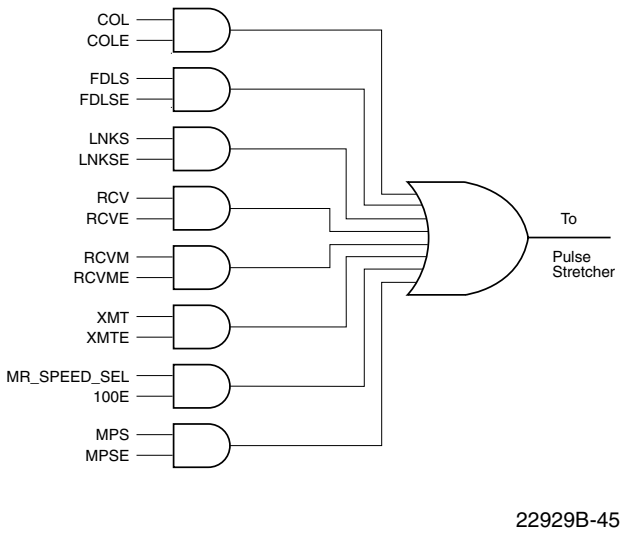


Figure 43. LED Control Logic

Power Savings Mode

Power Management Support

The Am79C976 controller supports power management as defined in the PCI Bus Power Management Interface Specification V1.1 and Network Device Class Power Management Reference Specification V1.0. These specifications define the network device power states, PCI power management interface including the Capabilities Data Structure and power management registers block definitions, power management events, and OnNow network Wake-up events. In addition, the Am79C976 controller supports legacy power management schemes, such as Remote Wake-Up (RWU) mode. The RWU mode can accommodate systems that sleep with PCI bus power off or on and the PCI clock running or stopped. The RWU pin can drive the CPU's System Management Interrupt (SMI) line or a system power controller.

The general scheme for the Am79C976 controller power management is that when a wake-up event is detected, a signal is generated to cause hardware external to the Am79C976 device to put the computer into the working (S0) mode. The Am79C976 device supports three types of wake-up events:

- Magic Packet Detect
- Link State Change
- Pattern Match Detect

The Am79C976 device supports two types of wake-up control mechanisms:

Downloaded from Arrow.com.

- PCI Bus Power Management Interface Specification (OnNow) wake-up
- RWU (hardware controlled) wake-up

All three wake-up events and both control mechanisms support wake-up from any power state including $D3_{cold}$ (PCI bus power off and clock stopped). Figure 44 shows the relationship between these Wake-up events and the various outputs used to signal to the external hardware.

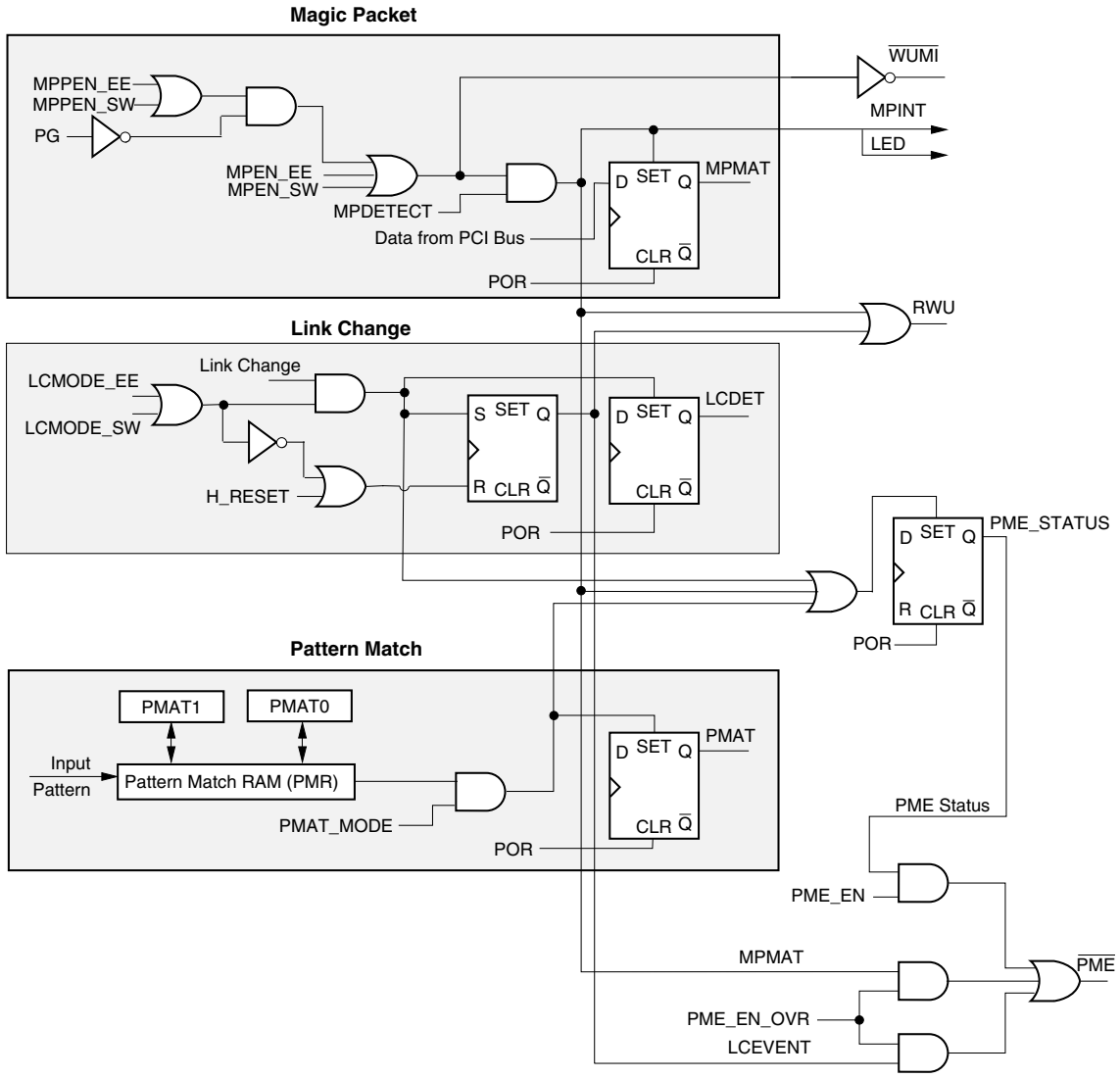


Figure 44. OnNow Functional Diagram

OnNow Wake-Up Sequence

The system software enables the $\overline{PME}$ pin by setting the PME_EN bit in the PMCSR register (PCI configuration registers, offset 48h, bit 8) to 1. When a Wake-up event is detected, the Am79C976 device sets the PME_STATUS bit in the PMCSR register (PCI configuration registers, offset 48h, bit 15). Setting this bit causes the $\overline{PME}$ signal to be asserted.

Assertion of the $\overline{PME}$ signal causes external hardware to wake up the CPU. The system software then reads the PMCSR register of every PCI device in the system to determine which device asserted the $\overline{PME}$ signal.

When the software determines that the signal came from the Am79C976 device, it writes to the device's PMCSR to put the device into power state D0. The software then writes a 0 to the PME_STATUS bit to clear the bit and turn off the $\overline{PME}$ signal, and it calls the device's software driver to tell it that the device is now in state D0. The system software can clear the PME_STATUS bit either before, after, or at the same time that it puts the device back into the D0 state.

Downloaded from Arrow.com.

The type of wake-up is configured by software using the bits LCMODE_SW, PMAT_MODE, MPEN_SW and MPPEN_SW in the CMD7 register. These bits are only reset by the power-on reset (POR) and are not loaded from the EEPROM so that they will maintain value across PCI bus resets and EEPROM read operations.

RWU Wake-Up Sequence

The RWU wake-up mechanism is used by systems that do not have software support for the PCI Bus Power Management Interface. The wake-up may be configured and controlled completely in hardware, using the EEPROM to load Am79C976 controller registers and using the PG pin to enable wake-up. Alternatively, if the PCI bus power is never removed, wake-up may be configured and enabled by software.

To accommodate systems with hardware that connects $\overline{\text{PME}}$ to the system power control logic running software that is not aware of the PCI Bus Power Management Interface, the RWU signal may be routed to the $\overline{\text{PME}}$ pin by setting the PME_EN_OVR bit (CMD3, bit 4). This is typically set by the EEPROM.

The RWU wake-up is configured by using the bits LCMODE_EE, MPEN_EE, MPPEN_EE, PME_EN_OVR, RWU_POL, RWU_DRIVER and RWU_GATE in the CMD3 register. These bits are reset by H_RESET and may be loaded from the EEPROM.

Do not use LCMODE_SW, MPEN_SW or MPPEN_SW to enable the RWU wake-up mechanism. These bits are intended for OnNow wake-up operation only.

The Pattern Match wake-up event is not supported by the RWU wake-up mechanism.

For legacy system support, the Magic Packet wake-up event may be routed to the $\overline{\text{INTA}}$ pin or to any of the four LED pins.

Link Change Detect

Link change detect is one of the Wake-up events that is defined by the OnNow specification and is supported by the RWU mode. Link Change Detect mode is set when the LCMODE_EE bit (CMD3, bit 5) is set either by software or loaded through the EEPROM or when the LCMODE_SW bit (CMD7, bit 0) is set by software.

When this bit is set, any change in the Link status will cause the LC_DET bit (STAT0, bit 10) to be set. When the LC_DET bit is set, the RWU pin will be asserted and the PME_STATUS bit (PMCSR register, bit 15) will be set. If either the PME_EN bit (PMCSR, bit 8) or the $\overline{\text{PME}}$_EN_OVR bit (CMD3, bit 4) are set, then the $\overline{\text{PME}}$ signal will also be asserted.

The Am79C976 controller may be configured to enter link change detect mode immediately upon initial power-up, regardless of the presence or absence of PCI bus power. This is accomplished by setting the LCMODE_EE bit from the EEPROM.

Magic Packet Mode

A Magic Packet is a frame that is addressed to the Am79C976 controller and contains a data sequence made up of 16 consecutive copies of the device's physical address (PADR[47:0]) anywhere in its data field. The frame must also cause an address match. By default, it must be a physical address match, but if the MPPLBA bit (CMD3, bit 9) is set, logical and broadcast address matches are also accepted. Regardless of the setting of MPPLBA, the sequence in the data field of the frame must be 16 repetitions of the Am79C976 device's physical address (PADR[47:0]).

Magic Packet mode is enabled by setting the MPEN_SW bit (CMD7, bit 1) or the MPEN_EE bit (CMD3, bit 6). Alternatively, Magic Packet mode may be enabled by setting the MPPEN_SW bit (CMD7, bit 2) or the MPPEN_EE bit (CMD3, bit 8) and deasserting the PG pin. Magic Packet mode is disabled by clearing the enable bit(s) or, if only MPPEN_EE and/or MPPEN_SW are set, by asserting PG.

The Am79C976 controller may be configured to enter Magic Packet mode immediately upon initial power-up if PCI bus power is off. This is accomplished by setting the MPPEN_EE bit from the EEPROM and enabling Magic Packet mode by the deassertion of PG.

Enabling Magic Packet mode has a similar effect to that of suspending both transmit and receive. After the FIFOs have emptied, no frames will be transmitted or received until Magic Packet mode is disabled.

The WUMI output will be asserted when Magic Packet mode is enabled.

When the Am79C976 controller detects a Magic Packet frame, it sets the MP_DET bit (STAT0, bit 11), the MPINT bit (INT0, bit 13), and the PME_STATUS bit (PMCSR, bit 15). The RWU pin will also be asserted and if the $\overline{\text{PME}}$_EN or the PME_EN_OVR bits are set, then the $\overline{\text{PME}}$ signal will be asserted as well. If INTREN (CMD0, bit 1) and MPINTEN (INTEN0, bit 13) are set to 1, $\overline{\text{INTA}}$ will be asserted. Any one of the four LED pins can be programmed to indicate that a Magic Packet frame has been received. MPSE (LED0-3, bit 9) must be set to 1 to enable that function.

Note: *The polarity of the LED pin can be programmed to be active HIGH by setting LEDPOL (LED0-3, bit 14) to 1.*

Once a Magic Packet frame is detected, the Am79C976 controller will discard the frame internally, but will not resume normal transmit and receive operations until Magic Packet mode is disabled. Once both of these events has occurred, indicating that the system has detected the Magic Packet and is awake, the controller will continue polling receive and transmit descriptor rings where it left off. It is not necessary to re-initialize the device. If the part is re-initialized, the in-

Downloaded from Arrow.com.

ternal pointers to the current descriptors will be lost, and the Am79C976 controller will not start where it left off.

If Magic Packet mode is disabled by the assertion of PG, then in order to immediately re-enable Magic Packet mode, the PG pin must remain asserted for at least 200 ns before it is deasserted. If Magic Packet mode is disabled by clearing the register bits, then it may be immediately re-enabled by setting MPEN_EE or MPEN_SW back to 1.

The PCI bus interface clock (CLK) is not required to be running while the device is operating in Magic Packet mode. Either of the INTA, the LED pins, RWU, or the PME signal may be used to indicate the receipt of a Magic Packet frame when the CLK is stopped.

OnNow Pattern Match Mode

In the OnNow Pattern Match Mode, the Am79C976 controller compares the incoming packets with up to eight patterns stored in the Pattern Match RAM (PMR). The stored patterns can be compared with part or all of incoming packets, depending on the pattern length and the way the PMR is programmed. When a pattern match has been detected, then PMAT_DET bit (STAT0, bit 12) is set. This causes the PME_STATUS bit (PMCSR, bit 15) to be set, which in turn will assert the PME pin if the PME_EN bit (PMCSR, bit 8) is set.

Pattern Match mode is enabled by setting the PMAT_MODE bit (CMD7,bit 3).The RUN bit (CMD0, bit 0) and RX_SPND bit (CMD0, bit 3) must also be set. If using the legacy registers, STRT (CSR0, bit 1) and SPND (CSR5, bit 0) must be set. Because Pattern Match mode must be configured by software, it is not possible to enable Pattern Match mode directly from the EEPROM.

Pattern Match RAM (PMR)

The PMR is organized as an array of 64 words by 40 bits as shown in Figure 45. The PMR is programmed indirectly through the PMAT0 and PMAT1 registers. For compatibility with legacy controllers, the PMR may also be programmed through BCR45, BCR46, and BCR47. Pattern Match mode must be disabled (PMAT_MODE bit cleared) to allow reading or writing the PMR.

A write access to the PMR begins with a write to the PMAT0 register. Bits 6:0 of PMAT0 specify the address in the PMR and bits 31:8 contain the data to be written to bits 23:0 of the specified address in the PMR. This is followed by a write to the PMAT1 register, with bits 15:0 of PMAT1 containing the data to be written to bits 39:24 of the specified address in the PMR. The actual write to the PMR occurs when PMAT1 is written.

A read access to the PMR also begins with a write to the PMAT0 register. Bits 6:0 of PMAT0 specify the address in the PMR to be read and the remaining bits of PMAT0 are ignored. This write is followed by a read of PMAT0, which returns bits 23:0 of PMR in bit positions 31:8 and a read of PMAT1, which returns bits 39:24 of PMR in bit positions 15:0. These reads may be done in any order.

The first two 40-bit words in the PMR serve as pointers and contain enable bits for the eight possible match patterns. The remainder of the RAM contains the match patterns and associated match pattern control bits. Byte 0 of the first word contains the Pattern Enable bits. Any bit position set in this byte enables the corresponding match pattern in the PMR. As an example, if bit 3 is set, then Pattern 3 is enabled for matching. Bytes 1 to 4 in the first word are pointers to the beginning of the patterns 0 to 3, and bytes 1 to 4 in the second word are pointers to the beginning of the patterns 4 to 7, respectively. Byte 0 of the second word has no function associated with it. Byte 0 of words 2 to 63 is the Control Field of the PMR. Bit 7 of this field is the End of Pattern (EOP) bit. When this bit is set, it indicates the end of a pattern in the PMR.

Bits 6-4 of the Control Field byte are the SKIP bits. The value of the SKIP field indicates the number of the Dwords to be skipped before the pattern in this PMR word is compared with data from the incoming frame. A maximum of seven Dwords may be skipped. Bits 3-0 of the Control Field byte are the MASK bits. These bits correspond to the pattern match bytes 3-0 of the same PMR word (PMR bytes 4-1). If bit n of this field is 0, then byte n of the corresponding pattern word is ignored. If this field is programmed to 3, then bytes 0 and 1 of the pattern match field (bytes 2 and 1 of the word) are used and bytes 3 and 2 are ignored in the pattern matching operation.

The contents of the PMR are not affected by any reset. The contents are undefined after a power-up reset (POR).

Downloaded from Arrow.com.

BCR Bit Number	BCR 47		BCR 46		BCR45	
	15 8	7 0	15 8	7 0	15 8	
	PMAT1		PMAT0			
	15 8	7 0	31 24	23 16	15 8	
	PMR_B4	PMR_B3	PMR_B2	PMR_B1	PMR_B0	
Pattern Match RAM Address	Pattern Match RAM Bit Number 39 32	31 24	23 16	15 8	7 0	Comments
0	P3 pointer	P2 pointer	P1 pointer	P0 pointer	Pattern Enable bits	First Address
1	P7 pointer	P6 pointer	P5 pointer	P4 pointer	X	Second Address
2	Data Byte 3	Data Byte 2	Data Byte1	Data Byte 0	Pattern Control	Start Pattern P_1
2+n	Data Byte 4n+3	Date Byte 4n+2	Data Byte 4n+1	Data Byte 4n+0	Pattern Control	End Pattern P_1
J	Data Byte 3	Data Byte 2	Data Byte 1	Data Byte 0	Pattern Control	Start Pattern P_k
J+m	Data Byte 4m+3	Data Byte 4m+2	Data Byte 4m+1	Data Byte 4m+0	Pattern Control	End Pattern P_k

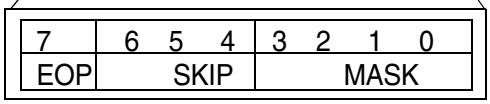


Figure 45. Pattern Match RAM

IEEE 1149.1 (1990) Test Access Port Interface

An IEEE 1149.1-compatible boundary scan Test Access Port is provided for board-level continuity test and diagnostics. All digital input, output, and input/output pins are tested. The following paragraphs summarize the IEEE 1149.1-compatible test functions implemented in the Am79C976 controller.

Boundary Scan Circuit

The boundary scan test circuit requires four pins (TCK, TMS, TDI and TDO), defined as the Test Access Port (TAP). It includes a finite state machine (FSM), an instruction register, a data register array. Internal pull-up resistors are provided for the TDI, TCK, and TMS pins.

TAP Finite State Machine

The TAP engine is a 16-state finite state machine (FSM), driven by the Test Clock (TCK), and the Test Mode Select (TMS) pins. The FSM is reset when TMS and TDI are high for five TCK periods.

Supported Instructions

In addition to the minimum IEEE 1149.1 requirements (BYPASS, EXTEST, and SAMPLE instructions), three additional instructions (IDCODE, TRIBYP, and SETBYP) are provided to further ease board-level testing. All unused instruction codes are reserved. See Table 17 for a summary of supported instructions.

Downloaded from Arrow.com.

.

Table 17. IEEE 1149.1 Supported Instruction Summary

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

Instruction Register and Decoding Logic

After the TAP FSM is reset, the IDCODE instruction is always invoked. The decoding logic gives signals to control the data flow in the Data registers according to the current instruction.

Boundary Scan Register

Each Boundary Scan Register (BSR) cell has two stages. A flip-flop and a latch are used for the Serial Shift Stage and the Parallel Output Stage, respectively. There are four possible operation modes in the BSR cell shown in Table 18.

Table 18. BSR Mode Of Operation

1	Capture
2	Shift
3	Update
4	System Function

Other Data Registers

Other data registers are the following:

1. Bypass Register (1 bit)
2. Device ID register (32 bits) (Table 19).

Table 19. Device ID Register

Bits 31-28	Version
Bits 27-12	Part Number: 0010 0110 0010 1000b (2628h)
Bits 11-1	Manufacturer ID. The 11 bit manufacturer ID code for AMD is 00000000001 in accordance with JEDEC publication 106-A.
Bit 0	Always a logic 1

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

Reset

There are five different types of RESET operations that may be performed on the Am79C976 device, H_RESET, EE_RESET, S_RESET, STOP, and POR. The following is a description of each type of RESET operation.

H_RESET

Hardware Reset (H_RESET) is an Am79C976 reset operation that has been created by the proper assertion of the $\overline{\text{RST}}$ pin of the Am79C976 device while the PG pin is HIGH. When the minimum pulse width timing as specified in the $\overline{\text{RST}}$ pin description has been satisfied, then an internal reset operation will be performed.

H_RESET will program most of the CSR and BCR registers to their default value. Note that there are several CSR and BCR registers that are undefined after H_RESET. See the sections on the individual registers for details.

H_RESET will clear most of the registers in the PCI configuration space. H_RESET will reset the internal state machines. Following the end of the H_RESET operation, the Am79C976 controller will attempt to read the EEPROM device through the EEPROM interface.

H_RESET will clear DWIO (BCR18, bit 7) and the Am79C976 controller will be in 16-bit I/O mode after the reset operation. A DWord write operation to the RDP (I/O offset 10h) must be performed to set the device into 32-bit I/O mode.

EE_RESET

Prior to starting a read of the serial EEPROM, the Am79C976 controller resets all the registers that can be programmed from the EEPROM. This provides a consistent starting point for register programming.

EE_RESET is also generated following EEPROM read if the EEPROM CRC check fails.

Downloaded from Arrow.com.

S_RESET

S_RESET is provided for compatibility with previous PCnet family devices. S_RESET occurs when the host CPU reads the Reset register, which is located at offset 14h if the device is operating in Word I/O mode or at offset 18h in DWord I/O mode.

S_RESET has the same effect as setting STOP except that S_RESET resets some Control and Status Register (CSR) bits that STOP does not change. See the descriptions of individual Control and Status Registers for details about which bits are affected. S_RESET does not trigger an automatic EEPROM read sequence.

New software should not use S_RESET. It should be replaced by a combination of clearing the RUN bit in the CMD0 register followed by explicit setting or clearing of control bits as required.

STOP

A STOP reset is generated by the assertion of the STOP bit in CSR0. Writing a 1 to the STOP bit of CSR0, when the stop bit currently has a value of 0, will initiate a STOP reset. If the STOP bit is already a 1, then writing a 1 to the STOP bit will not generate a STOP reset.

STOP will reset all or some portions of CSR0, 3, and 4 to default values. For the identity of individual CSRs and bit locations that are affected by STOP, see the individual CSR register descriptions. STOP will not affect any of the BCR and PCI configuration space locations. STOP will reset the internal state machines. Following the end of the STOP operation, the Am79C976 controller will not attempt to read the EEPROM device.

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

Power on Reset

Power on Reset (POR) is generated when the Am79C976 controller is powered up. POR generates a hardware reset (H_RESET). In addition, it clears some bits that H_RESET does not affect.

External PHY Reset

The Am79C976 controller provides the PHY_RST pin which may be connected to the reset input of an external PHY. The polarity of PHY_RST is determined by the RST_POL bit in CMD3 (RST_POL in CSR116). The PHY_RST pin will assert at the start of the read of the serial EEPROM and will deassert at least 240µs (the duration of the read of two bytes from the EEPROM) before the end of the serial EEPROM read, providing time for the PHY to recover from the reset.

The RST_POL bit may be programmed from the serial EEPROM. The default value is zero, corresponding to an active high PHY_RST signal. If an active low PHY_RST is required, the CMD3 register should be the first register programmed from the serial EEPROM.

The duration of the assertion of PHY_RST depends on the number of registers programmed by the serial EEPROM. Each register requires at least 240 µs. The time to program the CMD3 register and any registers programmed before CMD3 should be ignored in the calculation of PHY_RST duration if the RST_POL bit is programmed to 1.

If the number of registers programmed from the serial EEPROM results in PHY_RST being too short, the read-only register at offset 0x28 may be used for padding. Specify the address as 0x14 with arbitrary data and repeat as many times as necessary to achieve the required PHY_RST duration.

If the serial EEPROM is not used, the PHY may be reset by BIOS or driver software by programming the correct RST_POL value, disabling the internal port manager by setting the DISPM bit, disabling the auto-poll logic by clearing the APEP bit and then setting the RESET_PHY bit. All these bits are in the CMD3 register.

The PHY_RST will be asserted as long as RESET_PHY remains set. If the PHY requires a recovery time after reset, the software must provide the delay after clearing the RESET_PHY bit before accessing the PHY's registers or enabling the Am79C976 controller's port manager and/or auto-poll logic.

Downloaded from Arrow.com.

Software Access

PCI Configuration Registers

The Am79C976 controller implements the 256-byte configuration space as defined by the PCI specification revision 2.1. The 64-byte header includes all registers required to identify the Am79C976 controller and its function. Additionally, the optional PCI Power Management Interface registers are implemented at location 44h - 4Bh. The layout of the Am79C976 PCI configuration space is shown in Table 20.

The PCI configuration registers are accessible only by configuration cycles. All multi-byte numeric fields follow Little Endian byte ordering. All write accesses to Reserved locations have no effect; reads from these locations will return a data value of 0.

I/O Resources

The Am79C976 controller requires 4K bytes of memory address space for access to all the various internal registers as well as access to some setup information stored in an external serial EEPROM. For compatibility with previous PCnet family devices, the lower 32 bytes of the register space are also mapped into I/O space, but some functions of the Am79C976 controller (such as network statistics) are only available in memory space.

Table 20. PCI Configuration Space Layout

31 24	23 16	15 8	7 0	Offset
Device ID		Vendor ID		00h
Status		Command		04h
Base-Class	Sub-Class	Programming IF	Revision ID	08h
Reserved	Header Type	Latency Timer	Cache Line Size	0Ch
I/O Base Address				10h
Memory-Mapped I/O Base Address				14h
Reserved				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address				30h
Reserved			CAP-PTR	34h
Reserved				38h
MAX_LAT	MIN_GNT	Interrupt Pin	Interrupt Line	3Ch
Reserved				40h
PMC		NXT_ITM_PTR	CAP_ID	44h
DATA_REG	PMCSR_BSE	PMCSR		48h
Reserved				. .
Reserved				FCh

The Am79C976 controller supports mapping the address space to both I/O and memory space. The value in the PCI I/O Base Address register determines the start address of the I/O address space. The register is typically programmed by the PCI configuration utility after system power-up. The PCI configuration utility must also set the IOEN bit in the PCI Command register to enable I/O accesses to the Am79C976 controller. For memory mapped I/O access, the PCI Memory Mapped I/O Base Address register controls the start

Downloaded from Arrow.com.

address of the memory space. The MEMEN bit in the PCI Command register must also be set to enable the mode. Both base address registers can be active at the same time.

The Am79C976 controller requires that the memory space that it claims must be within the 32-bit address space.

The Am79C976 controller supports two modes for accessing the I/O resources. For backwards compatibility with AMD's 16-bit Ethernet controllers, Word I/O is the default mode after power up. The device can be configured to DWord I/O mode by software.

I/O Registers

The Am79C976 controller registers are divided into three groups: memory-mapped registers, Control and Status Registers (CSRs), and Bus Control Registers (BCRs). The CSRs and BCRs are included in the Am79C976 device for software compatibility with older PCnet family controllers that do not have the memory-mapped register group. The CSRs and BCRs are addressed indirectly through the Register Address Port (RAP), Register Data Port (RDP), and BCR Data Port (BDP) so that a large number of functions can be controlled through a small amount of I/O space.

All CSR and BCR functions can be accessed more efficiently through the memory-mapped registers. The memory-mapped registers are directly addressed as offsets from the address stored in the Memory Mapped I/O Base Address Register, which is located in PCI Configuration Space.

The Control and Status Registers (CSR) are used to configure the Ethernet MAC engine and to obtain status information. The Bus Control Registers (BCR) are used to configure the bus interface unit and the LEDs. Both sets of registers are accessed using indirect addressing.

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

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

Once the RAP has been written with a value, the RAP value remains unchanged until another RAP write occurs, or until an H_RESET or S_RESET occurs. RAP is cleared to all 0s when an H_RESET or S_RESET occurs. RAP is unaffected by setting the STOP bit.

Address PROM Space

Software written for early PCnet family products expects the 48-bit IEEE MAC address to be found in a small external PROM that occupies the first 16 bytes of the controller's I/O address space. The software copies this address from the PROM into the Initialization Block in host memory. For compatibility with this software the Am79C976 controller contains 16-bytes of read/write memory starting at offset 0 in the device's I/O or memory address space. This 16 bytes of space is known as the Address PROM Space (APROM). If compatibility with this software is required, the EEPROM should load the data shown in Table 21 into the Address PROM Space. The order of bytes in the MAC address is such that the first byte transmitted is located at offset 0.

Table 21. Address PROM Space Contents

Offset	Contents
00h-05h	MAC Address
06h-0Bh	00 00 00 00 00 00h
0Ch-0Dh	Check sum of bytes 0-11 and bytes 14-15
0Eh-0Fh	ASCII "W" (57h)

CPU access to the APROM space is controlled by the APROM Write Enable (APROMWE) bit (BCR2, bit 8). If this bit is cleared to 0, the host CPU can not write to the APROM space. However, the APROM space can be loaded from the EEPROM regardless of the state of APROMWE.

Reset Register

A read of the Reset register creates an internal software reset (S_RESET) pulse in the Am79C976 controller. The effect of this reset is the same as that of setting the STOP bit, except that S_RESET clears some bits that are not affected by setting STOP.

This register exists for backward compatibility with earlier PCnet family devices. It should not be used in new software.

The NE2100 LANCE-based family of Ethernet cards requires that a write access to the Reset register follows each read access to the Reset register. The Am79C976 controller does not have a similar requirement. The write access is not required and does not have any effect.

Downloaded from Arrow.com.

Word I/O Mode

After H_RESET, the Am79C976 controller is programmed to operate in Word I/O mode. DWIO (BCR18, bit 7) will be cleared to 0. Table 22 shows how the 32 bytes of address space are used in Word I/O mode.

All I/O resources must be accessed in word quantities and on word addresses. The Address PROM locations can also be read in byte quantities. The only allowed DWord operation is a write access to the RDP, which switches the device to DWord I/O mode. A read access other than those listed in Table 22 will yield undefined data, and a write operation may cause unexpected reprogramming of the Am79C976 control registers. Table 23 shows legal I/O accesses in Word I/O mode.

Table 22. I/O Map In Word I/O Mode (DWIO = 0)

Offset	No. of Bytes	Register
00h - 0Fh	16	APROM
10h	2	RDP
12h	2	RAP (shared by RDP and BDP)
14h	2	Reset Register
16h	2	BDP
18h - 1Fh	8	Reserved
20h-1FFh	224	Memory-mapped Registers
*	256	MIB Counters

Note: ** The offset of a MIB counter is the value contained in the MIB Offset Register plus the offset shown in Table 7 or Table 8.*

Double Word I/O Mode

The Am79C976 controller can be configured to operate in DWord (32-bit) I/O mode. The software can invoke the DWIO mode by performing a DWord write access to the I/O location at offset 10h (RDP). The data of the write access must be such that it does not affect the intended operation of the Am79C976 controller. Setting the device into 32-bit I/O mode is usually the first operation after H_RESET or S_RESET. The RAP register will point to CSR0 at that time. Writing a value of 0 to CSR0 is a safe operation. DWIO (BCR18, bit 7) will be set to 1 as an indication that the Am79C976 controller operates in 32-bit I/O mode.

Note: *Even though the I/O resource mapping changes when the I/O mode setting changes, the RDP location offset is the same for both modes. Once the DWIO bit has been set to 1, only H_RESET can clear it to 0. The DWIO mode setting is unaffected by S_RESET or setting of the STOP bit. Table 24 shows how the 32 bytes of address space are used in DWord I/O mode.*

All I/O resources must be accessed in DWord quantities and on DWord addresses. A read access other than listed in Table 25 will yield undefined data, and a write operation may cause unexpected reprogramming of the Am79C976 control registers.

DWIO mode applies to both I/O- and memory-mapped accesses. Either a 32-bit I/O write to offset 10h relative to the contents of the I/O Base Address Register (BAR) or a 32-bit memory write to offset 10h relative to the contents of the Memory BAR puts the device into DWIO mode. Once in DWIO mode, the offsets of the RAP, Reset, and BDP registers are 14h, 18h, and 1Ch relative to the contents of either the I/O BAR or the Memory BAR.

For example, in DWIO mode the BDP Register can be read either by an I/O read from offset 1Ch relative to the contents of the I/O BAR or by a memory read from offset 1Ch relative to the contents of the Memory BAR.

DWIO mode affects only the locations between 0 and 1Fh, which include the RAP, Reset, and BDP registers. Registers at offset 20h and above are directly accessed in memory space by byte address and are unaffected by DWIO.

The DWIO bit is also located at bit 28 of the CMD2 register and can be set or cleared by software regardless of its current setting.

Downloaded from Arrow.com.

Table 23. Legal I/O Accesses in Word I/O Mode (DWIO = 0)

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

Table 24. I/O Map In DWord I/O Mode (DWIO = 1)

Offset	No. of Bytes	Register
00h - 0Fh	16	APROM
10h	4	RDP
14h	4	RAP (shared by RDP and BDP)
18h	4	Reset Register
1Ch	4	BDP
20h-1FFh	224	Memory-mapped Registers
200h-2FFh	256	MIB Counters

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

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

Downloaded from Arrow.com.

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

10100	0000	RD	DWord read of RAP
11000	0000	RD	DWord read of Reset Register

Downloaded from Arrow.com.

USER ACCESSIBLE REGISTERS

The Am79C976 controller has four types of user registers: the PCI configuration registers, the memory-mapped registers, the Control and Status registers (CSRs), and the Bus Control registers (BCRs). The CSRs and BCRs are included for software compatibility with older PCnet family devices. However, all CSR and BCR functions can be accessed more efficiently through the memory-mapped registers. Software written for the Am79C976 device and future PCnet family devices does not have to access any CSR or BCR.

The Am79C976 controller implements all PCnet-ISA (Am79C960) registers, all C-LANCE (Am79C90) registers, all PCnet-FAST (Am79C971) registers, plus a number of additional registers. The Am79C976 CSRs are compatible upon power up with both the PCnet-ISA CSRs and all of the C-LANCE CSRs.

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

The Am79C976 controller's registers can be divided into several functional groups: PCI Configuration Alias registers, PCI Configuration registers, Setup registers, Running registers, and Test registers.

The PCI Configuration Alias registers are typically programmed through the EEPROM read operation. In this way, they are initialized before the BIOS accesses the PCI Configuration registers. This group includes the MAX_LAT_A, MIN_GNT_A, PCIDATA0 - 7, PMC_A, SID_A, SVID_A and VID_A registers and the ROM_CFG register.

The PCI Configuration registers are accessed by the system BIOS software to configure the Am79C976 controller. These registers include the Memory Base Address register, the Expansion ROM Base Address register, the Interrupt Line register, the PCI Command register, the PCI Status register and the PMC register. Typically, device information will also be read from the SID, SVID and VID registers.

The Setup registers include most of the remaining memory mapped registers. The programming of these is typically divided between the EEPROM and the driver software. Many of these registers are optional and need not be initialized unless the associated function is used.

Typically, the SRAM_SIZE, SRAM_BND and PADR registers are initialized from the EEPROM. The driver initializes the BADR, BADX, RCV_RING_LEN, XMT_RING_LEN and LADRF registers. The CMD2, CMD3, CTRL0, CTRL1 and CTRL2 registers are typically partially initialized from the EEPROM and partially by the driver. The CMD7 and CTRL3 registers are initialized by the driver. The driver reads the MIB_OFFSET and CHIPID registers at initialization time. The PHY_ACCESS register may be used at this time to initialize the external PHY.

Running registers are accessed by the driver software. These include the CMD0, INT0, INTEN0, STAT0, LADRF and MIB registers. The driver may access other registers during operation depending on the features being used.

PCI Configuration Registers

PCI Vendor ID Register

Offset 00h

The PCI Vendor ID register is a 16-bit register that identifies the manufacturer of the Am79C976 controller. AMD's Vendor ID is 1022h. Note that this vendor ID is not the same as the Manufacturer ID in CSR88 and CSR89. The vendor ID is assigned by the PCI Special Interest Group.

The PCI Vendor ID register is read only.

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

PCI Device ID Register

Offset 02h

The PCI Device ID register is a 16-bit register that uniquely identifies the Am79C976 controller within AMD's product line. The Am79C976 Device ID is 2000h. Note that this Device ID is not the same as the Part number in CSR88 and CSR89. The Device ID is assigned by AMD. The Device ID is the same as the PCnet-PCI II (Am79C970A) and PCnet-*FAST* (Am79C971) devices.

The PCI Device ID register is read only.

PCI Command Register

Offset 04h

The PCI Command register is a 16-bit register used to control the gross functionality of the Am79C976 controller. It controls the Am79C976 controller's ability to generate and respond to PCI bus cycles. To logically disconnect the Am79C976 device from all PCI bus cycles except configuration cycles, a value of 0 should be written to this register.

Downloaded from Arrow.com.

The PCI Command register is read and written by the host.

Bit	Name	Description
15-10	RES	Reserved locations. Read as zeros; write operations have no effect.
9	FBTBEN	Fast Back-to-Back Enable. When this bit is set to 1, the Am79C976 controller will generate Fast Back-to-Back cycles. When this bit is cleared to 0, the Am79C976 controller will not generate Fast Back-to-Back cycles. FBTBEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.
8	SERREN	SERR Enable. Controls the assertion of the SERR pin. SERR is disabled when SERREN is cleared. SERR will be asserted on detection of an address parity error and if both SERREN and PERREN (bit 6 of this register) are set. SERREN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.
7	RES	Reserved location. Read as zeros; write operations have no effect.
6	PERREN	Parity Error Response Enable. Enables the parity error response functions. When PERREN is 0 and the Am79C976 controller detects a parity error, it only sets the Detected Parity Error bit in the PCI Status register. When PERREN is 1, the Am79C976 controller asserts PERR on the detection of a data parity error. It also sets the DATAPERR bit (PCI Status register, bit 8), when the data parity error occurred during a master cycle. PERREN also enables reporting address parity errors through the SERR pin and the SERR bit in the PCI Status register. PERREN is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
5	VGASNOOP	VGA Palette Snoop. Read as zero; write operations have no effect.
4	MWIEN	Memory Write and Invalidate Cycle Enable. When this bit is set to 1, the Am79C976 controller will generate Memory Write and Invalidate (MWI) cycles when appropriate. When the bit is cleared to 0, the device will generate Memory Write cycles instead of MWI cycles. MWIEN is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
3	SCYCEN	Special Cycle Enable. Read as zero; write operations have no effect. The Am79C976 controller ignores all Special Cycle operations.
2	BMEN	Bus Master Enable. Setting BMEN enables the Am79C976 controller to become a bus master on the PCI bus. The host must set BMEN before setting the INIT or STRT bit in CSR0 of the Am79C976 controller. BMEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.
1	MEMEN	Memory Space Access Enable. The Am79C976 controller will ignore all memory accesses when MEMEN is cleared. The host must set MEMEN before the first memory access to the device. For memory mapped I/O, the host must program the PCI Memory Mapped I/O Base Address register with a valid memory address before setting MEMEN. For accesses to the Expansion ROM, the host must program the PCI Expansion ROM Base Address register at offset 30h with a valid memory address before set-

Downloaded from Arrow.com.

ting MEMEN. The Am79C976 controller will only respond to accesses to the Expansion ROM when both ROMEN (PCI Expansion ROM Base Address register, bit 0) and MEMEN are set to 1. Since MEMEN also enables the memory mapped access to the Am79C976 I/O resources, the PCI Memory Mapped I/O Base Address register must be programmed with an address so that the device does not claim cycles not intended for it.

MEMEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.

0 IOEN I/O Space Access Enable. The Am79C976 controller will ignore all I/O accesses when IOEN is cleared. The host must set IOEN before the first I/O access to the device. The PCI I/O Base Address register must be programmed with a valid I/O address before setting IOEN.

IOEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.

PCI Status Register

Offset 06h

The PCI Status register is a 16-bit register that contains status information for the PCI bus related events.

Bit	Name	Description
15	PERR	Parity Error. PERR is set when the Am79C976 controller detects a parity error. The Am79C976 controller samples the AD[31:0], C/$\overline{BE}$[3:0], and the PAR lines for a parity error at the following times: • In slave mode, during the address phase of any PCI bus command. • In slave mode, for all I/O, memory and configuration write commands that select the Am79C976 controller when data is transferred ($\overline{TRDY}$ and $\overline{IRDY}$ are asserted). • In master mode, during the data phase of all memory read commands. In master mode, during the data phase of the memory write command, the Am79C976 controller sets the PERR bit if the target reports a data parity error by asserting the $\overline{PERR}$ signal. PERR is not effected by the state of the Parity Error Response enable bit (PCI Command register, bit 6). PERR is set by the Am79C976 controller and cleared by writing a 1. Writing a 0 has no effect. PERR is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
14	SERR	Signaled SERR. SERR is set when the Am79C976 controller detects an address parity error and both SERREN and PERREN (PCI Command register, bits 8 and 6) are set. SERR is set by the Am79C976 controller and cleared by writing a 1. Writing a 0 has no effect. SERR is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
13	RMABORT	Received Master Abort. RMABORT is set when the Am79C976 controller terminates a master cycle with a master abort sequence. RMABORT is set by the Am79C976 controller and cleared by writing a 1. Writing a 0 has no effect. RMABORT is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
12	RTABORT	Received Target Abort. RTABORT is set when a target terminates an Am79C976 master cycle with a target abort sequence. RTABORT is set by the Am79C976 controller and

Downloaded from Arrow.com.

cleared by writing a 1. Writing a 0 has no effect. RTABORT is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

11 STABORT Send Target Abort. Read as zero; write operations have no effect. The Am79C976 controller will never terminate a slave access with a target abort sequence.

STABORT is read only.

10-9 DEVSEL Device Select Timing. DEVSEL is set to 01b (medium), which means that the Am79C976 controller will assert DEVSEL two clock periods after FRAME is asserted.

DEVSEL is read only.

8 DATAPERR Data Parity Error Detected. DATAPERR is set when the Am79C976 controller is the current bus master and it detects a data parity error and the Parity Error Response enable bit (PCI Command register, bit 6) is set.

During the data phase of all memory read commands, the Am79C976 controller checks for parity error by sampling the AD[31:0] and C/BE[3:0] and the PAR lines. During the data phase of all memory write commands, the Am79C976 controller checks the PERR input to detect whether the target has reported a parity error.

DATAPERR is set by the Am79C976 controller and cleared by writing a 1. Writing a 0 has no effect. DATAPERR is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

7 FBTBC Fast Back-To-Back Capable. Read as one; write operations have no effect. The Am79C976 controller is capable of accepting fast back-to-back transactions with the first transaction addressing a different target.

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

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

Read as one; write operations have no effect. The Am79C976 controller supports the Linked Additional Capabilities List.

3-0 RES Reserved locations. Read as zero; write operations have no effect.

PCI Revision ID Register

Offset 08h

The PCI Revision ID register is an 8-bit register that specifies the Am79C976 controller revision number. The value of this register is 5Xh with the lower four bits being silicon-revision dependent.

The PCI Revision ID register is read only.

PCI Programming Interface Register

Offset 09h

The PCI Programming Interface register is an 8-bit register that identifies the programming interface of Am79C976 controller. PCI does not define any specific register-level programming interfaces for network devices. The value of this register is 00h.

The PCI Programming Interface register is read only.

PCI Sub-Class Register

Offset 0Ah

The PCI Sub-Class register is an 8-bit register that identifies specifically the function of the Am79C976 controller. The value of this register is 00h which identifies the Am79C976 device as an Ethernet controller.

The PCI Sub-Class register is read only.

PCI Base-Class Register

Offset 0Bh

The PCI Base-Class register is an 8-bit register that broadly classifies the function of the Am79C976 controller. The value of this register is 02h which classifies the Am79C976 device as a network controller.

The PCI Base-Class register is read only.

Downloaded from Arrow.com.

PCI Cache Line Size Register

Offset 0Ch

This register indicates the system cache line size in units of 32-bit double words. This quantity is used to determine when to use the advanced PCI bus commands (MWI, MRL, and MRM). It is also used for aligning PCI burst transfers to cache line boundaries. Only the values 4, 8, and 16 are acceptable. If the host CPU attempts to write any other value to this location, the contents of the register will be set to 0.

This register is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.

PCI Latency Timer Register

Offset 0Dh

The PCI Latency Timer register is an 8-bit register that specifies the minimum guaranteed time the Am79C976 controller will control the bus once it starts its bus mastership period. The time is measured in clock cycles. Every time the Am79C976 controller asserts FRAME at the beginning of a bus mastership period, it will copy the value of the PCI Latency Timer register into a counter and start counting down. The counter will freeze at 0. When the system arbiter removes GNT while the counter is non-zero, the Am79C976 controller will continue with its data transfers. It will only release the bus when the counter has reached 0.

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

The six most significant bits of the PCI Latency Timer register are programmable. The two least significant bits are fixed at 0. The host should read the Am79C976 PCI MIN_GNT and PCI MAX_LAT registers to determine the latency requirements for the device and then initialize the Latency Timer register with an appropriate value.

The PCI Latency Timer register is read and written by the host. The PCI Latency Timer register is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

PCI Header Type Register

Offset 0Eh

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

Bit	Name	Description
7	FUNCT	Single-function/multi-function device. Read as zero; write operations have no effect. The Am79C976 controller is a single function device.
6-0	LAYOUT	PCI configuration space layout. Read as zeros; write operations have no effect. The layout of the PCI configuration space locations 10h to 3Ch is as shown in the table at the beginning of this section.

PCI I/O Base Address Register

Offset 10h

The PCI I/O Base Address register is a 32-bit register that determines the location of the Am79C976 I/O resources in all of I/O space.

Bit	Name	Description
31-5	IOBASE	I/O base address most significant 27 bits. These bits are written by the host to specify the location of the Am79C976 I/O resources in all of I/O space. IOBASE must be written with a valid address before the Am79C976 controller slave I/O mode is turned on by setting the IOEN bit (PCI Command register, bit 0). When the Am79C976 controller is enabled for I/O mode (IOEN is set), it monitors the PCI bus for a valid I/O command. If the value on AD[31:5] during the address phase of the cycles matches the value of IOBASE, the Am79C976 controller will drive DEVSEL indicating it will respond to the access. IOBASE is read and written by the host. IOBASE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
4-2	IOSIZE	I/O size requirements. Read as zeros; write operations have no effect. IOSIZE indicates the size of the I/O space the Am79C976 control-

Downloaded from Arrow.com.

ler requires. When the host writes a value of FFFF FFFFh to the I/O Base Address register, it will read back a value of 0 in bits 4-2. That indicates an Am79C976 I/O space requirement of 32 bytes.

1 RES — Reserved location. Read as zero; write operations have no effect.

0 IOSPACE — I/O space indicator. Read as one; write operations have no effect. Indicating that this base address register describes an I/O base address.

PCI Memory Mapped I/O Base Address Register

Offset 14h

The PCI Memory Mapped I/O Base Address register is a 32-bit register that determines the location of the Am79C976 I/O resources. Memory space claimed by the Am79C976 device may be mapped anywhere in 32-bit memory space.

Bit	Name	Description
31-12	MEMBASE	Memory mapped I/O base address, bits 31-12. These bits are written by the host to specify the location of the Am79C976 I/O resources in 32-bit memory space. MEMBASE must be written with a valid address before the Am79C976 controller slave memory mapped I/O mode is turned on by setting the MEMEN bit (PCI Command register, bit 1). When the Am79C976 controller is enabled for memory mapped I/O mode (MEMEN is set), it monitors the PCI bus for a valid memory command. If the value on AD[31:12] during the address phase of the cycles matches the value of MEMBASE, the Am79C976 controller will drive DEVSEL indicating it will respond to the access. MEMBASE is read and written by the host. MEMBASE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
11-4	MEMSIZE	Memory mapped I/O size requirements. Read as zeros; write operations have no effect. MEMSIZE indicates the size of the memory space the Am79C976 controller requires. When the host writes a value of FFFF FFFFh to the Memory Mapped I/O Base Address register, it will read back 0s in bit 11:4 to indicate an Am79C976 memory space requirement of 4K bytes.
3	PREFETCH	Prefetchable. The value of this read-only bit is the inverse of the value of the Disable Prefetchability (PREFETCH_DIS) bit (CMD3, bit 30). PREFETCH_DIS is normally loaded from EEPROM. Setting PREFETCH to 1 indicates that the memory space claimed by this device can be prefetched. Because of the side effects of reading the Reset Register at offset 14h or 18h (depending on the state of DWIO (CMD2, bit 28)), locations at offsets less than 20h cannot be prefetched. The Am79C976 device will disconnect any attempted burst transfer at offsets less than 20h. This bit is read-only. (However, its value is the inverse of PREFETCH_DIS, which can be loaded from EEPROM.)
2-1	TYPE	Memory type indicator. Read as zeros; write operations have no effect. Indicates that this base address register is 32 bits wide and mapping can be done anywhere in the 32-bit memory space.
0	MEMSPACE	Memory space indicator. Read as zero; write operations have no effect. Indicates that this base address register describes a memory base address.

PCI Subsystem Vendor ID Register

Offset 2Ch

The PCI Subsystem Vendor ID register is a 16-bit register that together with the PCI Subsystem ID uniquely identifies the add-in card or subsystem the Am79C976 controller is used in. Subsystem Vendor IDs can be obtained from the PCI SIG. A value of 0 (the default) indicates that the Am79C976 controller does not support subsystem identification. The PCI Subsystem Vendor

Downloaded from Arrow.com.

ID is an alias of BCR23, bits 15-0. It is programmable through the EEPROM.

The PCI Subsystem Vendor ID register is read only.

PCI Subsystem ID Register

Offset 2Eh

The PCI Subsystem ID register is a 16-bit register that together with the PCI Subsystem Vendor ID uniquely identifies the add-in card or subsystem the Am79C976 controller is used in. The value of the Subsystem ID is up to the system vendor. A value of 0 (the default) indicates that the Am79C976 controller does not support subsystem identification. The PCI Subsystem ID is an alias of BCR24, bits 15-0. It is programmable through the EEPROM.

The PCI Subsystem ID register is read only.

PCI Expansion ROM Base Address Register

Offset 30h

The PCI Expansion ROM Base Address register is a 32-bit register that defines the base address, size and address alignment of an Expansion ROM. The host CPU can determine the size and alignment requirements of the ROM by writing all 1s to this register, reading back the result, and masking out the least significant bit. The device will return 0s in all don't care bits.

The Am79C976 device supports ROMs ranging in size from 2 kbytes to 16 Mbytes. The amount of address space claimed by the ROM can be programmed through the ROM Configuration register, which can be loaded from the external serial EEPROM. The bits of the ROM Configuration Register (ROM_CFG) act as write enable bits for bits [23:11] and bit 0 of this register. See Figure 46.

As an example, assume that ROM_CFG is programmed to 0F001h. This means that bits [23:20] and bit 0 of ROMBASE are write enabled, and bits [19:11] are fixed at 0. In addition, bits [31:24] of ROMBASE are always write enabled, and bits [10:1] are always 0, regardless of the contents of ROM_CFG. Therefore, when the host CPU writes all 1s to ROMBASE, it will read back 0FFF00001h. Masking out bit 0 leaves 0FFF00000h. This means that bits [19:0] of the base address are don't cares, and the ROM occupies 2^{20} or 1 Mbytes of address space, and must be mapped to a 1 Mbyte boundary. If the CPU then writes 00300001h to this register and sets the MEMEN bit (PCI Command register, bit 1) to enable memory accesses, the ROM will claim the memory space between 00300000h and 003FFFFFh.

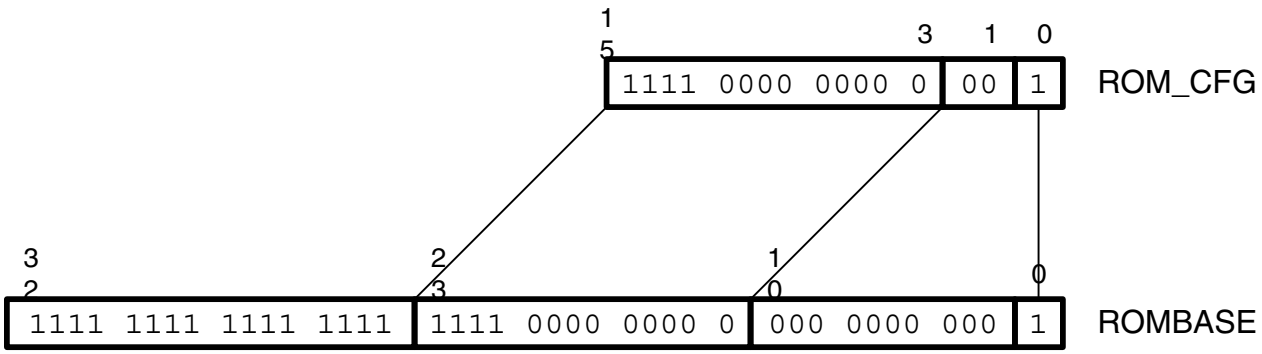


Figure 46. PCI Expansion ROM Base Address Register

Note: *The procedure described in the PCI Expansion ROM Base Address Register section specifies the amount of address space that the ROM claims. The ROM may be smaller than the amount of address space claimed. The actual size of the code in the Expansion ROM is always determined by reading the Expansion ROM header.*

Downloaded from Arrow.com.

Bit	Name	Description
31-24	ROMBASE	Expansion ROM base address most significant 8 bits. These bits are written by the host to specify the location of the Expansion ROM in PCI memory space. ROMBASE must be written with a valid address before the Am79C976 Expansion ROM access is enabled by setting ROMEN (PCI Expansion ROM Base Address register, bit 0) and MEMEN (PCI Command register, bit 1). When the Am79C976 controller is enabled for Expansion ROM access (ROMEN and MEMEN are set to 1), it monitors the PCI bus for a valid memory command. If the value on AD[31:2] during the address phase of the cycle falls in the address space specified by the contents of this register, the Am79C976 controller will drive DEVSEL indicating it will respond to the access. ROMBASE is read and written by the host. ROMBASE is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.
23-11	ROMSIZE	Expansion ROM base address bits [23:11]. Those bits in this field that are enabled by the corresponding bits in the ROM Configuration Register (ROM_CFG) are written by the host CPU to specify the location of the Expansion ROM in PCI memory space. Those bits in this field that are not enabled by the corresponding bits in ROM_CFG are fixed at 0.
10-1	ZEROS	Read as zeros; write operation has no effect.
0	ROMEN	Expansion ROM Enable. Written by the host to enable access to the Expansion ROM. The Am79C976 controller will only respond to accesses to the Expansion ROM when both ROMEN and MEMEN (PCI Command register, bit 1) are set to 1. This bit can be set to 1 only when bit 0 of ROM_CFG is set to 1. When bit 0 of ROM_CFG is cleared to 0, ROMEN is fixed a 0, and the Expansion ROM cannot be mapped into PCI memory space. ROMEN is read and written by the host. ROMEN is cleared by H_RESET and is not effected by S_RESET or by setting the STOP bit.

PCI Capabilities Pointer Register

Offset 34h

Bit	Name	Description
7-0	CAP_PTR	The PCI Capabilities pointer Register is a read-only 8-bit register that points to a linked list of capabilities implemented on this device. This register has the value 44h. The PCI Capabilities register is read only.

PCI Interrupt Line Register

Offset 3Ch

The PCI Interrupt Line register is an 8-bit register that is used to communicate the routing of the interrupt. This register is written by the POST software as it initializes the Am79C976 controller in the system. The register is read by the network driver to determine the interrupt channel which the POST software has assigned to the Am79C976 controller. The PCI Interrupt Line register is not modified by the Am79C976 controller. It has no effect on the operation of the device.

The PCI Interrupt Line register is read and written by the host. It is cleared by H_RESET and is not affected by S_RESET or by setting the STOP bit.

PCI Interrupt Pin Register

Offset 3Dh

This PCI Interrupt Pin register is an 8-bit register that indicates the interrupt pin that the Am79C976 controller is using. The value for the Am79C976 Interrupt Pin register is 01h, which corresponds to INTA.

The PCI Interrupt Pin register is read only.

PCI MIN_GNT Register

Offset 3Eh

The PCI MIN_GNT register is an 8-bit register that specifies the minimum length of a burst period that the Am79C976 device needs to keep up with the network activity. The length of the burst period is calculated assuming a clock rate of 33 MHz. The register value specifies the time in units of 1/4 µs. The PCI MIN_GNT register is an alias of the Minimum Grant Shadow Reg-

Downloaded from Arrow.com.

ister, which can be loaded from the serial EEPROM. It is recommended that the shadow register be programmed to a value of 18h, which corresponds to 6 µs.

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

The PCI MIN_GNT register is read only.

PCI MAX_LAT Register

Offset 3Fh

The PCI MAX_LAT register is an 8-bit register that specifies the maximum arbitration latency the Am79C976 controller can sustain without causing problems to the network activity. The register value specifies the time in units of 1/4 µs. The MAX_LAT register is an alias of the Maximum Latency Shadow Register, which can be loaded from the serial EEPROM. It is recommended that the shadow register be programmed to a value of 18h, which corresponds to 6 µs.

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

The PCI MAX_LAT register is read only

PCI Capability Identifier Register

Offset 44h

Bit	Name	Description
7-0	CAP_ID	This register identifies the linked list item as being the PCI Power Management registers. This is a read-only register whose value is fixed at 1h.

PCI Next Item Pointer Register

Offset 45h

Bit	Name	Description
7-0	NXT_ITM_PTR	The Next Item Pointer Register points to the starting address of the next capability. The pointer at this offset is a null pointer, indicating that this is the last capability in the linked list of the capabilities. This is a read-only register whose content is fixed at 0.

PCI Power Management Capabilities Register (PMC)

Offset 46h

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

Bit	Name	Description
15-11	PME_SPT	PME Support. This 5-bit field indicates the power states in which the function may assert $\overline{PME}$. A value of 0b for any bit indicates that the function is not capable of asserting the $\overline{PME}$ signal while in that power state. Bit(11) XXXX1b - $\overline{PME}$ can be asserted from D0. Bit(12) XXX1Xb - $\overline{PME}$ can be asserted from D1. Bit(13) XX1XXb - $\overline{PME}$ can be asserted from D2. Bit(14) X1XXXb - $\overline{PME}$ can be asserted from D3hot. Bit(15) 1XXXXb - $\overline{PME}$ can be asserted from D3cold. The value read from bit(15) is the AND of the value of BCR36, bit 15 and the current state of the VAUX_SENSE pin. Read only.
10	D2_SPT	D2 Support. If this bit is a 1, this function supports the D2 Power Management State. Read only.
9	D1_SPT	D1 Support. If this bit is a 1, this function supports the D1 Power Management State. Read only.
8-6	RES	Reserved locations. Written and read as zeros.
5	DSI	Device Specific Initialization. When this bit is 1, it indicates that special initialization of the function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. Read only.
4	RES	Reserved location. Written and read as zero.

Downloaded from Arrow.com.

3	PME_CLK	PME Clock. When this bit is a 1, it indicates that the function relies on the presence of the PCI clock for PME operation. When this bit is a 0 it indicates that no PCI clock is required for the function to generate PME. Functions that do not support PME generation in any state must return 0 for this field. Read only.
2-0	PMIS_VER	Power Management Interface Specification Version. A value of 010b indicates that this function complies with the revision 1.1 of the PCI Power Management Interface Specification.

PCI Power Management Control/Status Register (PMCSR)

Offset 48h

Bit	Name	Description
15	PME_STATUS	PME Status. This bit is set when the function would normally assert the PME signal independent of the state of the PME_EN bit. Writing a 1 to this bit will clear it and cause the function to stop asserting a PME (if enabled). Writing a 0 has no effect. If the function supports PME from D3cold then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded. Read/write accessible. Sticky bit. This bit is reset by POR. S_RESET or setting the STOP bit has no effect. If the function does not support PME# assertion from D3cold, either because bit 15 of the PMC Alias register is zero or the VAUX_SENSE pin is low, PME_STATUS will be reset following H_RESET. This reset is actually done at the end of the EEPROM read operation, since bit 15 of the PMC Alias register may be loaded from the EEPROM.
14-13	DATA_SCALE	Data Scale. This two bit read-only field indicates the scaling factor to be used when interpreting the value of the Data register. The value and meaning of this field will vary depending on the DATA_SCALE field. Read only.
12-9	DATA_SEL	Data Select. This optional four-bit field is used to select which data is reported through the Data register and DATA_SCALE field. Read/write accessible. Sticky bit. This bit is reset by POR. H_RESET, S_RESET, or setting the STOP bit has no effect.
8	PME_EN	PME Enable. When a 1, PME_EN enables the function to assert PME. When a 0, PME assertion is disabled. This bit defaults to “0” if the function does not support PME generation from D3cold. If the function supports PME from D3cold, then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded. Read/write accessible. Sticky bit. This bit is reset by POR. S_RESET or setting the STOP bit has no effect. If the function does not support PME assertion from D3cold, either because bit 15 of the PMC Alias register is zero or the VAUX_SENSE pin is low, PME_STATUS will be reset following H_RESET. This reset is actually done at the end of the EEPROM read operation, since bit 15 of the PMC Alias register may be loaded from the EEPROM.
7-2	RES	Reserved locations. Read only.
1-0	PWR_STATE	

Downloaded from Arrow.com.

Power State. This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state.

The definition of the field values is as follows:

00b - D0
01b - D1
10b - D2
11b - D3

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

Read/write accessible.

PCI PMCSR Bridge Support Extensions Register

Offset 4Ah

Bit	Name	Description
7-0	PMCSR_BSE	The PCI PMCSR Bridge Support Extensions Register is an 8-bit register. PMCSR Bridge Support Extensions are not supported. This is a read-only register whose content is 0.

PCI Data Register

Offset 4Bh

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

Bit	Name	Description
7-0	DATA_REG	The PCI Data Register is an 8-bit read-only register that is used as a window into an array of 8 ten-bit PCIDATA registers. The value read from this register is the 8 least significant bits of the PCIDATA register selected by DATA_SEL field of the PMCSR (offset 48h in PCI Configuration Space). The two most significant bits of the selected PCIDATA register are read from the DATA_SCALE field of the PMCSR. The interpretation of the contents of this register is described in the *PCI Bus Power Management Interface Specification*, Version 1.1.

Downloaded from Arrow.com.

Memory-Mapped Registers

The Memory-Mapped Registers give the host CPU access to all programmable features of the Am79C976 device. These registers are mapped directly into PCI memory space so that any programmable feature can be accessed with a single PCI memory read or write transaction. Data in these registers can be accessed as a single byte, a 16-bit word, or a 32-bit double word.

In addition, the Am79C976 controller's memory is prefetchable, which allows burst read and write operations. Accesses to consecutive registers or to registers logically wider than double word (BADX, BADR, PADR, etc.) may be treated as a single block move to take advantage of the PCI burst.

Registers that are logically wider than a double word are shown in the register descriptions as a single register. These may be accessed using multiple smaller accesses or with a single burst access.

Some registers that are smaller than a double word in width (STVAL, PADR[47:32], XMT_RING_LEN, RCV_RING_LEN) are placed in the memory map in the lower half of a double word. The upper half ignores writes and reads back zeros (STVAL, PADR[47:32]) or ones (XMT_RING_LEN, RCV_RING_LEN) as appropriate for sign extension. This allows these registers to be accessed with double word reads and writes.

Memory-mapped registers can be initialized with data loaded from the serial EEPROM.

The command and interrupt enable registers (CMD0, CMD2, CMD3, CMD7, and INTEN0) use a write access technique that in this document is called command style access. Command style access allows the host CPU to write to selected bits of a register without altering bits that are not selected. Command style registers are divided into 4 bytes that can be written independently. The high order bit of each byte is the 'value' bit that specifies the value that will be written to selected bits of the register. The 7 low order bits of each byte make up a bit map that selects which register bits will be altered. If a bit in the bit map is set to 1, the corresponding bit in the register will be loaded with the contents of the value bit. If a bit in the bit map is cleared to 0, the corresponding bit in the register will not be altered.

For example, if the value 10011010b is written to the least significant byte of a command style register, bits 1, 3, and 4 of the register will be set to 1, and the other bits will not be altered. If the value 00011010b is written to the same byte, bits 1, 3, and 4 will be cleared to 0, and the other bits will not be altered.

In the worst case it takes two write accesses to write to all of the bits in a command style register. One access writes to all bits that should be set to 1, and the other access writes to all bits that should be cleared to 0.

The EEPROM loading logic bypasses the command style access logic and treats command style registers just like the other writable memory-mapped registers. The value bits are ignored and the contents of the bit map fields are written directly into the corresponding registers. For example, if the EEPROM logic loads the value 00011010b into the least significant byte of a command style register, bits 1, 3, and 4 of the register will be set to 1, and bits 0, 2, 5, and 6 will be cleared to 0.

In the following register descriptions, the offset listed for each register is the offset relative to the contents of the PCI Memory-Mapped I/O Base Address Register.

MIB Offset

Offset 28h

This read-only register contains the offset of the block of statistics counters. For the Am79C976 device the content of this register is fixed at 200h. The contents of the MIB Offset Registers and therefore the locations of statistics counter blocks in other PCnet family devices may be different. Therefore, software should calculate the address of a particular MIB counter by adding the contents of the Memory-Mapped I/O Base Address Register plus the contents of this register plus the offset shown in Table 7 on page 76 or Table 8 on page 78.

AP_VALUE0: Auto-Poll Value0 Register

Offset 0A8h

The contents of this register are cleared to 0 when the $\overline{RST}$ pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 26. AP_VALUE0: Auto-Poll Value0 Register

Bit	Name	Description
15-0	AP_VALUE0	This register contains the results of the automatic polling of the user-selectable external PHY register, AP_REG0.

AP_VALUE1: Auto-Poll Value1 Register

Offset 0AAh

The contents of this register are cleared to 0 when the $\overline{RST}$ pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Downloaded from Arrow.com.

Table 27. AP_VALUE1: Auto-Poll Value1 Register

Bit	Name	Description
15-0	AP_VALUE 1	This register contains the results of the automatic polling of the user-selectable external PHY register, AP_REG1.

AP_VALUE2: Auto-Poll Value2 Register

Offset 0ACh

The contents of this register are cleared to 0 when the RST pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 28. AP_VALUE2: Auto-Poll Value2 Register

Bit	Name	Description
15-0	AP_VALUE2	This register contains the results of the automatic polling of the user-selectable external PHY register, AP_REG2.

AP_VALUE3: Auto-Poll Value3 Register

Offset 0AEh

The contents of this register are cleared to 0 when the RST pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 29. AP_VALUE3: Auto-Poll Value3 Register

Bit	Name	Description
15-0	AP_VALUE3	This register contains the results of the automatic polling of the user-selectable external PHY register, AP_REG3.

AP_VALUE4: Auto-Poll Value4 Register

Offset 0B0h

The contents of this register are cleared to 0 when the RST pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 30. AP_VALUE4: Auto-Poll Value4 Register

Bit	Name	Description
15-0	AP_VALUE4	This register contains the results of the automatic polling of the user-selectable external PHY register, AP_REG4.

AP_VALUE5: Auto-Poll Value5 Register

Offset 0B2h

The contents of this register are cleared to 0 when the RST pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 31. AP_VALUE5: Auto-Poll Value5 Register

Bit	Name	Description
15-0	AP_VALUE5	This register contains the results of the automatic polling of the user-selectable external PHY register, AP_REG5.

AUTOPOLL0: Auto-Poll0 Register

Offset 088h

This register controls the automatic polling of the status register of the default external PHY.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits except for bits 15 (AP_REG0_EN) and 12:8 (AP_REG0_ADDR) is 0. The default value for AP_REG0_EN is 1 and the default value for the AP_REG0_ADDR field is 00001b.

For legacy driver compatibility, the PHYAD field in BCR33, bits [9:5], is aliased from AUTOPOLL0, bits [9:5] when programmed through the EEPROM. The default values of AUTOPOLL0, bits [12:8] remain 00001 and are read-only. The contents of AUTOPOLL0, bits [7:5] remain reserved.

Downloaded from Arrow.com.

Table 32. AUTOPOLL0: Auto-Poll0 Register

Bit	Name	Description
15	AP_REG0_EN	Enable Bit for Autopoll Register 0. This bit is read-only and always has the value 1.
14-13	RES	Reserved locations. Written as zeros and read as undefined
12-8	AP_REG0_ ADDR	AP_REG0 Address. This field is read-only and always has the value 00001. See Note 1.
7-5	RES	Reserved locations. Written as zero and read as undefined. See Note 2.
4-0	AP_PHY0_ ADDR	Auto-Poll PHY0 Address. This field contains the address of the external PHY that contains AP-REG0. The Network Port Manager uses this PHY address for auto-negotiation. The Autopoll State Machine also uses this field as the default PHY address when one or more of the AP_PHYn_DFLT bits are set.

Notes:

1. *Bits [9:8] may be programmed via the EEPROM to alias the value into BCR33, bits [9:8].*
2. *Bits [7:5] may be programmed via the EEPROM to alias the value into BCR33, bits [7:5].*

AUTOPOLL1: Auto-Poll1 Register

Offset 08Ah

This register controls the automatic polling of a user-selectable external PHY register.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits in this register is 0.

Table 33. AUTOPOLL1: Auto-Poll1 Register

Bit	Name	Description
15	AP_REG1_EN	Enable Bit for Autopoll Register 1. When this bit and the Auto-Poll External PHY bit (APEP) in CMD3 are both set to 1, the Auto-Poll State Machine periodically reads the external PHY register selected by the AP_PHY1_ADDR and AP_REG1_ADDR fields and sets the APINT1 interrupt bit if it detects a change in the register's contents.
14-13	RES	Reserved locations. Written as zeros and read as undefined.
12-8	AP_REG1_ADDR	AP_REG1 Address. This field contains the register number of an external PHY register that the Auto-Poll State Machine will periodically read if the AP_REG1_EN bit in this register and the APEP bit (CMD3, bit 24) is set.
7	RES	Reserved location. Written as zero and read as undefined.
6	AP_PRE_SUP1	Auto-Poll Preamble Suppression. If this bit is set to 1, the Auto-Poll State Machine will suppress the preambles of the MII Management Frames that it uses to periodically read the external PHY register selected by the AP_PHY1_ADDR and AP_REG1_ADDR fields. This bit is ignored when the AP_PHY1_DFLT bit is set.
5	AP_PHY1_DFLT	Auto-Poll PHY1 Default. When this bit is set, the Auto-Poll State Machine ignores the contents of the AP_PHY1_ADDR and AP_PRE_SUP1 fields and uses the AP_PHY0_ADDR field for the address of the PHY device to be polled. If this bit is set, the Auto-Poll State Machine will suppress preambles only if the Port Manager has determined that the default external PHY can accept MII Management Frames without preambles. (The Port Manager examines bit 6 in register 1 of the default PHY to make this determination.)
4-0	AP_PHY1_ADDR	Auto-Poll PHY1 Address. This field contains the address of the external PHY that contains AP_REG1. This bit is ignored when the AP_PHY1_DFLT bit is set.

Downloaded from Arrow.com.

AUTOPOLL2: Auto-Poll2 Register

Offset 08Ch

This register controls the automatic polling of a user-selectable external PHY register, AP_REG2.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits in this register is 0.

Table 34. AUTOPOLL2: Auto-Poll2 Register

Bit	Name	Description
15	AP_REG2_EN	Enable Bit for Autopoll Register 2. When this bit and the Auto-Poll External PHY bit (APEP) in CMD3 are both set to 1, the Auto-Poll State Machine periodically reads the external PHY register selected by the AP_PHY2_ADDR and AP_REG2_ADDR fields and sets the APINT2 interrupt bit if it detects a change in the register's contents.
14-13	RES	Reserved locations. Written as zeros and read as undefined.
12-8	AP_REG2_ADDR	AP_REG2 Address. This field contains the register number of an external PHY register that the Auto-Poll State Machine will periodically read if the AP_REG2_EN bit in this register and the APEP bit (CMD3, bit 24) is set.
7	RES	Reserved location. Written as zero and read as undefined.
6	AP_PRE_SUP2	Auto-Poll Preamble Suppression. If this bit is set to 1, the Auto-Poll State Machine will suppress the preambles of the MII Management Frames that it uses to periodically read the external PHY register selected by the AP_PHY2_ADDR and AP_REG2_ADDR fields. This bit is ignored when the AP_PHY2_DFLT bit is set.
5	AP_PHY2_DFLT	Auto-Poll PHY2 Default. When this bit is set, the Auto-Poll State Machine ignores the contents of the AP_PHY2_ADDR and AP_PRE_SUP2 fields and uses the AP_PHY0_ADDR field for the address of the PHY device to be polled. If this bit is set, the Auto-Poll State Machine will suppress preambles only if the Port Manager has determined that the default external PHY can accept MII Management Frames without preambles. (The Port Manager examines bit 6 in register 1 of the default PHY to make this determination.)
4-0	AP_PHY2_ADDR	Auto-Poll PHY2 Address. This field contains the address of the external PHY that contains AP_REG2. This bit is ignored when the AP_PHY2_DFLT bit is set.

Downloaded from Arrow.com.

AUTOPOLL3: Auto-Poll3 Register

Offset 08Eh

This register controls the automatic polling of a user-selectable external PHY register, AP_REG3.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits in this register is 0.

Table 35. AUTOPOLL3: Auto-Poll3 Register

Bit	Name	Description
15	AP_REG3_EN	Enable Bit for Autopoll Register 3. When this bit and the Auto-Poll External PHY bit (APEP) in CMD3 are both set to 1, the Auto-Poll State Machine periodically reads the external PHY register selected by the AP_PHY3_ADDR and AP_REG3_ADDR fields and sets the APINT3 interrupt bit if it detects a change in the register's contents.
14-13	RES	Reserved locations. Written as zeros and read as undefined.
12-8	AP_REG3_ADDR	AP_REG3 Address. This field contains the register number of an external PHY register that the Auto-Poll State Machine will periodically read if the AP_REG3_EN bit in this register and the APEP bit (CMD3, bit 24) is set.
7	RES	Reserved location. Written as zero and read as undefined.
6	AP_PRE_SUP3	Auto-Poll Preamble Suppression. If this bit is set to 1, the Auto-Poll State Machine will suppress the preambles of the MII Management Frames that it uses to periodically read the external PHY register selected by the AP_PHY3_ADDR and AP_REG3_ADDR fields. This bit is ignored when the AP_PHY3_DFLT bit is set.
5	AP_PHY3_DFLT	Auto-Poll PHY3 Default. When this bit is set, the Auto-Poll State Machine ignores the contents of the AP_PHY3_ADDR and AP_PRE_SUP3 fields and uses the AP_PHY0_ADDR field for the address of the PHY device to be polled. If this bit is set, the Auto-Poll State Machine will suppress preambles only if the Port Manager has determined that the default external PHY can accept MII Management Frames without preambles. (The Port Manager examines bit 6 in register 1 of the default PHY to make this determination.)
4-0	AP_PHY3_ADDR	Auto-Poll PHY3 Address. This field contains the address of the external PHY that contains AP_REG3. This bit is ignored when the AP_PHY3_DFLT bit is set.

Downloaded from Arrow.com.

AUTOPOLL4: Auto-Poll4 Register

Offset 090h

This register controls the automatic polling of a user-selectable external PHY register, AP_REG4.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits in this register is 0.

Table 36. AUTOPOLL4: Auto-Poll4 Register

Bit	Name	Description
15	AP_REG4_EN	Enable Bit for Autopoll Register 4. When this bit and the Auto-Poll External PHY bit (APEP) in CMD3 are both set to 1, the Auto-Poll State Machine periodically reads the external PHY register selected by the AP_PHY4_ADDR and AP_REG4_ADDR fields and sets the APINT4 interrupt bit if it detects a change in the register's contents.
14-13	RES	Reserved locations. Written as zeros and read as undefined.
12-8	AP_REG4_ADDR	AP_REG4 Address. This field contains the register number of an external PHY register that the Auto-Poll State Machine will periodically read if the AP_REG4_EN bit in this register and the APEP bit (CMD3, bit 24) is set.
7	RES	Reserved location. Written as zero and read as undefined.
6	AP_PRE_SUP4	Auto-Poll Preamble Suppression. If this bit is set to 1, the Auto-Poll State Machine will suppress the preambles of the MII Management Frames that it uses to periodically read the external PHY register selected by the AP_PHY4_ADDR and AP_REG4_ADDR fields. This bit is ignored when the AP_PHY4_DFLT bit is set.
5	AP_PHY4_DFLT	Auto-Poll PHY4 Default. When this bit is set, the Auto-Poll State Machine ignores the contents of the AP_PHY4_ADDR and AP_PRE_SUP4 fields and uses the AP_PHY0_ADDR field for the address of the PHY device to be polled. If this bit is set, the Auto-Poll State Machine will suppress preambles only if the Port Manager has determined that the default external PHY can accept MII Management Frames without preambles. (The Port Manager examines bit 6 in register 1 of the default PHY to make this determination.)
4-0	AP_PHY4_ADDR	Auto-Poll PHY4 Address. This field contains the address of the external PHY that contains AP_REG4. This bit is ignored when the AP_PHY4_DFLT bit is set.

Downloaded from Arrow.com.

AUTOPOLL5: Auto-Poll5 Register

Offset 092h

This register controls the automatic polling of a user-selectable external PHY register, AP_REG5.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits in this register is 0.

Table 37. AUTOPOLL5: Auto-Poll5 Register

Bit	Name	Description
15	AP_REG5_EN	Enable Bit for Autopoll Register 5. When this bit and the Auto-Poll External PHY bit (APEP) in CMD3 are both set to 1, the Auto-Poll State Machine periodically reads the external PHY register selected by the AP_PHY5_ADDR and AP_REG5_ADDR fields and sets the APINT5 interrupt bit if it detects a change in the register's contents.
14-13	RES	Reserved locations. Written as zeros and read as undefined.
12-8	AP_REG5_ADDR	AP_REG5 Address. This field contains the register number of an external PHY register that the Auto-Poll State Machine will periodically read if the AP_REG5_EN bit in this register and the APEP bit (CMD3, bit 24) is set.
7	RES	Reserved location. Written as zero and read as undefined.
6	AP_PRE_SUP5	Auto-Poll Preamble Suppression. If this bit is set to 1, the Auto-Poll State Machine will suppress the preambles of the MII Management Frames that it uses to periodically read the external PHY register selected by the AP_PHY5_ADDR and AP_REG5_ADDR fields. This bit is ignored when the AP_PHY5_DFLT bit is set.
5	AP_PHY5_DFLT	Auto-Poll PHY5 Default. When this bit is set, the Auto-Poll State Machine ignores the contents of the AP_PHY5_ADDR and AP_PRE_SUP5 fields and uses the AP_PHY0_ADDR field for the address of the PHY device to be polled. If this bit is set, the Auto-Poll State Machine will suppress preambles only if the Port Manager has determined that the default external PHY can accept MII Management Frames without preambles. (The Port Manager examines bit 6 in register 1 of the default PHY to make this determination.)
4-0	AP_PHY5_ADDR	Auto-Poll PHY5 Address. This field contains the address of the external PHY that contains AP_REG5. This bit is ignored when the AP_PHY5_DFLT bit is set.

BADR: Receive Ring Base Address Register

Offset 120h

This 64-bit register allows the Receive Descriptor Ring to be located anywhere in a 64-bit address space. For systems with a 32-bit or smaller address space, it is only necessary to program the lower 32 bits of this register (by writing to offset 120h). The upper 32 bits will remain at the default value of 0.

The contents of this register are set to the default value 0 when the $\overline{RST}$ pin is asserted. This register is not affected by the serial EEPROM read operation or by a serial EEPROM read error.

Table 38. Receive Ring Base Address Register

Bit	Name	Description
63-0	BADR	Base Address of Receive Descriptor Ring. In systems with a 32-bit or smaller address space, it is only necessary to program the 32 low-order bits of this register. The low order 32 bits of this register are an alias of CSR24 and CSR25.

Downloaded from Arrow.com.

BADX: Transmit Ring Base Address Register

Offset 100h

This 64-bit register allows the Transmit Descriptor Ring to be located anywhere in a 64-bit address space. For systems with a 32-bit or smaller address space, it is only necessary to program the lower 32 bits of this register (by writing to offset 100h). The upper 32 bits will remain at the default value of 0.

The contents of this register are set to the default value 0 when the RST pin is asserted. This register is not affected by the serial EEPROM read operation or by a serial EEPROM read error.

Table 39. Transmit Ring Base Address Register

Bit	Name	Description
63-0	BADX	Base Address of Transmit Descriptor Ring. In systems with a 32-bit or smaller address space, it is only necessary to program the 32 low-order bits of this register. The low-order 32 bits of this register are an alias of CSR30 and CSR31.

CHIPID: Chip ID Register

Offset 0F0h

This read-only register is an alias of CSR88.

Table 40. CHIPID: Chip ID Register

Bit	Name	Description
31-28	VER	Version. This 4-bit pattern is silicon-revision dependent.
27-12	PARTID	Part number. The 16-bit code for the Am79C976 controller is 0010 0110 0010 1000b (2628h). This register is exactly the same as the Device ID register in the JTAG description. However, this part number is different from that stored in the Device ID register in the PCI configuration space.
11-1	MANFID	Manufacturer ID. The 11-bit manufacturer code for AMD is 00000000001b. This code is per the JEDEC Publication 106-A. Note that this code is not the same as the Vendor ID in the PCI configuration space.
0	ONE	Always logic 1.

CHPOLLTIME: Chain Poll Timer Register

Offset 18Ah

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 41. CHPOLLTIME: Chain Polling Interval Register

Bit	Name	Description
15-0	CHPOLL TIME	Chain Polling Interval. This register contains the time that the Am79C976 controller will wait between successive polling operations when the Buffer Management Unit is in the middle of a buffer chaining operation. The CHPOLLTIME value is expressed as the two's complement of the desired interval, where each bit of CHPOLLTIME approximately represents 3 ERCLK periods. CHPOLLTIME[3:0] are ignored. (CHPOLLTIME[16] is implied to be a 1, so CHPOLLTIME[15] is significant and does not represent the sign of the two's complement CHPOLLTIME value.) The default value of this register is 0000h. This corresponds to a polling interval of 65,536 clock periods (2.185 ms when ERCLK = 90 MHz). Setting the INIT bit starts an initialization process that sets CHPOLLTIME to its default value. If the user wants to program a value for CHPOLLTIME other than the default, then the user must change the value after the initialization sequence has completed. This register is an alias for CSR49.

Downloaded from Arrow.com.

CMD0: Command0

Offset 048h

CMD0 is a command-style register. All bits in this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 42. CMD0: Command0 Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	VAL1	Value bit for byte 1. The value of this bit is written to any bits in the CMD0 register that correspond to bits in the CMD0[14:8] bit map field that are set to 1.
14-13	RES	Reserved locations. Written as zeros and read as undefined.
12	RDMD	Receive Demand, when set, causes the Descriptor Management Unit to access the Receive Descriptor Ring without waiting for the chain poll-time counter to expire.
11-9	RES	Reserved locations. Written as zeros and read as undefined.
8	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. If the TXDPOLL bit in CMD2 is set, the host processor must set TDMD each time it is ready for the Am79C976 device to poll the Transmit Descriptor Ring. When TXDPOLL = 0, setting TDMD merely hastens the Am79C976 controller's next access to a Transmit Descriptor Ring Entry.
7	VAL0	Value bit for byte 0. The value of this bit is written to any bits in the CMD0 register that correspond to bits in the CMD0[6:0] bit map field that are set to 1.
6	UINTCMD	User Interrupt Command. UINTCMD can be used by the host to generate an interrupt unrelated to any network activity. Writing a 1 to this bit causes the UINT bit in the Interrupt Register to be set to 1, which in turn causes INTA to be asserted if interrupts are enabled. UINTCMD is always read as 0.
5	RX_FAST_ SPND	Receive Fast Suspend. Setting this bit causes the receiver to suspend its activities as quickly as possible without stopping in the middle of a frame reception. Setting RX_FAST_SPND does not stop the DMA controller from transferring frame data from the receive FIFO to host memory. If a frame is being received at the time that RX_FAST_SPND is set to 1, the reception of that frame will be completed, but no more frames will be received until RX_FAST_SPND is cleared to 0. After the receiver has suspended its activity, the RX_SUSPENDED bit in the Status register and the Suspend Interrupt (SPNDINT) bit in the Interrupt register will be set, which will cause an interrupt to occur if interrupts are enabled and the SPNDINTEN bit in the Interrupt Enable register is set.
4	TX_FAST_SPND	Transmit Fast Suspend. Setting this bit causes the transmitter to suspend its activities as quickly as possible without stopping in the middle of a frame transmission. Setting TX_FAST_SPND does not stop the DMA controller from transferring frame data from host memory to the transmit FIFO. If a frame is being transmitted at the time that TX_FAST_SPND is set to 1, the transmission of that frame will be completed, but no more frames will be transmitted until TX_FAST_SPND is cleared to 0. After the transmitter has suspended its activity, the TX_SUSPENDED bit in the Status register and the Suspend Interrupt (SPNDINT) bit in the INT0 register will be set, which will cause an interrupt to occur if interrupts are enabled and the SPNDINTEN bit in the Interrupt Enable register is set.

Downloaded from Arrow.com.

Bit	Name	Description
3	RX_SPND	Receive Suspend. Setting this bit causes the receiver to suspend its activities without stopping in the middle of a frame reception. After the receiver suspends its activities, the DMA controller continues copying data from the Receive FIFO into host system memory until the FIFO is empty or until no more receive descriptors are available. After the Receive FIFO has been emptied or after all receive descriptors have been used, the receive DMA controller suspends its polling activity, and the RX_SUSPENDED bit in the Status register and the Suspend Interrupt (SPNDINT) bit in the Interrupt register will be set. Setting the SPNDINT bit will cause an interrupt to occur if interrupts are enabled and the SPNDINTEN bit in the Interrupt Enable register is set.
2	TX_SPND	Transmit Suspend. Setting this bit causes the transmitter to suspend its activities without stopping in the middle of a frame transmission. After the transmitter suspends its activities, the DMA controller continues copying data from host system memory into the Transmit FIFO until the FIFO is full or until no more transmit descriptors are available. After the Transmit FIFO has been filled or after all transmit descriptors have been used, the transmit DMA controller suspends its polling activity, and the TX_SUSPENDED bit in the Status register and the Suspend Interrupt (SPNDINT) bit in the Interrupt register will be set. Setting the SPNDINT bit will cause an interrupt to occur if interrupts are enabled and the SPNDINTEN bit in the Interrupt Enable register is set.
1	INTREN	Interrupt Enable. This bit allows $\overline{\text{INTA}}$ to be asserted if any bit in the Interrupt register is set. If INTREN is cleared to 0, $\overline{\text{INTA}}$ will not be asserted, regardless of the state of the Interrupt register. INTREN is not an alias of the IENA bit in CSR0.
0	RUN	Setting the RUN bit enables the Am79C976 controller to start processing descriptors and transmitting and receiving packets. Clearing the RUN bit to 0 abruptly disables the transmitter, receiver, and descriptor processing logic, possibly while a frame is being transmitted or received. The act of changing the RUN bit from 1 to 0 causes the following bits to be reset to 0: IENA, TX_SPND, RX_SPND, TDMD, RDMD, UINTCMD, TX_FAST_SPND, RINT, TINT, TXSTRTINT, TXDNINT, MPINT, and UINT.

CMD2: Command2

Offset 050h

CMD2 is a command-style register. All bits in this register are cleared to 0 when the $\overline{\text{RST}}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 43. CMD2: Command2 Register

Bit	Name	Description
31	VAL3	Value bit for byte 3. The value of this bit is written to any bits in the CMD2 register that correspond to bits in the CMD2[30:24] bit map field that are set to 1.
30	NOUFLO	No Underflow on Transmit. When the NOUFLO bit is set to 1, the Am79C976 controller will not start transmitting the preamble for a packet until the Transmit Start Point (CTRL1, bits 16-17) requirement has been met *and* the complete packet has been copied into the transmit FIFO. When the NOUFLO bit is cleared to 0, the Transmit Start Point is the only restriction on when preamble transmission begins for transmit packets. Setting the NOUFLO bit guarantees that the Am79C976 controller will never suffer transmit underflows, because the arbiter that controls transfers to and from the SSRAM guarantees a worst case latency on transfers to and from the MAC and Bus Transmit FIFOs such that it will never underflow if the complete packet has been copied into the Am79C976 controller before packet transmission begins. This bit is an alias of BCR18, bit 11. It is included only to allow programming from the EEPROM for compatibility with legacy software.

Downloaded from Arrow.com.

Bit	Name	Description
29	LEDPE	LED Program Enable. When LEDPE is set to 1, programming of the LED functions through BCR4, BCR5, BCR6, and BCR7 is enabled. When LEDPE is cleared to 0, programming of LED functions through these registers is disabled. Writes to these registers will be ignored. However, LEDPE does not affect the programming of LED functions through the memory mapped LED registers, LED0, LED1, LED2 and LED3. The memory-mapped LED registers are always enabled, regardless of the state of LEDPE. This bit is an alias of BCR2, bit 12. It is included only to allow programming from the EEPROM for compatibility with legacy software.
28	DWIO	Double Word I/O. When set, this bit indicates that the Am79C976 controller is programmed for DWord I/O (DWIO) mode. When cleared, this bit indicates that the Am79C976 controller is programmed for Word I/O (WIO) mode. This bit affects the I/O Resource Offset map and it affects the defined width of the Am79C976 controllers I/O resources. See the DWIO and WIO sections for more details. The initial value of the DWIO bit is determined by the programming of the EEPROM. The value of DWIO can be altered automatically by the Am79C976 controller. Specifically, the Am79C976 controller will set DWIO if it detects a DWord write access to offset 10h from the Am79C976 controller I/O base address (corresponding to the RDP resource). This bit is an alias of BCR18, bit 7. It is included only to allow programming from the EEPROM for compatibility with legacy software.
27	APROMWE	Address PROM Write Enable. The Am79C976 controller contains a 16-byte shadow RAM on board that emulates a small PROM that was used in conjunction with early AMD Ethernet controllers. Accesses to Address PROM I/O space will be directed to this RAM. When APROMWE is set to 1, then write access to the shadow RAM will be enabled. This bit is an alias of BCR2, bit 8. It is included only to allow programming from the EEPROM for compatibility with legacy software.
26-24	RES	Reserved locations. Written as zeros and read as undefined.
23	VAL2	Value bit for byte 2. The value of this bit is written to any bits in the CMD2 register that correspond to bits in the CMD2[22:16] bit map field that are set to 1.
22-21	RES	Reserved locations. Written as zeros and read as undefined.
20	FDRPA	Full-Duplex Runt Packet Accept. When FDRPA is cleared to 0 and full-duplex mode is enabled, the Am79C976 controller will only receive frames that meet the minimum Ethernet frame length of 64 bytes. Receive DMA will not start until at least 64 bytes or a complete frame have been received. By default, FDRPA is cleared to 0. When FDRPA is set to 1, the Am79C976 controller will accept any frame of 12 bytes or greater, and receive DMA will start according to the programming of the receive FIFO watermark. This bit is the inverse of BCR9, bit 2.
19	RPA	Runt Packet Accept. This bit forces the Am79C976 controller to accept runt packets (packets shorter than 64 bytes). The minimum packet size that can be received is 12 bytes. This bit is an alias of CSR124, bit 3.
18	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the Am79C976 controller will be disabled. Frames addressed to the node's individual physical address will not be recognized. This bit is an alias of CSR15, bit 13.
17	DRCVBC	Disable Receive Broadcast. When set, disables the Am79C976 controller from receiving broadcast messages. Used for protocols that do not support broadcast addressing, except as a function of multicast. DRCVBC is cleared by activation of H_RESET or S_RESET (broadcast messages will be received) and is unaffected by STOP. This bit is an alias of CSR15, bit 14.
16	PROM	Promiscuous Mode. When PROM = 1, all incoming receive frames are accepted, regardless of their destination addresses. This bit is an alias of CSR15, bit 15.

Downloaded from Arrow.com.

Bit	Name	Description
15	VAL1	Value bit for byte 1. The value of this bit is written to any bits in the CMD2 register that correspond to bits in the CMD2[14:8] bit map field that are set to 1.
14	RES	Reserved location. Written as zero and read as undefined.
13	ASTRP_RCV	Auto Strip Receive. When set, ASTRP_RCV enables the automatic pad stripping feature. For any receive frame whose length field has a value less than 46, the pad and FCS fields will be stripped and not placed in the FIFO. This bit is an alias of CSR4, bit 10.
12	FCOLL	Force Collision. This bit allows the collision logic to be tested. The Am79C976 controller must be in internal loopback for FCOLL to be valid. If FCOLL = 1, a collision will be forced during loopback transmission attempts, which will result in a Retry Error. If FCOLL = 0, the Force Collision logic will be disabled. This bit is an alias of CSR15, bit 4.
11	EMBA	Enable Modified Back-off Algorithm (see Contention Resolution section in *Media Access Management* section for more details). If EMBA is set, a modified back-off algorithm is implemented. This bit is an alias of CSR3, bit 3.
10	DXMT2PD	Disable Transmit Two Part Deferral (see Medium Allocation section in the *Media Access Management* section for more details). If DXMT2PD is set, Transmit Two Part Deferral will be disabled. This bit is an alias of CSR3, bit 4.
9	LTINTEN	Last Transmit Interrupt Enable. When this bit is set to 1, the LTINT bit in transmit descriptors can be used to determine when transmit interrupts occur. The Transmit Interrupt (TINT) bit will be set after a frame has been copied to the Transmit FIFO if the LTINT bit in the frame's last transmit descriptor is set. If the LTINT bit in the frame's last descriptor is 0 TINT will not be set after the frame has been copied to the Transmit FIFO. This bit is an alias of CSR5, bit 14.
8	DXMTFCS	Disable Transmit CRC (FCS). When DXMTFCS is set to 0, the transmitter will generate and append an FCS to the transmitted frame. When DXMTFCS is set to 1, no FCS is generated or sent with the transmitted frame. DXMTFCS is overridden when ADD_FCS and ENP bits are set in the transmit descriptor. When the auto padding logic, which is enabled by the APAD_XMT bit (CMD2, bit6), adds padding to a frame, a valid FCS field is appended to the frame, regardless of the state of DXMTFCS. If DXMTFCS is set and ADD_FCS is clear for a particular frame, no FCS will be generated. If ADD_FCS is set for a particular frame, the state of DXMTFCS is ignored and a FCS will be appended on that frame by the transmit circuitry. See also the ADD_FCS bit in the transmit descriptor. This bit was called DTCR in the LANCE (Am7990) device. This bit is an alias of CSR15, bit 3.
7	VAL0	Value bit for byte 0. The value of this bit is written to any bits in the CMD2 register that correspond to bits in the CMD2[6:0] bit map field that are set to 1.
6	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. When the auto padding logic modifies a frame, a valid FCS field will be appended to the frame, regardless of the state of the DXMTFCS bit (CMD2, bit 8) and of the ADD_FCS bit in the transmit descriptor. This bit is an alias of CSR4, bit 11.
5	DRTY	Disable Retry. When DRTY is set to 1, the Am79C976 controller will attempt only one transmission. In this mode, the device will not protect the first 64 bytes of frame data in the Transmit FIFO from being overwritten, because automatic retransmission will not be necessary. When DRTY is set to 0, the Am79C976 controller will attempt 16 transmissions before signaling a retry error. This bit is an alias of CSR15, bit 5.

Downloaded from Arrow.com.

Bit	Name	Description
4	INLOOP	Internal Loopback. When this bit is set, the transmitter is internally connected to the receiver so that the TXD[3:0] outputs are connected internally to the RXD[3:0] inputs, the TX_EN output is connected to the RX_DV input, and RX_CLK is connected to TX_CLK. The device is forced into full duplex mode so that collisions can not occur. The INLOOP and EXLOOP bits should not be set at the same time. Setting INLOOP to 1 is equivalent to setting LOOP (in CSR15) to 1 and MIIILP (in BCR32) to 1.
3	EXLOOP	External Loopback. When this bit is set, the device is forced into full duplex mode so that collisions can not occur during loop back testing. If the TXD[3:0] outputs are connected externally to the RXD[3:0] inputs, the TX_EN output is externally connected to the RX_DV input, and RX_CLK is connected to TX_CLK, then transmitted frames will also be received. This connection can be made by attaching an external jumper or by programming an attached PHY to loopback mode. The INLOOP and EXLOOP bits should not be set at the same time. Setting EXLOOP to 1 is equivalent to setting LOOP (in CSR15) to 1 and MIIILP (in BCR32) to 0.
2	LAPPEN	Look Ahead Packet Processing Enable. When set to a 1, the LAPPEN bit will cause the Am79C976 controller to generate an interrupt following the descriptor write operation to the first buffer of a receive frame. This interrupt will be generated in addition to the interrupt that is generated following the descriptor write operation to the last buffer of a receive packet. The interrupt will be signaled through the RINT bit of CSR0 or the INT register. Setting LAPPEN to a 1 also modifies the way the controller accesses the Receive Descriptors. See the *Look Ahead Packet Processing* section. This bit is an alias of CSR3, bit 5. See Appendix A for more information on the Look Ahead Packet Processing concept.
1	CHDPOLL	Disable Chain Polling. If CHDPOLL is set, the Buffer Management Unit will disable chain polling. Likewise, if CHDPOLL is cleared, automatic chain polling is enabled. If CHDPOLL is set and the Buffer Management Unit is in the middle of a buffer-changing operation, setting the RDMD bit in CMD0 or CSR7 will cause a poll of the current receive descriptor, and setting the TDMD bit in CMD0 or CRR0 will cause a poll of the current transmit descriptor. If CHDPOLL is set, the RDMD bit in CSR7 or CMD0 can be set to initiate a manual poll of a receive or transmit descriptor if the Buffer Management Unit is in the middle of a buffer-chaining operation. This bit is an alias of CSR7, bit 12.
0	TXDPOLL	Disable Transmit Polling. If TXDPOLL is set, the Buffer Management Unit will disable transmit polling. Likewise, if TXDPOLL is cleared, automatic transmit polling is enabled. If TXDPOLL is set, TDMD bit in CSR0 or CMD0 must be set in order to initiate a manual poll of a transmit descriptor. Transmit descriptor polling will not take place if TXON is reset. Transmit polling will take place following Receive activities. This bit is an alias of CSR4, bit 12.

Downloaded from Arrow.com.

CMD3: Command3

Offset 054h

CMD3 is a command-style register. All bits in this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 44. CMD3: Command3 Register

Bit	Name	Description
31	VAL3	Value bit for byte 3. The value of this bit is written to any bits in the CMD3 register that correspond to bits in the CMD3[30:24] bit map field that are set to 1.
30	PREFETCH_DIS	Disable Prefetchability. This bit, which can be loaded from EEPROM, is the inverse of the value reported in the PREFETCH bit in the PCI Memory-Mapped I/O Base Address Register, which is read-only. Setting PREFETCH_DIS to 1 indicates that the memory space claimed by this device can not be prefetched. Because of the side effects of reading the Reset Register at offset 14h or 18h (depending on the state of DWIO (CMD2, bit 28)), locations at offsets less than 20h cannot be prefetched. The Am79C976 device will disconnect any attempted burst transfer at offsets less than 20h. If a logical 1 is written to this bit position, the corresponding bit in the register will be loaded with the contents of the VAL3 bit. If a logical 0 is written to this bit position, the corresponding bit in the register will not be altered.
29	DIS_READ_WAIT	Disable Read Wait. When this bit is set to 1, the controller will not insert $\overline{\text{IRDY}}$ wait states in burst read transfers.
28	DIS_WRITE_WAIT	Disable Write Wait. When this bit is set to 1, the controller will not insert $\overline{\text{IRDY}}$ wait states in burst write transfers.
27	DISABLE_MWI	Disable MWI. When this bit is set to 1, the controller will not generate MWI PCI bus commands.
26	RST_PHY	Reset PHY. When this bit is set to 1, the controller will assert the PHY_RST signal. The signal will remain asserted for as long as this bit remains set.
25	INIT_MIB	Initialize Management Information Base Counters. Setting this bit will cause all of the MIB counters to be reset to 0. Resetting these counters takes about 55 ERCLK cycles. This bit is cleared automatically after the counters have all been reset to 0. This bit must not be set to 1 by the EEPROM logic. If a logical 1 is written to this bit position, the corresponding bit in the register will be loaded with the contents of the VAL3 bit. If a logical 0 is written to this bit position, the corresponding bit in the register will not be altered.
24	APEP	MII Auto-Poll External PHY (APEP) When set to 1, the Am79C976 controller will poll the MII status register in the external PHY. This feature allows the software driver or upper layers to see any changes in the status of the external PHY. An interrupt, when enabled, is generated when the contents of the new status is different from the previous status. This bit is an alias of BCR32, bit 11.
23	VAL2	Value bit for byte 2. The value of this bit is written to any bits in the CMD3 register that correspond to bits in the CMD3[22:16] bit map field that are set to 1.
22	RES	Reserved. Written as 1 and read as undefined.
21	JUMBO	Accept Jumbo Frames. This bit affects the way the MIB counters count long frames. If JUMBO is 0, only frames that are between 64 and 1518 bytes (or 1522 bytes if VLAN is set to 1) are counted as valid frames. When JUMBO is 1, any frame longer than 63 bytes with a valid FCS field is counted as a valid frame.
20	VSIZE	VLAN Frame Size. This bit determines the maximum frame size used for determining when to increment the XmtPkts1024to1518Octets, XmtExcessiveDefer, RcvPkts1024to1518Octets, and RcvOversizePkts MIB counters and when to assert the Excessive Deferral Interrupt. When this bit is set to 1 the maximum frame size is 1522 bytes. When it is cleared to 0, the maximum frame size is 1518 bytes.

Downloaded from Arrow.com.

Bit	Name	Description
19	VLONLY	Admit Only VLAN Frames. When this bit is set to 1, only frames with a VLAN Tag Header containing a non-zero VLAN ID field will be received. All other frames will be rejected.
18	REX_RTRY	Retransmit on Retry Error. When this bit is set to 1, if collisions occur for 16 attempts to transmit a frame, the back-off logic is reset, but the frame is not discarded. Instead, it is treated as if it were the next frame in the transmit queue. When this bit is cleared to 0, if collisions occur for 16 attempts to transmit a frame, the frame is dropped. In either case, if collisions occur for 16 attempts to transmit a frame, the XmtExcessiveCollision counter is incremented.
17	REX_UFLO	Retransmit on Underflow. When this bit is set to 1, if the transmitter is forced to abort a transmission because the transmit FIFO underflows, the transmitter will not discard the frame. Instead, it will automatically wait until the entire frame has been loaded into the transmit FIFO and then will restart the transmission process. When this bit is cleared to 0, if the transmitter is forced to abort a transmission because the FIFO underflows, the transmitter will discard the frame. In either case, the XmtUnderrunPkts counter will be incremented.
16	RTRY_LCOL	Retry on Late Collision. When this bit is set to 1, late collisions are treated like normal collisions, except that the XmtLateCollision counter and the XmtCollisions counter will both be incremented. When this bit is cleared to 0, if a late collision occurs, the transmitter will discard the frame that was being transmitted when the collision occurred.
15	VAL1	Value bit for byte 1. The value of this bit is written to any bits in the CMD3 register that correspond to bits in the CMD3[14:8] bit map field that are set to 1.
14	DISPM	Disable Port Manager. (The corresponding bit in older PCnet family devices is called Disable Auto-Negotiation Auto Setup or DANAS. The name has been changed, but not the function.) When DISPM is set, the Network Port Manager function is disabled, and the host CPU is responsible for ensuring that the MAC and external PHY are operating in the same mode. This bit is an alias of BCR32, bit 7.
13	INTLEVEL	Interrupt Level. This bit allows the interrupt output signals to be programmed for level- or edge-sensitive applications. When INTLEVEL is cleared to 0, the $\overline{\text{INTA}}$ pin is configured for level-sensitive applications. In this mode, an interrupt request is signaled by a low level driven on the $\overline{\text{INTA}}$ pin by the Am79C976 controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is tri-stated by the Am79C976 controller and allowed to be pulled to a high level by an external pull-up device. This mode is intended for systems which allow the interrupt signal to be shared by multiple devices. When INTLEVEL is set to 1, the $\overline{\text{INTA}}$ pin is configured for edge-sensitive applications. In this mode, an interrupt request is signaled by a high level driven on the $\overline{\text{INTA}}$ pin by the Am79C976 controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is driven to a low level by the Am79C976 controller. This mode is intended for systems that do not allow interrupt channels to be shared by multiple devices. INTLEVEL should not be set to 1 when the Am79C976 controller is used in a PCI bus application. This bit is an alias of BCR2, bit 7.
12	FORCE_FD	Force Full Duplex. (This bit is called Full-Duplex Enable (FDEN) in other PCnet family devices.) This bit controls whether full-duplex operation is enabled. When FORCE_FD is cleared and the Port Manager is disabled, the Am79C976 controller will always operate in the half-duplex mode. When FORCE_FD is set, the Am79C976 controller will operate in full-duplex mode. ***Do not set this bit when the Port Manager is enabled***. This bit is an alias of BCR9, bit 0.
11	FORCE_LS	Force Link Status. When this bit is set, the internal link status is forced to the Pass state regardless of the actual state of the PHY device. When this bit is cleared to 0, the internal link status is determined by the Port Manager.

Downloaded from Arrow.com.

Bit	Name	Description
10	RXFRTGEN	Receive Frame Tag Enable. When this bit is set, frame tag data that is shifted in through the External Address Detection Interface (EADI) while a frame is being received will be copied to the receive descriptor.
9	MPPLBA	Magic Packet Physical Logical Broadcast Accept. If MPPLBA is at its default value of 0, the Am79C976 controller will only detect a Magic Packet frame if the destination address of the packet matches the content of the physical address register (PADR). If MPPLBA is set to 1, the destination address of the Magic Packet frame can be unicast, multicast, or broadcast. Note that the setting of MPPLBA only affects the address detection of the Magic Packet frame. The Magic Packet frame's data sequence must be made up of 16 consecutive physical addresses (PADR[47:0]) regardless of what kind of destination address it has. This bit is OR'ed with EMPPLBA bit (CSR116, bit 6). This bit is an alias of CSR5, bit 5.
8	MPPEN_EE	Magic Packet Pin Enable. When either this bit or the MPPEN_SW bit in CMD7 is set, the device enters the Magic Packet mode when the PG input goes LOW. This bit has the same function as MPPEN_SW except that H_RESET clears MPPEN_EE to 0, while H_RESET has no effect on MPPEN_SW. This bit is an alias of CSR116, bit 4.
7	VAL0	Value bit for byte 0. The value of this bit is written to any bits in the CMD3 register that correspond to bits in the CMD3[6:0] bit map field that are set to 1.
6	MPEN_EE	Magic Packet Enable. When either this bit or the MPEN_SW bit in CMD7 is set, the device enters the Magic Packet mode. This bit has the same function as MPEN_SW except that H_RESET clears MPEN_EE to 0, while H_RESET has no effect on MPEN_SW.
5	LCMODE_EE	Link Change Wake-up Mode. When either this bit or the LCMODE_SW bit in CMD7 is set to 1, the LCDET bit gets set when the MII auto polling logic detects a Link Change. This bit has the same function as LCMODE_SW except that H_RESET clears LCMODE_EE to 0, while H_RESET has no effect on LCMODE_SW. This bit is an alias of CSR116, bit 8.
4	PME_EN_OVR	PME_EN Overwrite. When this bit is set and the MPMAT or LCDET bit is set, the $\overline{\text{PME}}$ pin will always be asserted regardless of the state of PME_EN bit. This bit is an alias of CSR116, bit 10.
3	RWU_DRIVER	RWU Driver Type. If this bit is set to 1, RWU is a totem pole driver; otherwise, RWU is an open drain output. This bit is an alias of CSR116, bit 3.
2	RWU_GATE	RWU Gate Control. If this bit is set, RWU is forced to the high Impedance State when PG is LOW, regardless of the state of the MPMAT and LCDET bits. This bit is an alias of CSR116, bit 2.
1	RWU_POL	RWU Pin Polarity. If RWU_POL is set to 1, the RWU pin is normally HIGH and asserts LOW; otherwise, RWU is normally LOW and asserts HIGH. This bit is an alias of CSR116, bit 1.
0	RST_POL	PHY_RST Pin Polarity. If the RST_POL is set to 1, the PHY_RST pin is active LOW; otherwise, PHY_RST is active HIGH. This bit is an alias of CSR116, bit 0.

Downloaded from Arrow.com.

CMD7: Command7

Offset 064h

CMD7 is a command-style register. All bits in this register are cleared to 0 when power is first applied to the device (power-on reset). The contents of this register are not affected by the state of the $\overline{RST}$ pin. The contents of this register can not be loaded from the EEPROM. Therefore, the contents of this register are not disturbed when PCI bus power is removed and re-applied.

Table 45. CMD7: Command7 Register

Bit	Name	Description
31-8	RES	Reserved locations. Written as zeros and read as undefined.
7	VAL0	Value bit for byte 0. The value of this bit is written to any bits in the CMD7 register that correspond to bits in the CMD7[6:0] bit map field that are set to 1.
6-4	RES	Reserved locations. Written as zeros and read as undefined.
3	PMAT_MODE	Pattern Match Mode. Writing a 1 to this bit will enable Pattern Match Mode and should only be done after the Pattern Match RAM has been programmed. Pattern Match Mode is enabled when either this bit or BCR45, bit 7 is set.
2	MPPEN_SW	Magic Packet Pin Enable. When either this bit or the MPPEN_EE bit in CMD3 is set, the device enters the Magic Packet mode when the PG input goes LOW. This bit has the same function as MPPEN_EE except that H_RESET clears MPPEN_EE to 0, while H_RESET has no effect on MPPEN_SW.
1	MPEN_SW	Magic Packet Software Mode Enable. The Am79C976 controller enters the Magic Packet mode when this bit is set to 1. This bit has the same function as MPEN_EE except that H_RESET clears MPEN_EE to 0, while H_RESET has no effect on MPEN_SW.
0	LCMODE_SW	Link Change Wake-up Mode. When either this bit or the LCMODE_EE bit in CMD3 is set to 1, the LCDET bit gets set when the MII auto polling logic detects a Link Change. This bit has the same function as LCMODE_EE except that H_RESET clears LCMODE_EE to 0, while H_RESET has no effect on LCMODE_SW.

Downloaded from Arrow.com.

CTRL0: Control0 Register

Offset 068h

This register contains several miscellaneous control bits. Each byte of this register controls a single function. It is not necessary to do a read-modify-write operation to change a function's settings if only a single byte of the register is written.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits except for bits 11:8 (ROMTMG) is 0. The default value for the ROMTMG field is 1001b.

Table 46. CTRL0: Control0 Register

Bit	Name	Description
31-25	RES	Reserved locations. Written as zeros and read as undefined.
24	BSWP	Byte Swap. This bit is used to choose between big and little Endian modes of operation. When BSWP is set to a 1, big Endian mode is selected. When BSWP is set to 0, little Endian mode is selected. When big Endian mode is selected, the Am79C976 controller will swap the order of bytes on the AD bus during a data phase on accesses to the FIFOs only. Specifically, AD[31:24] becomes Byte 0, AD[23:16] becomes Byte 1, AD[15:8] becomes Byte 2, and AD[7:0] becomes Byte 3 when big Endian mode is selected. When little Endian mode is selected, the order of bytes on the AD bus during a data phase is: AD[31:24] is Byte 3, AD[23:16] is Byte 2, AD[15:8] is Byte 1, and AD[7:0] is Byte 0. Byte swap only affects data transfers that involve the FIFOs. Initialization block transfers are not affected by the setting of the BSWP bit. Descriptor transfers are not affected by the setting of the BSWP bit. RDP, RAP, BDP and PCI configuration space accesses are not affected by the setting of the BSWP bit. Address PROM transfers are not affected by the setting of the BSWP bit. Expansion ROM accesses are not affected by the setting of the BSWP bit. Note that the byte ordering of the PCI bus is defined to be little Endian. BSWP should not be set to 1 when the Am79C976 controller is used in a PCI bus application. This bit is an alias of CSR3, bit 2.
23:18	RES	Reserved locations. Written as zeros and read as undefined.
17-16	SRAM_TYPE	SSRAM Type. This field must be set up to indicate the type of external SSRAM that is connected to the external memory interface. **SRAM_TYPE[1:0]** / **External Memory Type** 00 – Reserved 01 – ZBT 10 – Reserved 11 – Pipelined Burst
15-12	RES	Reserved locations. Written as zeros and read as undefined.

Downloaded from Arrow.com.

Bit	Name	Description
11-8	ROMTMG	Expansion ROM Timing. The value of ROMTMG is used to tune the timing for all accesses to the external Flash/EPROM. ROMTMG defines the amount of time that a valid address is driven on the ERA[19:0] pins. The register value specifies delay in number of ROMCLK cycles, where ROMCLK is an internal clock signal that runs at one fourth the speed of ERCLK. ***Note**: Programming ROMTMG with a value of 0 is not permitted.* To ensure adequate expansion ROM setup time, ROMTMG should be set to 1 plus tACC / (ROMCLK period), where tACC is the access time of the expansion ROM device (Flash or EPROM). (The extra ROMCLK cycle is added to account for the ERA[19:0] output delay from ROMCLK plus the ERD[7:0] setup time to ROMCLK.) ROMTMG is set to the default value of 1001b by H_RESET. It is also set to its default value before the EEPROM read process starts or when the EEPROM read process fails. The default value allows using an Expansion ROM with an access time of 350 ns if ERCLK is running at 90 MHz. This field is an alias of BCR18, bits 15:12.
7-5	RES	Reserved locations. Written as zeros and read as undefined.
4	BURST_ALIGN	PCI Bus Burst Align. When this bit is set, if a burst transfer starts in the middle of a cache line, the transfer will stop at the first cache line boundary.
3-0	BURST_LIMIT	PCI Bus Burst Limit. This 4-bit field limits the maximum length of a burst transfer. If the contents of this register are 0, the burst length is limited by the amount of data available or by the amount of FIFO space available. If the contents of this field are not zero, a burst transfer will end when the transfer has crossed the number of cache line boundaries equal to the contents of this field.

CTRL1: Control1 Register

Offset 06Ch

This register contains several miscellaneous control bits. Each byte of this register controls a single function. It is not necessary to do a read-modify-write operation to change a function's settings if only a single byte of the register is written.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits except for bits 17:16 (XMTSP) and bits 1:0 (RCVFW) is 0. The default value for the XMTSP field is 01b (64 bytes). The default value for the RCVFW field is also 01b.

Table 47. CTRL1: Control1 Register

Bit	Name	Description
31-26	RES	Reserved locations. Written as zeros and read as undefined.
25-24	SLOTMOD	Slot Time Modulation. This field determines the value of the slot time parameter used for the MAC half-duplex backoff algorithm. **Value** / **Slot Time (bits)** 00 — 512 (standard) 01 — 256 10 — 1024 11 — Reserved If SLOTMOD is set to anything other than the default value of 0, the controller will not conform to IEEE Std 802.3.
23:18	RES	Reserved locations. Written as zeros and read as undefined.

Downloaded from Arrow.com.

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>17-16</td><td>XMTSP</td><td>Transmit Start Point. XMTSP controls the point at which preamble transmission attempts to commence in relation to the number of bytes written to the MAC Transmit FIFO for the current transmit frame. When the entire frame is in the MAC Transmit FIFO, transmission will start regardless of the value in XMTSP.
If any of REX_UFLO, REX_RTRY, or RTRY_LCOL are set, no frame data will be overwritten until the frame has been transmitted or discarded. Otherwise, no data will be overwritten until at least 64 bytes have been transmitted.
Note that when the No Underflow (NOUFLO) bit (BCR18, bit 11) is set to 1 or XMTSP = 11b, transmission will not start until the complete frame has been copied into the Transmit FIFO. This mode is useful in a system where high latencies cannot be avoided.

<table><tr><th>XMTSP[1:0]</th><th>NOUFLO</th><th>Bytes Written</th></tr><tr><td>00</td><td>0</td><td>16</td></tr><tr><td>01</td><td>0</td><td>64</td></tr><tr><td>10</td><td>0</td><td>128</td></tr><tr><td>11</td><td>0</td><td>Full Frame</td></tr><tr><td>XX</td><td>1</td><td>Full Frame</td></tr></table>
The default value for XMTSP[1:0] is 01b (64 bytes).
This field is an alias of CSR80, bits 11:10.</td></tr>
<tr><td>15-10</td><td>RES</td><td>Reserved locations. Written as zeros and read as undefined.</td></tr>
<tr><td>9-8</td><td>XMTFW</td><td>Transmit FIFO Watermark. XMTFW specifies the point at which transmit DMA is requested, based upon the number of bytes that could be written to the Transmit FIFO without FIFO overflow. Transmit DMA is requested when the number of bytes specified by XMTFW could be written to the FIFO without causing Transmit FIFO overflow, and the internal state machine has reached a point where the Transmit FIFO is checked to determine if DMA servicing is required.

<table><tr><th>XMTFW[1:0]</th><th>Bytes Available</th></tr><tr><td>00</td><td>16</td></tr><tr><td>01</td><td>64</td></tr><tr><td>10</td><td>128</td></tr><tr><td>11</td><td>256</td></tr></table>
The default value for XMTFW[1:0] is 00b (16 bytes).
This field is an alias of CSR80, bits 9:8.</td></tr>
</table>

Downloaded from Arrow.com.

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>7-2</td><td>RES</td><td>Reserved locations. Written as zeros and read as undefined.</td></tr>
<tr><td>1-0</td><td>RCVFW</td><td>Receive FIFO Watermark. RCVFW controls the point at which receive DMA is requested in relation to the number of received bytes in the Receive FIFO. RCVFW specifies the number of bytes which must be present (once the frame has been verified as a non-runt) before receive DMA is requested. Note, however, that if the network interface is operating in half-duplex mode, in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively avoids having to react to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature is enabled or if the network interface is operating in full-duplex mode, receive DMA will be requested as soon as either the RCVFW threshold is reached, or a complete valid receive frame is detected (regardless of length). When the Full Duplex Runt Packet Accept Disable (FDRPAD) bit in CMD2 is set and the Am79C976 controller is in full-duplex mode, in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively disables the runt packet accept feature in full-duplex mode.
<table>
<tr><th>RCVFW[1:0]</th><th>Bytes Received</th></tr>
<tr><td>00</td><td>48</td></tr>
<tr><td>01</td><td>64</td></tr>
<tr><td>10</td><td>128</td></tr>
<tr><td>11</td><td>256</td></tr>
</table>
The default value for RCVFW[1:0] is 01b (64 bytes).
This field is an alias of CSR80, bits 13:12.</td></tr>
</table>

Downloaded from Arrow.com.

CTRL2: Control2 Register

Offset 070h

This register contains several miscellaneous control bits. Each byte of this register controls a single function. It is not necessary to do a read-modify-write operation to change a function's settings if only a single byte of the register is written.

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits except for bits 2:0 (APDW) is 0. The default value for the APDW field is 100b.

Table 48. CTRL2: Control2 Register

Bit	Name	Description
31-19	RES	Reserved locations. Written as zeros and read as undefined.
18-16	FS	Force Speed. This three-bit field sets the MAC's internal speed indicator according to the table below. The speed indication is used only for LEDs. **FS[2:0] / Speed** 000 – Speed determined by PHY 001 – Reserved 010 – 10 Mb/s 011 – 100 Mb/s 1XX – Reserved
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	FMDC	Fast Management Data Clock. When FMDC is set to 2h the MII Management Data Clock will run at 10 MHz max. The Management Data Clock will no longer be IEEE 802.3u-compliant and setting this bit should be used with care. The accompanying external PHY must also be able to accept management frames at the new clock rate. When FMDC is set to 1h, the MII Management Data Clock will run at 5 MHz max. The Management Data Clock will no longer be IEEE 802.3u-compliant and setting this bit should be used with care. The accompanying external PHY must also be able to accept management frames at the new clock rate. When FMDC is set to 0h, the MII Management Data Clock will run at 2.5 MHz max and will be fully compliant to IEEE 802.3u standards. This field is an alias of BCR32, bits13:12
7	RES	Reserved location. Written as zero and read as undefined.
6	XPHYRST	External PHY Reset. When XPHYRST is set, the Am79C976 controller after an H_RESET or S_RESET will issue an MII management frame that will reset the external PHY. This bit is needed when there is no way to guarantee the state of the external PHY. This bit must be reprogrammed after every H_RESET. XPHYRST is only valid when the internal Network Port Manager is scanning for a network port. This bit is an alias of BCR32, bit 6.
5	XPHYANE	External PHY Auto-Negotiation Enable. This bit will force the external PHY into enabling Auto-Negotiation. When set to 0 the Am79C976 controller will send an MII management frame disabling Auto-Negotiation. XPHYANE is only valid when the internal Network Port Manager is scanning for a network port. This bit is an alias of BCR32, bit 5.
4	XPHYFD	External PHY Full Duplex. When set, this bit will force the external PHY into full duplex when Auto-Negotiation is not enabled. XPHYFD is only valid when the internal Network Port Manager is scanning for a network port. This bit is an alias of BCR32, bit 4.

Downloaded from Arrow.com.

Bit	Name	Description
3	XPHYSP	External PHY Speed. When set, this bit will force the external PHY into 100 Mbps mode when Auto-Negotiation is not enabled. XPHYSP is only valid when the internal Network Port Manager is scanning for a network port. This bit is an alias of BCR32, bit 3.
2-0	APDW	MII Auto-Poll Dwell Time. APDW determines the dwell time between MII Management Frames accesses when Auto-Poll is turned on. The MDC clock is ERCLK divided by 40 and will be slower than 2.5MHz. The actual interval between polls will be the time shown in the following table multiplied by 100 / (ERCLK frequency in MHz). **APDW** / **Auto-Poll™ Dwell Time** 000 — Continuous (27µs @ 2.5 MHz) 001 — Every 64 MDC cycles (77 µs @ 2.5 MHz) 010 — Every 128 MDC cycles (129 µs @ 2.5 MHz) 011 — Every 256 MDC cycles (232 µs @ 2.5 MHz) 100 — Every 512 MDC cycles (436 µs @ 2.5 MHz) 101 — Every 1024 MDC cycles (845 µs @ 2.5 MHz) 110-111 — Reserved The default value of this field is 100b. This field is an alias of BCR32, bits 10:8.

CTRL3: Control3 Register

Offset 074h

All bits in this register are set to their default values by H_RESET. All bits are also set to their default values before EEPROM data are loaded or after an EEPROM read failure.

The default value for all bits is 0.

Table 49. CTRL3: Control3 Register

Bit	Name	Description
31-8	RES	Reserved locations. Written as zeros and read as undefined.
7-0	SWSTYLE	Software Style register. The value in this register determines the style of register and memory resources that shall be used by the Am79C976 controller. The Software Style selection will affect the interpretation of a few bits within the CSR space, the order of the descriptor entries, and the width of the descriptors and initialization block entries. All Am79C976 controller CSR bits and BCR bits and all descriptor, buffer, and initialization block entries not cited in Table 50 are unaffected by the Software Style selection and are, therefore, always fully functional as specified in the CSR and BCR sections.

Downloaded from Arrow.com.

Table 50. Software Styles

SWSTYLE [7:0]	Style Name	SSIZE32	Initialization Block Entries	Descriptor Ring Entries
00h	LANCE/ PCnet-ISA controller	0	16-bit software structures, non-burst or burst access	16-bit software structures, non-burst access only
01h	RES	1	RES	RES
02h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only
03h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst or burst access
04h	VLAN	1	Not used	32-bit software structures, non-burst or burst access
05h	64-bit address	1	Not used	32-bit software structures, 32-byte descriptors, non-burst or burst access
All Other	Reserved	Undefined	Undefined	Undefined

DATAMBIST: Memory Built-in Self-Test Access Register

Offset 1A0h

This register is used to control and indirectly access the Memory Built-in Self-Test (MBIST) logic that automatically tests the external SSRAM.

The contents of this register are cleared to 0 when the RST pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 51. DATAMBIST: Memory Built-in Self-Test Access Register

Bit	Name	Description
63	DM_DONE	MBIST done indicator. This bit is set to 1 when the automatic memory test has stopped, either because the test has completed or because an error was detected. It is cleared to 0 when either DM_START or DM_RESUME is set. This bit is read-only.
62	DM_ERROR	MBIST error indicator. This bit is set to 1 when the memory test logic has detected a memory error. It is cleared to 0 when either DM_START or DM_RESUME is set. This bit is read-only.
61	DM_START	MBIST Start. Setting this bit to 1 resets the MBIST logic, including the DM_ERROR and DM_TEST_FAIL bits, and starts the memory test process. DM_START should not be set at the same time that the DM_RESUME bit is set. DM_START is automatically cleared when the memory test stops. This bit is read/write.
60	DM_RESUME	MBIST Resume. Setting this bit to 1 restarts the memory test sequence at the point where it last stopped. Setting this bit clears the DM_ERROR bit, but it does not clear the DM_TEST_FAIL bit. This bit should not be set at the same time that the DM_START bit is set. DM_RESUME is automatically cleared when the memory test stops. This bit is read/write.
59	DM_FAIL_STOP	MBIST Stop on Failure Control. When this bit is set to 1, the memory test will stop each time an error is detected. When this bit is cleared to 0, the memory test will run to completion, regardless of the number of errors that are detected. This bit is read/write.

Downloaded from Arrow.com.

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>58</td><td>DM_TEST_FAIL</td><td>MBIST Test Failure Indicator. This bit is set when a memory test error is detected. It is reset when DM_START is set to 1. It is not cleared when DM_RESUME is set to 1.
This bit is read-only.</td></tr>
<tr><td>57</td><td>RES</td><td>Reserved. Written as 0, read as undefined.</td></tr>
<tr><td>56</td><td>DM_DIR</td><td>MBIST Test Direction. This bit is set to 1 when the MBIST memory pointer was counting down when the test stopped. It is cleared to 0 when the MBIST memory pointer was counting up when the test stopped.
This bit is read-only.</td></tr>
<tr><td>55-54</td><td>DM_FAIL_STATE</td><td>MBIST Error Offset Indicator. This field indicates offset of the location of the last memory test error with respect to the value of the DM_ADDR field. The interpretation of this field depends on the value of the DM_DIR field, which indicates whether the address pointer was counting up or down when the error was detected.
<table>
<tr><th>DM_DIR</th><th>FAIL_STATE</th><th>Error Location</th></tr>
<tr><td>0</td><td>00</td><td>Error at DM_ADDRESS</td></tr>
<tr><td>0</td><td>01</td><td>Error at DM_ADDRESS-1</td></tr>
<tr><td>0</td><td>10</td><td>Error at DM_ADDRESS-2</td></tr>
<tr><td>0</td><td>11</td><td>Error at DM_ADDRESS-3</td></tr>
<tr><td>1</td><td>00</td><td>Error at DM_ADDRESS</td></tr>
<tr><td>1</td><td>01</td><td>Error at DM_ADDRESS+1</td></tr>
<tr><td>1</td><td>10</td><td>Error at DM_ADDRESS+2</td></tr>
<tr><td>1</td><td>11</td><td>Error at DM_ADDRESS+3</td></tr>
</table>
This field is read-only.</td></tr>
<tr><td>53-52</td><td>DM_BACKG</td><td>MBIST Background. This field contains the background pattern that the memory test logic was using when the test stopped.
This field is read-only.</td></tr>
<tr><td>51-32</td><td>DM_ADDR</td><td>MBIST Address. This field contains the contents of the MBIST address pointer at the time that the test stopped. Because of the pipelined nature of the external SSRAM, this value may not be the location of the memory error. The actual error location is obtained by adding or subtracting the contents of the DM_FAIL_STATE field as described above.
This field is read-only.</td></tr>
<tr><td>31-0</td><td>DM_DATA</td><td>MBIST Data. This field contains the last data that the memory test logic read from the external SSRAM. If the DM_ERR and DM_FAIL_STOP bits are both set to 1, the contents of this field contains an error.
This field is read-only.</td></tr>
</table>

Downloaded from Arrow.com.

Reerved Register

Offset 0C0h

The contents of this register are written as 0, and read back as undefined.

Table 52. Reserved Register

Bit	Name	Description
31-21	RES	Reserved. Written as 0, read as undefined.
20-16	RES	Reserved. Written as 0, read as undefined.
15-11	RES	Reserved. Written as 0, read as undefined.
10-0	RES	Reserved. Written as 0, read as undefined.

Downloaded from Arrow.com.

EEPROM_ACC: EEPROM Access Register

Offset 17Ch

The contents of this register are set to default values when the RST pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error. The default value for all bits in this register is 0.

This register is an alias of BCR19.

Table 53. EEPROM_ACC: EEPROM Access Register

Bit	Name	Description
15	PVALID	EEPROM Valid status bit. PVALID is read only; write operations have no effect. A value of 1 in this bit indicates that a PREAD operation has occurred, and that (1) there is an EEPROM connected to the Am79C976 controller interface pins and (2) the contents read from the EEPROM have passed the CRC verification operation. A value of 0 in this bit indicates a failure in reading the EEPROM. The CRC for the EEPROM is incorrect or no EEPROM is connected to the interface pins. PVALID is set to 0 during H_RESET. However, following the H_RESET operation, an automatic read of the EEPROM will be performed. Just as is true for the normal PREAD command, at the end of this automatic read operation, the PVALID bit may be set to 1. Therefore, H_RESET will set the PVALID bit to 0 at first, but the automatic EEPROM read operation may later set PVALID to a 1. If PVALID becomes 0 following an EEPROM read operation (either automatically generated after H_RESET, or requested through PREAD), then all EEPROM-programmable registers will be reset to their default values. The content of the Address PROM locations, however, will not be cleared. If no EEPROM is present at the EESK, EEDI, and EEDO pins, then all attempted PREAD commands will terminate early and PVALID will *not* be set. This applies to the automatic read of the EEPROM after H_RESET, as well as to host-initiated PREAD commands.
14	PREAD	EEPROM Read command bit. When this bit is set to a 1 by the host, the PVALID bit (bit 15) will immediately be reset to a 0, and then the Am79C976 controller will perform a read operation from the external serial EEPROM. The EEPROM data that is fetched during the read will be stored in the appropriate internal registers on board the Am79C976 controller. Upon completion of the EEPROM read operation, the Am79C976 controller will assert the PVALID bit. At the end of the read operation, the PREAD bit will automatically be reset to a 0 by the Am79C976 controller and PVALID will be set, provided that an EEPROM existed on the interface pins and that the CRC for the EEPROM was correct. Note that when PREAD is set to a 1, then the Am79C976 controller will no longer respond to any accesses directed toward it until the PREAD operation has completed successfully. The Am79C976 controller will terminate these accesses with the assertion of DEVSEL and STOP while TRDY is not asserted, signaling to the initiator to disconnect and retry the access at a later time. If a PREAD command is given to the Am79C976 controller but no EEPROM is attached to the interface pins, the PREAD bit will be cleared to a 0, and the PVALID bit will remain reset with a value of 0. This applies to the automatic read of the EEPROM after H_RESET as well as to host initiated PREAD commands. All EEPROM programmable registers will be set to their default values by such an aborted PREAD operation. At the end of the read operation, if bit 15 of the PMC Alias register is zero or the VAUX_SENSE pin is low, the PME_STATUS and PME_EN bits of the PMCSR register will be reset.
13	EEDET	EEPROM Detect. This bit indicates whether or not an EEPROM was detected by the ERPROM read operation. If this bit is a 1, it indicates that an EEPROM was detected. If this bit is a 0, it indicates that an EEPROM was not detected. EEDET is read only; write operations have no effect. The value of this bit is determined at the end of the H_RESET operation.
12-5	RES	Reserved locations. Written as zeros and read as undefined.

Downloaded from Arrow.com.

Bit	Name	Description
4	EEN	EEPROM Port Enable. When this bit is set to a 1, it causes the values of ECS, ESK, and EDI to be driven onto the EECS, EESK, and EEDI pins, respectively. If EEN = 0 and no EEPROM read function is currently active, then EECS will be driven LOW. When EEN = 0 and no EEPROM read function is currently active, EESK and EEDI pins will be driven by the LED1 and LED0 functions, respectively. See Table 54.
3	RES	Reserved location. Written as 0; read as undefined.
2	ECS	EEPROM Chip Select. This bit is used to control the value of the EECS pin of the interface when the EEN bit is set to 1 and the PREAD bit is set to 0. If EEN = 1 and PREAD = 0 and ECS is set to a 1, then the EECS pin will be forced to a HIGH level at the rising edge of the next clock following bit programming. If EEN = 1 and PREAD = 0 and ECS is set to a 0, then the EECS pin will be forced to a LOW level at the rising edge of the next clock following bit programming. ECS has no effect on the output value of the EECS pin unless the PREAD bit is set to 0 and the EEN bit is set to 1.
1	ESK	EEPROM Serial Clock. This bit and the EDI/EDO bit are used to control host access to the EEPROM. Values programmed to this bit are placed onto the EESK pin at the rising edge of the next clock following bit programming, except when the PREAD bit is set to 1 or the EEN bit is set to 0. If both the ESK bit and the EDI/EDO bit values are changed during the same register write operation, while EEN = 1, then setup and hold times of the EEDI pin value with respect to the EESK signal edge are not guaranteed. ESK has no effect on the EESK pin unless the PREAD bit is set to 0 and the EEN bit is set to 1.
0	EDI/EDO	EEPROM Data In/EEPROM Data Out. Data that is written to this bit will appear on the EEDI output of the interface, except when the PREAD bit is set to 1 or the EEN bit is set to 0. Data that is read from this bit reflects the value of the EEDO input of the interface. EDI/EDO has no effect on the EEDI pin unless the PREAD bit is set to 0 and the EEN bit is set to 1.

Table 54. Interface Pin Assignment

$\overline{\text{RST}}$ Pin	PREAD or Auto Read in Progress	EEN	EECS	EESK	EEDI
Low	X	X	0	Tri-State	Tri-State
High	1	X	Active	Active	Active
High	0	1	From ECS Bit	From ESK Bit	From EDI Bit
High	0	0	0	LED1	LED0

Downloaded from Arrow.com.

FLASH_ADDR: Flash Address Register

Offset 198h

The contents of this register are cleared to 0 when the RST pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 55. FLASH_ADDR: Flash Address Register

Bit	Name	Description
31	LAAINC	Lower Address Auto Increment. When the LAAINC bit is set to 1, the low-order 16-bit portion of the Flash Address Register will automatically increment by one after a read or write access to the Flash Data Register. When the low-order 16-bit portion of the Flash Address reaches FFFFh and LAAINC is set to 1, the low-order 16-bit portion of the Flash Address will roll over to 0000h. When the LAAINC bit is set to 0, the Flash Address Register will not be affected in any way after an access to the Flash Data Register. This bit is an alias of BCR29, bit 14.
30-24	RES	Reserved locations. Written as zeros and read as undefined.
23-0	FLASH_ ADDR	Flash Address. This field selects the byte of the external Flash/ROM device that will be accessed when the Flash Data Register is accessed. This field is an alias of BCR29, bits [7:0] concatenated with BCR28, bits [15:0].

FLASH_DATA: Flash Data Register

Offset 19Ch

The contents of this register are cleared to 0 when the RST pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

This register is an alias of BCR30.

Table 56. FLASH_DATA: Flash Data Register

Bit	Name	Description
15-8	RES	Reserved locations. Written as zeros and read as undefined.
7-0	FLASH_DATA	Flash Data. This field contains data written to or read from the external Flash/ROM device. When the host CPU writes to this register, the contents of this field are written to the external Flash device at the location selected by the Flash Address Register. When the host CPU reads this register, the Am79C976 device first reads the location in the Flash device selected by the Flash Address Register, then returns the value read in this field.

Downloaded from Arrow.com.

FLOW: Flow Control Register

Offset 0C8h

FLOW is a command-style register. All bits in this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 57. FLOW: Flow Control Register

Bit	Name	Description
31-24	RES	Reserved locations. Written as zeros and read as undefined.
23	VAL2	Value bit for byte 2. The value of this bit is written to any bits in the FLOW register that correspond to bits in the FLOW[22:16] bit map field that are set to 1.
22-21	RES	Reserved locations. Written as zeros and read as undefined.
20	FPA	Force Pause Ability. When this bit is set, Pause Ability is enabled regardless of the Pause Ability state of the external PHY's link partner. When Pause Ability is enabled, the receipt of a MAC Control Pause Frame causes the device to stop transmitting for a time period that is determined by the contents of the Pause Frame. If a logical 1 is written to this bit position, the corresponding bit in the register will be loaded with the contents of the VAL2 bit. If a logical 0 is written to this bit position, the corresponding bit in the register will not be altered.
19	NPA	Negotiate Pause Ability. When this bit is set and the Force Pause Ability bit is not set, Pause Ability is enabled only if the auto-negotiation process determines that the external PHY's link partner supports IEEE 802.3 flow control. When Pause Ability is enabled, the receipt of a MAC Control Pause Frame causes the device to stop transmitting for a time period that is determined by the contents of the Pause Frame. If a logical 1 is written to this bit position, the corresponding bit in the register will be loaded with the contents of the VAL2 bit. If a logical 0 is written to this bit position, the corresponding bit in the register will not be altered.
18	FIXP	Fixed Length Pause. When this bit is set to 1, all MAC Control Pause Frames transmitted from the device will contain a Request_operand field that is copied from the PAUSE_LEN field of this register. When this bit is cleared to 0, a Pause Frame with its Request_operand field set to 0FFFFh will be sent when the FCCMD bit in this register is changed from 0 to 1 or when the signal on the FC pin changes from 0 to 1 while the FCPEN bit has the value 1. Also a Pause Frame with its Request_operand field set to 0000h will be sent when the FCCMD bit in this register is changed from 1 to 0 or when the signal on the FC pin changes from 1 to 0 while the FCPEN bit has the value 1. If a logical 1 is written to this bit position, the corresponding bit in the register will be loaded with the contents of the VAL2 bit. If a logical 0 is written to this bit position, the corresponding bit in the register will not be altered.
17	FCPEN	Flow Control Pin Enable. When the value of this bit is 1, MAC Control Pause frames will be transmitted or half-duplex back pressure will be applied when the FC pin is asserted. When the value of this bit is 0, the state of the FC pin is ignored. If a logical 1 is written to this bit position, the corresponding bit in the register will be loaded with the contents of the VAL2 bit. If a logical 0 is written to this bit position, the corresponding bit in the register will not be altered.

Downloaded from Arrow.com.

Bit	Name	Description
16	FCCMD	Flow Control Command. In full-duplex mode this bit allows the host CPU to cause a Pause Frame to be sent by issuing a single command. In half-duplex mode, setting this bit puts the device into back-pressure mode, which has the effect of forcing the remote transmitter to delay transmissions while the local system is freeing up congested resources. If the device is operating in full-duplex mode and the value of the FIXP bit is 1, when FCCMD is changed from 0 to 1, a Pause Frame is sent with its Request_operand field copied from the PAUSE_LEN field of this register. After the Pause Frame is sent, the FCCMD bit is automatically reset to 0. If the device is operating in full-duplex mode and the value of the FIXP bit is 0, when FCCMD is changed from 0 to 1, a Pause Frame is sent with its Request_operand field filled with all 1s, and the FCCMD bit is not automatically reset to 0. When FCCMD is changed from 1 to 0, a Pause Frame is sent with its Request_operand field filled with all 0s. If the device is operating in half-duplex mode, setting FCCMD to 1 puts the MAC into back-pressure mode. In back-pressure mode, whenever the MAC detects receiver activity, it transmits a series of alternating 1s and 0s to force a collision. If a logical 1 is written to this bit position, the corresponding bit in the register will be loaded with the contents of the VAL2 bit. If a logical 0 is written to this bit position, the corresponding bit in the register will not be altered.
15-0	PAUSE_LEN	Pause Length. The contents of this field are copied into the Request_operand fields of MAC Control Pause Frames that are transmitted while the FIXP bit in this register is set to 1.

IFS1: Inter-Frame Spacing Part 1 Register

Offset 18Ch

The contents of this register are set to 3ch when the $\overline{\text{RST}}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 58. IFS1: Inter-Frame Spacing Part 1 Register

Bit	Name	Description
7-0	IFS1	InterFrameSpacingPart1. Changing IFS1 allows the user to program the value of the InterFrame-SpacePart1 timing. The Am79C976 controller sets the default value at 60 bit times (3ch). See the subsection on *Medium Allocation* in the section *Media Access Management* for more details. The equation for setting IFS1 when IPG ≥ 96 bit times is: IFS1 = IPG - 36 bit times IPG should be programmed to the nearest nibble. The two least significant bits are ignored. For example, programming IPG to 63h has the same effect as programming it to 60h. This register is an alias for CSR125, bits [7:0].

Downloaded from Arrow.com.

INT0: Interrupt0

Offset 038h

INT0 identifies the source or sources of an interrupt. With the exception of INTR, all bits in this register are "write 1 to clear" so that the CPU can clear the interrupt condition by reading the register and then writing back the same data that it read. Writing a 0 to a bit in this register has no effect.

All bits in this register are cleared to 0 by H_RESET. In addition, TINT, TXDNINT, TXSTRTINT, and RINT are cleared when the RUN bit in CMD0 is cleared.

Table 59. INT0: Interrupt0 Register

Bit	Name	Description
31	INTR	Interrupt Summary. This bit indicates that one or more of the other interrupt bits in this register are set and the associated enable bit or bits in INTEN0 are also set. If INTREN in CMD0 is set to 1 and INTR is set, $\overline{INTA}$ will be active. When INTR is set by SINT, $\overline{INTA}$ will be active independent of the state of INEA. INTR is read only. INTR is cleared by clearing all of the active individual interrupt bits that have not been masked out.
31-28	RES	Reserved locations. Written as zeros and read as undefined.
27	LCINT	Link Change Interrupt. This bit is set when the Port Manager detects a change in the link status of the external PHY.
26	APINT5	Auto-Poll Interrupt from Register 5. This bit is set when the Auto-Poll State Machine has detected a change in the external PHY register whose address is stored in Auto-Poll Register 5.
25	APINT4	Auto-Poll Interrupt from Register 4. This bit is set when the Auto-Poll State Machine has detected a change in the external PHY register whose address is stored in Auto-Poll Register 4.
24	APINT3	Auto-Poll Interrupt from Register 3. This bit is set when the Auto-Poll State Machine has detected a change in the external PHY register whose address is stored in Auto-Poll Register 3.
23	RES	Reserved locations. Written as zeros and read as undefined.
22	APINT2	Auto-Poll Interrupt from Register 2. This bit is set when the Auto-Poll State Machine has detected a change in the external PHY register whose address is stored in Auto-Poll Register 2.
21	APINT1	Auto-Poll Interrupt from Register 1. This bit is set when the Auto-Poll State Machine has detected a change in the external PHY register whose address is stored in Auto-Poll Register 1.
20	APINT0	Auto-Poll Interrupt from Register 0. This bit is set when the Auto-Poll State Machine has detected a change in the external PHY register whose address is stored in Auto-Poll Register 0.
19	MIIPDTINT	MII PHY Detect Transition Interrupt. The MII PHY Detect Transition Interrupt is set by the Am79C976 controller whenever the MIIPD bit in STAT0 transitions from 0 to 1 or vice versa. This bit is an alias of CSR7, bit 1.
18	MCCIINT	MII Management Command Complete Internal Interrupt. The MII Management Command Complete Interrupt is set by the Am79C976 controller when a read or write operation on the MII management port is complete from an internal operation. Examples of internal operations are Auto-Poll or Network Port Manager generated MII management frames. This bit is an alias of CSR7, bit 3.
17	MCCINT	MII Management Command Complete Interrupt. The MII Management Command Complete Interrupt is set by the Am79C976 controller when a read or write operation to the MII Data Port (PHY Access Register) is complete. This bit is an alias of CSR7, bit 5.

Downloaded from Arrow.com.

Bit	Name	Description
16	MREINT	MII Management Read Error Interrupt. The MII Read Error interrupt is set by the Am79C976 controller to indicate that the currently read register from the external PHY is invalid. The contents of the PHY Access Register are incorrect and that the operation should be performed again. The indication of an incorrect read comes from the PHY. During the read turnaround time of the MII management frame the external PHY should drive the MDIO pin to a LOW state. If this does not happen, it indicates that the PHY and the Am79C976 controller have lost synchronization. This bit is an alias of CSR7, bit 9
15	RES	Reserved locations. Written as zeros and read as undefined.
14	SPNDINT	Suspend Interrupt. This bit is set when a receiver or transmitter suspend operation has finished.
13	MPINT	Magic Packet Interrupt. Magic Packet Interrupt is set by the Am79C976 controller when the device is in the Magic Packet mode and the Am79C976 controller receives a Magic Packet frame. This bit is an alias of CSR5, bit 4.
12	SINT	System Interrupt is set by the Am79C976 controller when it detects a system error during a bus master transfer on the PCI bus. System errors are data parity error, master abort, or a target abort. The setting of SINT due to data parity error is not dependent on the setting of PERREN (PCI Command register, bit 6). Note that because INEA is cleared by the STOP reset generated by the system error, the system interrupt bypasses the global interrupt enable bits INEA and INTREN. This means that if SINTEN in INTEN0 or SINTE in CSR5 is set to 1, $\overline{\text{INTA}}$ will be asserted when SINT is 1 regardless of the state of INEA and INTREN. The state of SINT is not affected by clearing any of the PCI Status register bits that get set when a data parity error (DATAPERR, bit 8), master abort (RMABORT, bit 13), or target abort (RTABORT, bit 12) occurs. This bit is an alias of CSR5, bit 11.
11-9	RES	Reserved locations. Written as zeros and read as undefined.
8	TINT	Transmit Interrupt is set by the Am79C976 controller after the OWN bit in the last descriptor of a transmit frame has been cleared to indicate the frame has been copied to the transmit FIFO. This bit is an alias of CSR0, bit 9.
7	UINT	User Interrupt. UINT is set by the Am79C976 controller after the host has issued a user interrupt command by setting UINTCMD in the CMD0 register. This bit is an alias of CSR4, bit 6.
6	TXDNINT	Transmission Done Interrupt. This bit is set when the transmitter has finished sending a frame. This bit is included for debugging purposes.
5	TXSTRTINT	Transmit Start Interrupt. This bit is set when the transmitter begins the transmission of a frame. This bit is included for debugging purposes. This bit is an alias of CSR4, bit 3.
4	STINT	Software Timer Interrupt. The Software Timer interrupt is set by the Am79C976 controller when the Software Timer counts down to 0. The Software Timer will immediately load the contents of the Software Timer Value Register, STVAL, into the Software Timer and begin counting down. This bit is an alias of CSR7, bit 11.
3-1	RES	Reserved locations. Written as zeros and read as undefined.
0	RINT	Receive Interrupt is set by the Am79C976 controller after the last descriptor of a receive frame has been updated by writing a 0 to the OWNership bit. RINT may also be set when the first descriptor of a receive frame has been updated by writing a 0 to the OWNership bit if the LAPPEN bit in CMD2 has been set to a 1. This bit is an alias of CSR0, bit 10.

Downloaded from Arrow.com.

INTEN0: Interrupt0 Enable

Offset 040h

This register allows the software to specify which types of interrupt events will cause the INTR bit in the Interrupt0 register to be set, which in turn will cause INTA pin to be asserted if the INTREN bit in CMD0 is set. Each bit in this register corresponds to a bit in the Interrupt0 register. Setting a bit in this register enables the corresponding bit in the Interrupt0 register to cause the INTR bit to be set.

INTEN0 is a command style register. The high order bit of each byte of this register is a 'value' bit that specifies the value that will be written to selected bits of the register. The seven low order bits of each byte make up a bit map that selects which register bits will be altered.

All bits in this register are cleared to 0 by H_RESET. All bits are also cleared before EEPROM data are loaded or after an EEPROM read failure.

The RINTEN and TINTEN bits are set after S_RESET (but not H_RESET).

Table 60. INTEN0: Interrupt0 Enable Register

Bit	Name	Description
31	VAL3	Value bit for byte 3. The value of this bit is written to any bits in the INTEN0 register that correspond to bits in the INTEN0[30:24] bit map field that are set to 1.
30-28	RES	Reserved locations. Written as zeros and read as undefined.
27	LCINTEN	Link Change Interrupt Enable. When this bit is set, the INTR bit will be set when the LCINT bit in INT0 is set.
26	APINT5EN	Auto-Poll Interrupt from Register 5 Enable. When this bit is set, the INTR bit will be set when the APINT5 bit in INT0 is set.
25	APINT4EN	Auto-Poll Interrupt from Register 4 Enable. When this bit is set, the INTR bit will be set when the APINT4 bit in INT0 is set.
24	APINT3EN	Auto-Poll Interrupt from Register 3 Enable. When this bit is set, the INTR bit will be set when the APINT3 bit in INT0 is set.
23	VAL2	Value bit for byte 2. The value of this bit is written to any bits in the INTEN0 register that correspond to bits in the INTEN0[22:16] bit map field that are set to 1.
22	APINT2EN	Auto-Poll Interrupt from Register 2 Enable. When this bit is set, the INTR bit will be set when the APINT2 bit in INT0 is set.
21	APINT1EN	Auto-Poll Interrupt from Register 1 Enable. When this bit is set, the INTR bit will be set when the APINT1 bit in INT0 is set.
20	APINT0EN	Auto-Poll Interrupt from Register 0 Enable. When this bit is set, the INTR bit will be set when the APINT0 bit in INT0 is set.
19	MIIPDTINTEN	MII PHY Detect Transition Interrupt Enable. When this bit is set, the INTR bit will be set when the MIIPDTINT bit in INT0 is set. This bit is an alias of CSR7, bit 0.
18	MCCIINTEN	MII Management Command Complete Internal Interrupt Enable. When this bit is set, the INTR bit will be set when the MCCIINT bit in INT0 is set. This bit is an alias of CSR7, bit 2.
17	MCCINTEN	MII Management Command Complete Interrupt Enable. When this bit is set, the INTR bit will be set when the MCCINT bit in INT0 is set. This bit is an alias of CSR7, bit 4.
16	MREINTEN	MII Management Read Error Interrupt Enable. When this bit is set, the INTR bit will be set when the MREINT bit in INT0 is set. This bit is an alias of CSR7, bit 8
15	VAL1	Value bit for byte 1. The value of this bit is written to any bits in the INTEN0 register that correspond to bits in the INTEN0[14:8] bit map field that are set to 1.
14	SPNDINTEN	Suspend Interrupt Enable. When this bit is set, the INTR bit will be set when the SPNDINT bit in INT0 is set.

Downloaded from Arrow.com.

Bit	Name	Description
13	MPINTEN	Magic Packet Interrupt Enable. When this bit is set, the INTR bit will be set when the MPINT bit in INT0 is set. This bit is an alias of CSR5, bit 3.
12	SINTEN	System Interrupt Enable. When this bit is set, the INTR bit will be set when the SINT bit in INT0 is set. This bit is an alias of CSR5, bit 10.
11-9	RES	Reserved locations. Written as zeros and read as undefined.
8	TINTEN	Transmit Interrupt Enable. When this bit is set, the INTR bit will be set when the TINT bit in INT0 is set. This bit is an alias of CSR3, bit 9 with reversed polarity. (When TINTM in CSR3 is set, the transmit interrupt is disabled.) Previous devices in the PCnet family enable receive and transmit interrupts following reset. The Am79C976 controller disables these interrupts following H_RESET (but not S_RESET). For compatibility with legacy software, the Am79C976 controller will set RINTEN and TINTEN following S_RESET. If, in addition, the user programs the EEPROM to load ones into RINTEN and TINTEN, these interrupts will also be enabled after H_RESET. This matches the behavior of the previous PCnet devices.
7	VAL1	Value bit for byte 1. The value of this bit is written to any bits in the INTEN0 register that correspond to bits in the INTEN0[6:0] bit map field that are set to 1.
6	TXDNINTEN	Transmission Done Interrupt Enable. When this bit is set, the INTR bit will be set when the TXDNINT bit in INT0 is set. This bit is an alias of CSR5, bit 12.
5	TXSTRTINTEN	Transmit Start Interrupt Enable. When this bit is set, the INTR bit will be set when the TXSTRTINT bit in INT0 is set. This bit is an alias of CSR4, bit 2.
4	STINTEN	Software Timer Interrupt Enable. When this bit is set, the INTR bit will be set when the STINT bit in INT0 is set. This bit is an alias of CSR7, bit 10.
3-1	RES	Reserved locations. Written as zeros and read as undefined.
0	RINTEN	Receive Interrupt Enable. When this bit is set, the INTR bit will be set when the RINT bit in INT0 is set. This bit is an alias of CSR3, bit 10 with reversed polarity. (When RINTM in CSR3 is set, the receive interrupt is disabled.) Previous devices in the PCnet family enable receive and transmit interrupts following reset. The Am79C976 controller disables these interrupts following H_RESET (but not S_RESET). For compatibility with legacy software, the Am79C976 controller will set RINTEN and TINTEN following S_RESET. If, in addition, the user programs the EEPROM to load ones into RINTEN and TINTEN, these interrupts will also be enabled after H_RESET. This matches the behavior of the previous PCnet devices.

Downloaded from Arrow.com.

IPG: Inter-Packet Gap Register

Offset 18Dh

The contents of this register are set to 60h when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 61. IPG: Inter-Packet Gap Register

Bit	Name	Description
7-0	IPG	Inter Packet Gap. This value indicates the minimum number of network bit times after the end of a frame that the transmitter will wait before it starts transmitting another frame. In half-duplex mode the end of the frame is determined by CRS, while in full-duplex mode the end of the frame is determined by TX_EN. The IPG value can be adjusted to compensate for delays through the external PHY device. IPG should be programmed to the nearest nibble. The two least significant bits are ignored. For example, programming IPG to 63h has the same effect as programming it to 60h. ***CAUTION**: Use this parameter with care. By lowering the IPG below the IEEE 802.3 standard 96 bit times, the Am79C976 controller can interrupt normal network behavior.* This register is an alias for CSR125, bits [15:8].

LADRF: Logical Address Filter Register

Offset 168h

The contents of this register are cleared to 0 when the RST pin is asserted. This register is not cleared by the serial EEPROM read operation or by a serial EEPROM read error.

Table 62. Logical Address Filter Register

Bit	Name	Description
63-0	LADRF	Logical Address Filter, LADRF[63:0]. This register contains a 64-bit mask that is used to accept incoming logical (or multicast) addresses. If the first bit in the incoming address (as transmitted on the wire) is a 1, the destination address is a logical address. A logical address is passed through the CRC generator to produce a 32-bit result. The high order 6 bits of this result 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 copied into host system 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 responsibility of the host CPU to compare the destination address of the stored message with a list of acceptable multicast addresses to determine whether or not the message is actually intended for the node. The contents of this register should be loaded from EEPROM. This register can also be loaded from the initialization block after the INIT bit in CSR0 has been set. This register is an alias for CSR8, CSR9, CSR10, and CSR11.

Downloaded from Arrow.com.

LED0 Control Register

Offset 0E0h

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

All bits in this register are restored to their default values when the $\overline{\text{RST}}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

This register is an alias of BCR4.

Table 63. LED0 Control Register

Bit	Name	Description
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of 1 in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (bits 8 and 6-0).
14	LEDPOL	LED Polarity. When this bit has the value 0, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false (i.e., the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit). When this bit has the value 1, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true, and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e., the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit.). The setting of this bit will not effect the polarity of the LEDOUT bit for this register. The default value of this bit is 0.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then the LED output value will be governed by the LEDOUT and LEDPOL values. The default value of this bit is 0.
12	100E	100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C976 controller is operating in 100 Mbps mode. The default value of this bit is 0.
11-10	RES	Reserved locations. Written and read as undefined.
9	MPSE	Magic Packet Status Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when Magic Packet frame mode is enabled and a Magic Packet frame is detected on the network. The default value of this bit is 0.
8	FDLSE	Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LEDOUT signal when the Am79C976 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C976 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal. The default value of this bit is 0.
7	PSE	Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher. The default value of this bit is 1.
6	LNKSE	Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register when in Link Pass state. The default value of this bit is 1.

Downloaded from Arrow.com.

Bit	Name	Description
5	RCVME	Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network that has passed the address match function for this node. All address matching modes are included: physical, logical filtering, broadcast and promiscuous. The default value of this bit is 0.
4	XMTE	Transmit Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is transmit activity on the network. The default value of this bit is 0.
3	RES	Reserved location. Written and read as undefined.
2	RCVE	Receive Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network. The default value of this bit is 0.
1	SFBDE	Start Frame/Byte Delimiter Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the RXD[3:0] pins are presenting the least significant nibble of valid frame data. The default value of this bit is 0.
0	COLE	Collision Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is collision activity on the network. The default value of this bit is 0.

LED1 Control Register

Offset 0E2h

This register controls the function(s) that the $\overline{\text{LED1}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. This register defaults to transmit or receive activity (XMTE = 1 and RCVE = 1) with pulse stretcher enabled (PSE = 1) and is fully programmable.

All bits in this register are restored to their default values when the $\overline{\text{RST}}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

This register is an alias of BCR5.

The functions of the bits in this register are identical to those of the LED0 Control Register, except that for this register the default value for the XMTE, RCVE, and PSE bits is 1, and the default value for all other bits is 0.

LED2 Control Register

Offset 0E4h

This register controls the function(s) that the $\overline{\text{LED2}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. This register defaults to 100 Mb/s speed indication (100E = 1) with pulse stretcher enabled (PSE = 1) and is fully programmable.

All bits in this register are restored to their default values when the $\overline{\text{RST}}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

This register is an alias of BCR6.

The functions of the bits in this register are identical to those of the LED0 Control Register, except that for this register the default value for the 100E and PSE bits is 1, and the default value for all other bits is 0.

LED3 Control Register

Offset 0E6h

This register controls the function(s) that the $\overline{\text{LED3}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. This register defaults to collision indication (COLE = 1) with pulse stretcher enabled (PSE = 1) and is fully programmable.

All bits in this register are restored to their default values when the $\overline{\text{RST}}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

This register is an alias of BCR7.

The functions of the bits in this register are identical to those of the LED0 Control Register, except that for this register the default value for the COLE and PSE bits is 1, and the default value for all other bits is 0.

Downloaded from Arrow.com.

MAX_LAT_A: PCI Maximum Latency Alias Register

Offset 1B1h

This register is a writable alias of the Maximum Latency field at offset 3Fh in PCI configuration space, which is read only. The purpose of this register is to allow the PCI Maximum Latency value to be loaded from the serial EEPROM.

The contents of this register are set to 18h when the $\overline{RST}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 64. MAX_LAT_A: PCI Maximum Latency Alias Register

Bit	Name	Description
7-0	MAX_LAT	Maximum Latency. Specifies the maximum arbitration latency the Am79C976 controller can sustain without causing problems to the network activity. The register value specifies the time in units of 1/4 microseconds. MAX_LAT is aliased to the PCI configuration space register MAX_LAT (offset 3Fh). The host will use the value in the register to determine the setting of the Am79C976 Latency Timer register. The default value for MAX_LAT is 18h which corresponds to 6 µs. This register is an alias of BCR22, bits [15:8] and of offset 3Fh in PCI configuration space.

MIN_GNT_A: PCI Minimum Grant Alias Register

Offset 1B0h

This register is a writable alias of the Minimum Grant field at offset 3Eh in PCI configuration space, which is read only. The purpose of this register is to allow the PCI Minimum Grant value to be loaded from the serial EEPROM.

The contents of this register are set to 18h when the $\overline{RST}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 65. MIN_GNT_A: PCI Minimum Grant Alias Register

Bit	Name	Description
7-0	MIN_GNT	Minimum Grant. Specifies the minimum length of a burst period the Am79C976 controller needs to keep up with the network activity. The length of the burst period is calculated assuming a clock rate of 33 MHz. The register value specifies the time in units of 1/4 µs. MIN_GNT is aliased to the PCI configuration space register MIN_GNT (offset 3Eh). The host will use the value in the register to determine the setting of the Am79C976 Latency Timer register. The default value for MIN_GNT is 18h which corresponds to 6 µs. This register is an alias of BCR22, bits [7:0] and of offset 3Eh in PCI configuration space.

PADR: Physical Address Register

Offset 160h

The contents of this register are cleared to 0 when the $\overline{RST}$ pin is asserted. This register is not cleared by the serial EEPROM read operation or by a serial EEPROM read error.

Downloaded from Arrow.com.

Table 66. PADR: Physical Address Register

Bit	Name	Description
47-0	PADR	MAC Physical Address, PADR[47:0]. This register contains 48-bit, globally unique station address assigned to this device. If the least significant bit of the first byte of a received frame is 0, the destination address of the frame is a unicast address, which will be compared with the contents of the PADR. If this bit is 0 and the frame's destination address exactly matches the contents of PADR, the frame is accepted and copied into the host system memory. The byte order is such that PADR7[7:0] corresponds to the first address byte transferred over the network. Unicast address matching can be disabled by setting the Disable Receive Physical Address bit (DCRVPA, bit 18 in CMD2). If DRCVPA is set to 1, a match of a frame's destination address with the contents of PADR will not cause the frame to be accepted and copied into the host memory. The contents of this register should be loaded from EEPROM. This register can also be loaded from the initialization block after the INIT bit in CSR0 has been set. This register is an alias for CSR12, CSR13, and CSR14.

Pause Count Register

Offset 0DEh

All bits in this register are cleared to 0 when the $\overline{\text{RST}}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error. This register is read-only.

Table 67. PAUSE_CNT: Pause Count Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined
15-0	PAUSE_CNT	Pause Count. This field indicates the pause time parameter that was contained in the request_operand field of the most recently received MAC Control Pause frame.

PCIDATA0: PCI DATA Register Zero Alias Register

Offset 1BCh

This register contains the data that appears in the DATA_SCALE field of the PCI Power Management Control/Status Register (PMCSR) and the PCI Data Register when the DATA_SEL field of PMCSR is set to 0. Since the DATA_SCALE and DATA fields in the configuration space are read only, this register provides a means of programming them indirectly, normally by loading them from the serial EEPROM.

This register is an alias of BCR37.

The contents of this register are cleared to 0 when the $\overline{\text{RST}}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 68. PCIDATA0: PCI DATA Register Zero Alias Register

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D0_SCALE	These bits correspond to the DATA_SCALE field of the PMCSR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field.
7-0	DATA0	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field.

PCIDATA1: PCI DATA Register One Alias Register

Offset 1BEh

This register is identical to the PCI Data Register Zero Alias Register except that it contains the data that appears in the DATA_SCALE field of the PCI Power Management Control/Status Register (PMCSR) and the PCI Data Register when the DATA_SEL field of PMCSR is set to 1.

Downloaded from Arrow.com.

This register is an alias of BCR38.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

PCIDATA2: PCI DATA Register Two Alias Register

Offset 1C0h

This register is identical to the PCI Data Register Zero Alias Register except that it contains the data that appears in the DATA_SCALE field of the PCI Power Management Control/Status Register (PMCSR) and the PCI Data Register when the DATA_SEL field of PMCSR is set to 2.

This register is an alias of BCR39.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

PCIDATA3: PCI DATA Register Three Alias Register

Offset 1C2h

This register is identical to the PCI Data Register Zero Alias Register except that it contains the data that appears in the DATA_SCALE field of the PCI Power Management Control/Status Register (PMCSR) and the PCI Data Register when the DATA_SEL field of PMCSR is set to 3.

This register is an alias of BCR40.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

PCIDATA4: PCI DATA Register Four Alias Register

Offset 1C4h

This register is identical to the PCI Data Register Zero Alias Register except that it contains the data that appears in the DATA_SCALE field of the PCI Power Management Control/Status Register (PMCSR) and the PCI Data Register when the DATA_SEL field of PMCSR is set to 4.

This register is an alias of BCR41.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

PCIDATA5: PCI DATA Register Five Alias Register

Offset 1C6h

This register is identical to the PCI Data Register Zero Alias Register except that it contains the data that appears in the DATA_SCALE field of the PCI Power Management Control/Status Register (PMCSR) and the PCI Data Register when the DATA_SEL field of PMCSR is set to 5.

This register is an alias of BCR42.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

PCIDATA6: PCI DATA Register Six Alias Register

Offset 1C8h

This register is identical to the PCI Data Register Zero Alias Register except that it contains the data that appears in the DATA_SCALE field of the PCI Power Management Control/Status Register (PMCSR) and the PCI Data Register when the DATA_SEL field of PMCSR is set to 6.

This register is an alias of BCR43.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

PCIDATA7: PCI DATA Register Seven Alias Register

Offset 1CAh

This register is identical to the PCI Data Register Zero Alias Register except that it contains the data that appears in the DATA_SCALE field of the PCI Power Management Control/Status Register (PMCSR) and the PCI Data Register when the DATA_SEL field of PMCSR is set to 7.

This register is an alias of BCR44.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Downloaded from Arrow.com.

PHY Access Register

Offset 0D0h

This register gives the host CPU indirect access to the MII Management Bus (MDC/MDIO). Through this register the host CPU can read or write any external PHY register that is accessible through the MII Management Bus.

All bits in this register are cleared to 0 when the $\overline{RST}$ pin is asserted. This register is not affected by the serial EEPROM read operation or by a serial EEPROM read error.

Table 69. PHY_ACCESS: PHY Access Register

Bit	Name	Description
31	PHY_CMD_DONE	PHY Command Complete. This bit is cleared by writing a 1 to it. It is set at the end of the MII Management Frame that is generated by setting any one of PHY_WR_CMD, PHY_BLK_RD_CMD or PHY_NBLK_RD_CMD bits. When the value of this bit is 1 and the previous operation was a read, the PHY_DATA field contains valid data.
30	PHY_WR_CMD	PHY Write Command. When this bit is set, an MII Management Frame will be sent to write the contents of the PHY_DATA field to the external PHY register addressed by the PHY_ADDR and PHY_REG_ADDR fields. This bit must not be set at the same time that the PHY_BLK_RD_CMD bit or the PHY_NBLK_RD_CMD bit is set.
29	PHY_BLK_RD_CMD	PHY Blocking Read Command. When the bit is set, an MII Management Frame will be sent to read the contents of the PHY_DATA field to the external PHY register addressed by the PHY_ADDR and PHY_REG_ADDR fields. After this bit is set, the next attempt to read this register will cause PCI bus retries to occur until the PHY_DATA field has been updated with data read from the selected PHY register. This bit must not be set at the same time that the PHY_WR_CMD bit or the PHY_NBLK_RD_CMD bit is set.
28	PHY_NBLK_RD_CMD	PHY Non-Blocking Read Command. When the bit is set, an MII Management Frame will be sent to read the contents of the PHY_DATA field to the external PHY register addressed by the PHY_ADDR and PHY_REG_ADDR fields. After this bit is set, the host CPU can read this register again and again until the PHY_CMD_DONE bit returns the value 1, indicating that the PHY_DATA field contains valid data read from the selected PHY register. Alternatively, the host CPU can wait for the MCCINT interrupt (INT0, bit 17). This bit must not be set at the same time that the PHY_WR_CMD bit or the PHY_BLK_RD_CMD bit is set.
27	PHY_PRE_SUP	Preamble Suppression. If this bit is set, the MII Management Frame will be sent without a preamble. Before setting this bit the host CPU must make sure that the external PHY addressed by the PHY_ADDR field is capable of accepting MII Management Frames without preambles.
26	RES	Reserved location. Written as zero and read as undefined.
25-21	PHY_ADDR	PHY Address. The address of the external PHY device to be accessed.
20-16	PHY_REG_ADDR	PHY Register Address. The address of the register in the external PHY device to be accessed.
15-0	PHY_DATA	Data written or read from the external PHY register specified by the PHY_ADDR and PHY_REG_ADDR. Data read from this field is valid only following a PHY register read when PHY_CMD_DONE is 1. It is meaningless after a PHY register write.

Downloaded from Arrow.com.

PMAT0: OnNow Pattern Register 0

Offset 190h

This register is used to control and indirectly access the Pattern Match RAM (PMR). When the PMAT_MODE bit (CMD7, bit3) is 1, Pattern Match logic is enabled. No bus accesses into PMR are possible, and PMAT0 and PMAT1 are ignored. When PMAT_MODE is set, a read of PMAT0 or PMAT1 returns all undefined bits.

When the PMAT_MODE bit (CMD7, bit3) is 0, the Pattern Match logic is disabled and accesses to the PMR are possible. Bits 6-0 of PMAT0 specify the address of the PMR word to be accessed. Following the write to PMAT0, the PMR word may be read by reading PMAT1 and the high order bytes of PMAT0 in any order. To write to PMR word, the write to PMAT0 must be followed by a write to PMAT1 to complete the operation. The RAM will not actually be written until the write to PMAT1 is complete. The write to PMAT1 causes all 5 bytes (two bytes of PMAT1 and the upper three bytes of PMAT0) to be written to whatever PMR word is addressed by bits 6:0 of PMAT0.

The contents of this register are cleared to 0 when the $\overline{\text{RST}}$ pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 70. PMAT0: OnNow Pattern Register 0

Bit	Name	Description
31-24	PMR_B2	Pattern Match RAM Byte 2. This byte is written into or read from Byte 2 of the Pattern Match RAM. This field is an alias of BCR46, bits [15:8].
23-16	PMR_B1	Pattern Match RAM Byte 1. This byte is written into or read from Byte 1 of the Pattern Match RAM. This field is an alias of BCR46, bits [7:0].
15-8	PMR_B0	Pattern Match RAM Byte 0. This byte is written into or read from Byte 0 of the Pattern Match RAM. This field is an alias of BCR45, bits [15:8].
7	RES	Reserved location. Written as zero and read as undefined.
6-0	PMR_ADDR	Pattern Match RAM Address. These bits are the Pattern Match RAM address to be written to or read from.

PMAT1: OnNow Pattern Register 1

Offset 194h

This register is used to control and indirectly access the Pattern Match RAM (PMR). When the PMAT_MODE bit (CMD7, bit3) is 1, Pattern Match logic is enabled. No bus accesses into PMR are possible, and PMAT0 and PMAT1 are ignored. When PMAT_MODE is set, a read of PMAT0 or PMAT1 returns all undefined bits.

When the PMAT_MODE bit (CMD7, bit3) is 0, the Pattern Match logic is disabled and accesses to the PMR are possible. Bits 6-0 of PMAT0 specify the address of the PMR word to be accessed. Following the write to PMAT0, the PMR word may be read by reading PMAT1 and the high order bytes of PMAT0 in any order. To write to PMR word, the write to PMAT0 must be followed by a write to PMAT1 to complete the operation. The RAM will not actually be written until the write to PMAT1 is complete. The write to PMAT1 causes all 5 bytes (two bytes of PMAT1 and the upper three bytes of PMAT0) to be written to whatever PMR word is addressed by bits 6:0 of PMAT0.

The contents of this register are cleared to 0 when the $\overline{\text{RST}}$ pin is asserted. The register is not cleared at the start of a serial EEPROM read operation or after a serial EEPROM read error.

Table 71. PMAT1: OnNow Pattern Register 1

Bit	Name	Description
15-8	PMR_B4	Pattern Match RAM Byte 4. This byte is written into or read from Byte 4 of the Pattern Match RAM. This field is an alias of BCR47, bits [15:8].
7-0	PMR_B3	Pattern Match RAM Byte 3. This byte is written into or read from Byte 3 of the Pattern Match RAM. This field is an alias of BCR47, bits [7:0].

Downloaded from Arrow.com.

PMC_A: PCI Power Management Capabilities Alias Register

Offset 1B8h

This register is an alias of the PMC register located at offset 42h of the PCI Configuration Space. Since the PMC register is read only, this register provides a means of programming it through the EEPROM. For the definition of the bits in this register, refer to the PMC register definition.

The contents of this register are set to the default value of 0C802h when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Receive Protect Register

Offset 0DCh

The contents of this register are set to the default value 64 when the RST pin is asserted. This register is not affected by the serial EEPROM read operation or by a serial EEPROM read error.

Table 72. Receive Protect Register

Bit	Name	Description
15-0	RCV_ PROTECT	Receive Protect. This register indicates the number of bytes of an incoming frame that must be received before the DMA controller starts to copy the frame data into the host system memory. If the size of the frame (in bytes) is less than the contents of this register and the frame contains a valid FCS, the DMA transfer can start any time after the end of the frame is received. The Receive Protect Register also determines the period during which the External Address Reject (EAR) pin is monitored when the External Address Detection Interface (EADI) is used. The state of the EAR pin is ignored except for a period of time that starts when the Start of Frame Delimiter of an incoming frame is received and ends when the number of frame data bytes indicated by the Receive Protect Register have been received.

RCV_RING_LEN: Receive Ring Length Register

Offset 150h

The contents of this register are set to the default value 0 when the RST pin is asserted. This register is not affected by the serial EEPROM read operation or by a serial EEPROM read error.

Table 73. RCV_RING_LEN: Receive Ring Length Register

Bit	Name	Description
15-0	RCV_RING_ LEN	Receive Ring Length. Contains the two's complement of the receive descriptor ring length. This register is initialized during the optional Am79C976 controller initialization routine based on the value in the RLEN field of the initialization block. However, this register can be manually altered. The actual receive ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. This register is an alias of CSR76.

Downloaded from Arrow.com.

ROM_CFG: ROM Base Address Configuration Register

Offset 18Eh

This register, which should normally be loaded from the serial EEPROM, determines which bits in the Expansion ROM Base Address Register (ROMBASE) in PCI Configuration Space can be altered by PCI Configuration Space write accesses. Therefore this register indirectly determines the amount of PCI memory space that the Am79C976 device will claim for the PCI Expansion ROM.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 74. ROM_CFG: ROM Base Address Configuration Register

Bit	Name	Description
15-3	ROMBASE [23:11]	This field contains write enable bits for bits [23:11] of the Expansion ROM Base Address Register (ROMBASE) in PCI Configuration Space. If a bit in this field is set to 1, the corresponding bit in the ROMBASE register can be written by PCI Configuration Space accesses. If a bit in this field is cleared to 0, the corresponding bit in the ROMBASE register will be fixed at 0, and can not be altered by PCI Configuration Space accesses. Bits 15-3 of this field correspond to bits 23-11 of the ROMBASE register.
2-1	RES	Reserved locations. Written as zeros; read as undefined.
0	ROMBASE[0]	This bit is the write enable bit for bit [0] of the Expansion ROM Base Address Register (ROMBASE) in PCI Configuration Space. If this bit is set to 1, the Expansion ROM Enable bit in the ROMBASE register can be written by PCI Configuration Space accesses. If a bit in this field is cleared to 0, the Expansion ROM Enable bit in the ROMBASE register will be fixed at 0, and can not be altered by PCI Configuration Space accesses. This bit should be set to 1, normally by the EEPROM read operation, if an expansion ROM is present in the system. It should be cleared to 0 (the default state) if there is no expansion ROM in the system.

SID_A: PCI Subsystem ID Alias Register

Offset 1B4h

This register is a writable alias of the Subsystem ID field at offset 2Eh in PCI configuration space, which is read only. The purpose of this register is to allow the PCI Subsystem Vendor ID value to be loaded from the serial EEPROM.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 75. SID_A: PCI Subsystem ID Alias Register

Bit	Name	Description
15-0	SID	Subsystem ID. SID is used together with SVID (BCR23, bits 15-0) to uniquely identify the add-in board or subsystem the Am79C976 controller is used in. The value of SID is determined by the system vendor. A value of 0 (the default) indicates that the Am79C976 controller does not support subsystem identification. This register is an alias of BCR24 and of the PCI configuration space Subsystem ID field at offset 2Eh.

Downloaded from Arrow.com.

SRAM Boundary Register

Offset 17Ah

All bits in this register are cleared to 0 when the $\overline{RST}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 76. SRAM Boundary Register

Bit	Name	Description
15-0	SRAM_BND	SRAM Boundary. Specifies the size of the transmit buffer portion of the SRAM in units of 512-byte pages. For example, if SRAM_BND is set to 10, then 5120 bytes of the SRAM will be allocated for the transmit buffer and the rest will be allocated for the receive buffer. The transmit buffer in the SRAM begins at address 0 and ends at the address (SRAM_BND*512)-1. Therefore, the receive buffer always begins on a 512-byte boundary. SRAM_BND must be initialized to an appropriate value, either by the EEPROM or by the host CPU. SRAM_BND must be set to a value less than or equal to IFFCh. Values larger than IFFCh will cause incorrect behavior. ***Note**: The minimum allowed number of pages for normal network operation is four, and the maximum is SRAM_SIZE - 4.* This register is an alias of BCR26.

SRAM Size Register

Offset 178h

All bits in this register are cleared to 0 when the $\overline{RST}$ pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 77. SRAM Size Register

Bit	Name	Description
15-0	SRAM_SIZE	SRAM Size. Specifies the total size of the SSRAM buffer in units of 512-byte pages. For example, assume that the external memory consists of one 64K X 32 bit SSRAM, for a total of 256K bytes. In this case SRAM_SIZE should be set to 512 (256K divided by 512). This field must be initialized to the appropriate value, either by the EEPROM or by the host CPU. SRAM_SIZE must be set to a value less than or equal to 2000h. Values larger than 2000h will cause incorrect behavior. ***Note**: The minimum allowed number of pages is eight for normal network operation.* This register is an alias of BCR25.

Downloaded from Arrow.com.

STAT0: Status0

Offset 030h

STAT0 indicates the status of various Am79C976 functions. All bits in this register except for bits 12:10 indicate current status and are read only. Bits 12:10 are latches that indicate the cause of a wake-up event. These three bits are cleared to 0 when power is first applied to the device (power-on reset), but they are not affected by the state of the $\overline{RST}$ pin so that they are not disturbed when PCI bus power is removed and reapplied. Bits 12:10 are “write 1 to clear”. Therefore, the CPU can clear bits 12:10 by reading the register and then writing back the same data that it read.

Table 78. STAT0: Status0 Register

Bit	Name	Description
31-15	RES	Reserved locations. Written as zeros and read as undefined.
14	PAUSE_PEND	Pause Pending. This bit is set to 1 during the interval between the time that the Am79C976 device receives a command to transmit a MAC Control Pause frame and the time that the device finishes transmitting the frame. The host CPU should not attempt to write to the Pause Length Register while this bit is 1. This bit is read only and is cleared by H_RESET.
13	PAUSING	Pausing. This bit indicates that the device has received a MAC Control Pause frame, and the pause timer has not yet timed out. This bit is read only and is cleared by H_RESET.
12	PMAT_DET	Pattern Match Detected. This bit indicates that an OnNow pattern match has occurred while the Am79C976 device was in the OnNow pattern match mode. This bit is an alias of PMAT in CSR116. This bit can be cleared to 0 either by writing 0 to CSR116, bit 7, or by writing 1 to STAT0, bit 12. This bit is cleared to 0 when power is first applied to the device, but not by the assertion of $\overline{RST}$.
11	MP_DET	Magic Packet Frame Detected. This bit indicates that a Magic Packet pattern match has occurred while the Am79C976 device was in the Magic Packet mode. This bit is an alias of MPMAT in CSR116. This bit can be cleared to 0 either by writing 0 to CSR116, bit 5, or by writing 1 to STAT0, bit 11. This bit is cleared to 0 when power is first applied to the device, but not by the assertion of $\overline{RST}$.
10	LC_DET	Link Change Detected. This bit indicates that a change in the link status of the external PHY device has been detected while the device was in the Link Change Wake-up mode. This bit is an alias of LCDET in CSR116. This bit can be cleared to 0 either by writing 1 to CSR116, bit 9, or by writing 1 to STAT0, bit 10. This bit is cleared to 0 when power is first applied to the device, but not by the assertion of $\overline{RST}$.
9-7	SPEED	Speed. This field indicates the bit rate at which the network is running. The following encoding is used: 000 Unknown 001 Reserved 010 10 Mb/s 011 100 Mb/s 100-111 Reserved These bits are read only and are cleared by H_RESET.
6	FULL_DPLX	Full Duplex. This bit is set when the device is operating in full-duplex mode. This bit is read only and is cleared by H_RESET.

Downloaded from Arrow.com.

Bit	Name	Description
5	LINK_STAT	Link Status. This bit is set to the value of the Link Status bit in the status register (R1) of the default external PHY. (The default external PHY is the PHY addressed by the AP_PHY0_ADDR field of the AUTOPOLL0 Register.) This bit is updated each time the external PHY's status register is read, either by the Auto-Poll State Machine, by the Network Port Manager, or by a CPU-initiated read. However, when the Force Link Status bit in CMD3 is set to 1, this bit is forced to 1, regardless of the contents of the external PHY's status register. This bit is read only and is cleared by H_RESET.
4	AUTONEG_ COMPLETE	Auto-negotiation Complete. This bit is set to the value of the Auto-Negotiation Complete bit in register 1 of the external PHY as determined by the most recent Port Manager polling cycle. This bit is read only and is cleared by H_RESET.
3	MIIPD	MII PHY Detect. MIIPD reflects the quiescent state of the MDIO pin. MIIPD is continuously updated whenever there is no management operation in progress on the MII interface. When a management operation begins on the interface, the state of MIIPD is preserved until the operation ends, when the quiescent state is again monitored and continuously updates the MIIPD bit. When the MDIO pin is at a quiescent LOW state, MIIPD is cleared to 0. When the MDIO pin is at a quiescent HIGH state, MIIPD is set to 1. Any transition on the MIIPD bit will set the MIIPDTINT bit in INT0. This bit is an alias of BCR32, bit 14. This bit is read only and is cleared by H_RESET.
2	RX_ SUSPENDED	Receiver Suspended. This bit has the value 1 while the receiver is suspended, and it has the value 0 when the receiver is not suspended. This bit is read only and is cleared by H_RESET.
1	TX_ SUSPENDED	Transmitter Suspended. This bit has the value 1 while the transmitter is suspended, and it has the value 0 when the transmitter is not suspended. This bit is read only and is cleared by H_RESET.
0	RUNNING	This bit is a read-only alias of the RUN bit in CMD0. When this bit is set, the device is enabled to transmit and receive frames and process descriptors. Note that even though RUNNING is set, the receiver or transmitter might be disabled because RX_SPND or TX_SPND is set (in CMD0). This bit is read only and is cleared by H_RESET.

Software Timer Value Register

Offset 0D8h

The contents of this register are set to the default value (0FFFFh) when the $\overline{\text{RST}}$ pin is asserted. This register is not affected by the serial EEPROM read operation or by a serial EEPROM read error.

Table 79. Software Timer Value Register

Bit	Name	Description
15-0	STVAL	Software Timer Value. STVAL controls the maximum time for the Software Timer to count before generating the STINT (CSR7, bit 11) interrupt. The Software Timer is a free-running timer that is started upon the first write to STVAL. After the first write, the Software Timer will continually count and set the STINT interrupt at the STVAL period. The STVAL value is interpreted as an unsigned number with a resolution of 10.24µs. For instance, if STVAL is set to 48,828 (0BEBCh), the Software Timer period will be 0.5 s. The default value (0FFFFh) corresponds to 0.6710784 s. Setting STVAL to a value of 0 will result in erratic behavior. This register is an alias of BCR31.

Downloaded from Arrow.com.

SVID_A: PCI Subsystem Vendor ID Alias Register

Offset 1B6h

This register is a writable alias of the Subsystem Vendor ID field at offset 2Ch in PCI configuration space, which is read only. The purpose of this register is to allow the PCI Subsystem Vendor ID value to be loaded from the serial EEPROM.

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 80. SVID: PCI Subsystem Vendor ID Shadow Register

Bit	Name	Description
15-0	SVID	Subsystem Vendor ID. SVID is used together with the PCI Subsystem ID Register to uniquely identify the add-in board or subsystem the Am79C976 controller is used in. Subsystem Vendor IDs can be obtained form the PCI SIG. A value of 0 (the default) indicates that the Am79C976 controller does not support subsystem identification. SVID is aliased to the PCI configuration space register Subsystem Vendor ID (offset 2Ch). This register is an alias of BCR23 and of the PCI configuration space Subsystem Vendor ID field at offset 2Ch.

TEST0: Test Register

Offset 1A8h

This register is used for factory test purposes only. All bits must be zero for normal operation.

Table 81. TEST0: Test Register

Bit	Name	Description
31:18	RES	Factory test bits; must be 0 for normal operation.
17	CBIO_EN	Factory test bit; must be 0 for normal operation.
16:14	RES	Factory test bits; must be 0 for normal operation.
13	EEBUSY_T	Factory test bit; must be 0 for normal operation.
12	RES	Factory test bit; must be 0 for normal operation.
11	TSEL	Factory test bit; must be 0 for normal operation.
10	MFSM_RESET	Factory test bit; must be 0 for normal operation.
9	BFD_SCALE_DOWN	Factory test bit; must be 0 for normal operation.
8	ANTST	Factory test bit; must be 0 for normal operation.
7:6	RES	Factory test bits; must be 0 for normal operation.
5	LEDCHTTST	Factory test bit; must be 0 for normal operation.
4	RTYTST_BUMP	Factory test bit; must be 0 for normal operation.
3	RTYTST_OUT	Factory test bit; must be 0 for normal operation.
2	RTYTST_RANGEN	Factory test bit; must be 0 for normal operation.
1	RTYTST_SLOT	Factory test bit; must be 0 for normal operation.
0	SERRLEVEL	Factory test bit; must be 0 for normal operation.

Downloaded from Arrow.com.

VID_A: PCI Vendor ID Alias Register

Offset 1B2h

This register is a writable alias of the Vendor ID field at offset 0 in PCI configuration space, which is read only. The purpose of this register is to allow the PCI Vendor ID value to be loaded from the serial EEPROM.

The contents of this register are set to 1022h when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Table 82. VID_A: PCI Vendor ID Alias Register

Bit	Name	Description
15-0	VID	Vendor ID. The PCI Vendor ID register is a 16-bit register that identifies the manufacturer of the Am79C976 controller. AMD's Vendor ID is 1022h. Note that this Vendor ID is not the same as the Manufacturer ID in CSR88 and CSR89. The Vendor ID is assigned by the PCI Special Interest Group. The Vendor ID is not normally programmable, but the Am79C976 controller allows this due to legacy operating systems that do not look at the PCI Subsystem Vendor ID and the Vendor ID to uniquely identify the add-in board or subsystem that the Am79C976 controller is used in. ***Note: If the operating system or the network operating system supports PCI Subsystem Vendor ID and Subsystem ID, use those to identify the add-in board or subsystem and program the VID with the default value of 1022h.*** Software supplied by AMD may not operate if the VID is anything other than 1022h. This register is an alias of the PCI configuration space Vendor ID field at offset 00h and of BCR35.

XMT_RING_LEN: Transmit Ring Length Register

Offset 140h

The contents of this register are set to the default value 0 when the RST pin is asserted. This register is not affected by the serial EEPROM read operation or by a serial EEPROM read error.

Table 83. XMT_RING_LEN: Transmit Ring Length Register

Bit	Name	Description
15-0	XMT_RING_LEN	Transmit Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the optional Am79C976 controller initialization routine based on the value in the TLEN field of the initialization block. However, this register can be manually altered. The actual transmit ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. This register is an alias of CSR78.

XMTPOLLTIME: Transmit Poll Timer Register

Offset 188h

The contents of this register are cleared to 0 when the RST pin is asserted, before the serial EEPROM is read, and after a serial EEPROM read error.

Downloaded from Arrow.com.

Table 84. XMTPOLLTIME: Transmit Polling Interval Register

Bit	Name	Description
15-0	XMTPOLLTIME	Transmit Polling Interval. This register contains the time that the Am79C976 controller will wait between successive polling operations. The XMTPOLLTIME value is expressed as the two's complement of the desired interval, where each bit of XMTPOLLTIME represents ERCLK clock periods. XMTPOLLTIME[3:0] are ignored. (XMTPOLLTIME[16] is implied to be a one, so XMTPOLLTIME[15] is significant and does not represent the sign of the two's complement XMTPOLLTIME value.) The default value of this register is 0000h. This corresponds to a polling interval of 65,536 clock periods (2.185 ms when ERCLK = 90 MHz). Setting the INIT bit starts an initialization process that sets XMTPOLLTIME to its default value. If the user wants to program a value for XMTPOLLTIME other than the default, then he must change the value after the initialization sequence has completed. This register is an alias for CSR47.

RAP Register

The RAP (Register Address Pointer) register is used to gain access to CSR and BCR registers on board the Am79C976 controller. The RAP contains the address of a CSR or BCR.

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

RAP: Register Address Port

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	RES	Reserved locations. Read and written as zeros.
7-0	RAP	Register Address Port. The value of these 8 bits determines which CSR or BCR will be accessed when an I/O access to the RDP or BDP port, respectively, is performed. A write access to undefined CSR or BCR locations may cause unexpected reprogramming of the Am79C976 control registers. A read access will yield undefined values. Read/Write accessible. RAP is cleared by H_RESET or S_RESET and is unaffected by setting the STOP bit.

Downloaded from Arrow.com.

Control and Status Registers

The Control and Status Registers (CSRs) are included for compatibility with older PCnet Family software. All CSR functions can be accessed more efficiently through the memory-mapped registers.

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

Am79C976 CSRs can be accessed at any time. For older PCnet family devices certain CSRs could only be accessed when the device is stopped.

CSR0: Am79C976 Controller Status and Control Register

Certain bits in CSR0 indicate the cause of an interrupt. The register is designed so that these indicator bits are cleared by writing ones to those bit locations. This means that the software can read CSR0 and write back the value just read to clear the interrupt condition.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	ERR	Obsolete function. Read/Write accessible. Read returns zero.
14	BABL	Obsolete function. Read/Write accessible. Read returns zero.
13	CERR	Obsolete function. Read/Write accessible. Read returns zero.
12	MISS	Obsolete function. Read/Write accessible. Read returns zero.
11	MERR	Obsolete function. Read/Write accessible. Read returns zero.
10	RINT	Receive Interrupt is set by the Am79C976 controller after the last descriptor of a receive frame has been updated by writing a 0 to the OWNership bit. RINT may also be set when the first descriptor of a receive frame has been updated by writing a 0 to the OWNership bit if the LAPPEN bit of CSR3 has been set to a 1. When RINT is set, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit RINTM (CSR3, bit 10) is 0. Read/Write accessible. RINT is cleared by the host by writing a 1. Writing a 0 has no effect. RINT is cleared by H_RESET, S_RESET, or by setting the STOP bit.
9	TINT	Transmit Interrupt is set by the Am79C976 controller after the OWN bit in the last descriptor of a transmit frame has been cleared to indicate the frame has been copied to the transmit FIFO. When TINT is set, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit TINTM (CSR3, bit 9) is 0. Read/Write accessible. TINT is cleared by the host by writing a 1. Writing a 0 has no effect. TINT is cleared by H_RESET, S_RESET, or by setting the STOP bit.
8	IDON	Initialization Done is set by the Am79C976 controller after the initialization sequence has completed. When IDON is set, the Am79C976 controller has read the initialization block from memory. When IDON is set, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit IDONM (CSR3, bit 8) is 0. Read/Write accessible. IDON is cleared by the host by writing a 1. Writing a 0 has no effect. IDON is cleared by H_RESET, S_RESET, or by setting the STOP bit.
7	INTR	Interrupt Flag indicates that one or more following interrupt causing conditions has occurred: IDON, RINT, SINT, TINT, TXSTRT, UINT, STINT, MREINT, MCCINT, MCCIINT, MIIPDTINT, MAPINT, MPINT, APINT, LCINT, SPNDINT and the associated mask or enable bit is programmed to allow the event to cause an interrupt. If IENA is set to 1 and INTR is set, $\overline{\text{INTA}}$ will be active. When INTR is set by SINT or SLPINT, $\overline{\text{INTA}}$ will be active independent of the state of IENA.

Downloaded from Arrow.com.

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

6 IENA — Interrupt Enable allows $\overline{\text{INTA}}$ to be active if the Interrupt Flag is set. If IENA = 0, then $\overline{\text{INTA}}$ will be disabled regardless of the state of INTR.

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

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

4 TXON — Transmit On indicates that the transmit function is enabled. TXON is set if DTX (CSR15, bit 1) is set to 0 after the STRT bit is set. If INIT and STRT are set together, TXON will not be set until after the initialization block has been read in.

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

TXON is read only. TXON is cleared by H_RESET or S_RESET and setting the STOP bit.

3 TDMD — Transmit Demand, when set, causes the Buffer Management Unit to access the Transmit Descriptor Ring without waiting for the poll-time counter to elapse. If TXON is not enabled, TDMD bit will be reset and no Transmit Descriptor Ring access will occur.

TDMD is required to be set if the TXDPOLL bit in CSR4 is set. Setting TDMD while TXDPOLL = 0 merely hastens the Am79C976 controller's next access to a Transmit Descriptor Ring Entry.

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

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

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

1 STRT — STRT assertion enables Am79C976 controller to send and receive frames, and perform buffer management operations. Setting STRT clears the STOP bit. If STRT and INIT are set together, the Am79C976 controller initialization will be performed first.

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

0 INIT — INIT assertion enables the Am79C976 controller to begin the initialization procedure which reads in the initialization block from memory. Setting INIT clears the STOP bit. If STRT and INIT are set together, the Am79C976 controller initialization will be performed first. INIT is not cleared when the initialization sequence has completed.

Read/Write accessible. INIT is set by writing a 1. Writing a 0 has

Downloaded from Arrow.com.

no effect. INIT is cleared by H_RESET, S_RESET, or by setting the STOP bit.

CSR1: Initialization Block Address 0

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

CSR2: Initialization Block Address 1

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	IADR[31:24]	If SSIZE32 is set (BCR20, bit 8), then the IADR[31:24] bits will be used strictly as the upper 8 bits of the initialization block address. However, if SSIZE32 is reset (BCR20, bit 8), then the IADR[31:24] bits will be used to generate the upper 8 bits of all bus mastering addresses, as required for a 32-bit address bus. Note that the 16-bit software structures specified by the SSIZE32 = 0 setting will yield only 24 bits of address for the Am79C976 bus master accesses, while the 32-bit hardware for which the Am79C976 controller is intended will require 32 bits of address. Therefore, whenever SSIZE32 = 0, the IADR[31:24] bits will be appended to the 24-bit initialization address, to each 24-bit descriptor base address and to each beginning 24-bit buffer address in order to form complete 32-bit addresses. The upper 8 bits that exist in the descriptor address registers and the buffer address registers which are stored on board the Am79C976 controller will be overwritten with the IADR[31:24] value, so that CSR accesses to these registers will show the 32-bit address that includes the appended field. If SSIZE32 = 1, then software will provide 32-bit pointer values for all of the shared software structures - i.e., descriptor bases and buffer addresses, and therefore, IADR[31:24] will not be written to the upper 8 bits of any of these resources, but it will be used as the upper 8 bits of the initialization address. This register is aliased with CSR17. Read/Write accessible. Unaffected by H_RESET, S_RESET, or by setting the STOP bit.
7-0	IADR[23:16]	Bits 23 through 16 of the address of the Initialization Block. Whenever this register is written, CSR17 is updated with CSR2's contents. Read/Write accessible Unaffected by H_RESET, S_RESET, or by setting the STOP bit.

CSR3: Interrupt Masks and Deferral Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-13	RES	Reserved locations. Read and written as zero.
12	MISSM	Obsolete function. Writing has no effect. Read as undefined.
11	MERRM	Obsolete function. Writing has no effect. Read as undefined.
10	RINTM	Receive Interrupt Mask. If RINTM is set, the RINT bit will be masked and unable to set the INTR bit. Read/Write accessible. RINTM is set by H_RESET but cleared by S_RESET and is not affected by STOP.

Downloaded from Arrow.com.

9	TINTM	Transmit Interrupt Mask. If TINTM is set, the TINT bit will be masked and unable to set the INTR bit. Read/Write accessible. TINTM is set by H_RESET but cleared by S_RESET and is not affected by STOP.
8	IDONM	Initialization Done Mask. If IDONM is set, the IDON bit will be masked and unable to set the INTR bit. Read/Write accessible. IDONM is cleared by H_RESET or S_RESET and is not affected by STOP.
7	RES	Reserved location. Read and written as zeros.
6	DXSUFLO	Obsolete function. Writing has no effect. Read as undefined.
5	LAPPEN	Look Ahead Packet Processing Enable. When set to a 1, the LAPPEN bit will cause the Am79C976 controller to generate an interrupt following the descriptor write operation to the first buffer of a receive frame. This interrupt will be generated in addition to the interrupt that is generated following the descriptor write operation to the last buffer of a receive packet. The interrupt will be signaled through the RINT bit of CSR0. Setting LAPPEN to a 1 also enables the Am79C976 controller to read the STP bit of receive descriptors. The Am79C976 controller will use the STP information to determine where it should begin writing a receive packet's data. Note that while in this mode, the Am79C976 controller can write intermediate packet data to buffers whose descriptors do not contain STP bits set to 1. Following the write to the last descriptor used by a packet, the Am79C976 controller will scan through the next descriptor entries to locate the next STP bit that is set to a 1. The Am79C976 controller will begin writing the next packets data to the buffer pointed to by that descriptor. Note that because several descriptors may be allocated by the host for each packet, and not all messages may need all of the descriptors that are allocated between descriptors that contain STP = 1, then some descriptors/buffers may be skipped in the ring. While performing the search for the next STP bit that is set to 1, the Am79C976 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 Am79C976 controller ownership of the descriptor but also indicate STP = 0, then the Am79C976 controller will reset the OWN bit to 0 in these entries. If a scanned entry indicates host ownership with STP = 0, then the Am79C976 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 Am79C976 controller, then the Am79C976 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 Am79C976 controller, then the Am79C976 controller will stop advancing through the ring entries, store the descriptor information that it has just read, and wait for the next receive to arrive. This behavior allows the host software to pre-assign buffer space in such a manner that the *header* portion of a receive packet will always be written to a particular memory area, and the *data* portion of a receive packet will always be written to a separate memory area. The interrupt is

Downloaded from Arrow.com.

generated when the *header* bytes have been written to the *header* memory area.

Read/Write accessible. The LAPPEN bit will be reset to 0 by H_RESET or S_RESET and will be unaffected by STOP.

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

4 DXMT2PD Disable Transmit Two Part Deferral (see Medium Allocation section in the *Media Access Management* section for more details). If DXMT2PD is set, Transmit Two Part Deferral will be disabled.

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

3 EMBA Enable Modified Back-off Algorithm (see Contention Resolution section in *Media Access Management* section for more details). If EMBA is set, a modified back-off algorithm is implemented.

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

2 BSWP Byte Swap. This bit is used to choose between big and little Endian modes of operation. When BSWP is set to a 1, big Endian mode is selected. When BSWP is set to 0, little Endian mode is selected.

When big Endian mode is selected, the Am79C976 controller will swap the order of bytes on the AD bus during a data phase on accesses to the FIFOs only. Specifically, AD[31:24] becomes Byte 0, AD[23:16] becomes Byte 1, AD[15:8] becomes Byte 2, and AD[7:0] becomes Byte 3 when big Endian mode is selected. When little Endian mode is selected, the order of bytes on the AD bus during a data phase is: AD[31:24] is Byte 3, AD[23:16] is Byte 2, AD[15:8] is Byte 1, and AD[7:0] is Byte 0.

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

Note that the byte ordering of the PCI bus is defined to be little Endian. BSWP should not be set to 1 when the Am79C976 controller is used in a PCI bus application.

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

1-0 RES Reserved locations. The values written to these bits have no effect on the operation of the device. These bits should be read as undefined.

CSR4: Test and Features Control

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	RES	Reserved location. The value written to this bit has no effect on the operation of the device. This bit should be read as undefined.
14	DMAPLUS	Writing and reading from this bit has no effect. DMAPLUS is always 0.

Downloaded from Arrow.com.

13	RES	Reserved Location. Written as zero and read as undefined.
12	TXDPOLL	Disable Transmit Polling. If TXDPOLL is set, the Buffer Management Unit will disable transmit polling. Likewise, if TXDPOLL is cleared, automatic transmit polling is enabled. If TXDPOLL is set, TDMD bit in CSR0 must be set in order to initiate a manual poll of a transmit descriptor. Transmit descriptor polling will not take place if TXON is reset. Transmit polling will take place following Receive activities. Read/Write accessible. TXDPOLL is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.
11	APAD_XMT	Auto Pad Transmit. When set, APAD_XMT enables the automatic padding feature. Transmit frames will be padded to extend them to 64 bytes including FCS. The FCS is calculated for the entire frame, including pad, and appended after the pad field. When the auto padding logic modifies a frame, a valid FCS field will be appended to the frame, regardless of the state of the DXMTFCS bit (CSR15, bit 3) and of the ADD_FCS bit in the transmit descriptor. Read/Write accessible. APAD_XMT is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.
10	ASTRP_RCV	Auto Strip Receive. When set, ASTRP_RCV enables the automatic pad stripping feature. The pad and FCS fields will be stripped from receive frames and not placed in the FIFO. Read/Write accessible. ASTRP_RCV is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.
9	MFCO	Obsolete function. Writing has no effect. Read as undefined.
8	MFCOM	Obsolete function. Writing has no effect. Read as undefined.
7	UINTCMD	User Interrupt Command. UINTCMD can be used by the host to generate an interrupt un related to any network activity. Writing a 1 to this bit causes UINT to be set to 1, which in turn causes $\overline{\text{INTA}}$ to be asserted if interrupts are enabled. Read/Write accessible. UINTCMD is always read as 0.
6	UINT	User Interrupt. UINT is set by the Am79C976 controller after the host has issued a user interrupt command by setting UINTCMD (CSR4, bit 7) to 1. Read/Write accessible. UINT is cleared by the host by writing a 1. Writing a 0 has no effect. UINT is cleared by H_RESET or S_RESET or by setting the STOP bit.
5	RCVCCO	Obsolete function. Writing has no effect. Read as undefined.
4	RCVCCOM	Obsolete function. Writing has no effect. Read as undefined.
3	TXSTRT	Transmit Start status is set by the Am79C976 controller whenever it begins transmission of a frame. When TXSTRT is set, $\overline{\text{INTA}}$ is asserted if IENA is 1 and the mask bit TXSTRTM is 0. Read/Write accessible. TXSTRT is cleared by the host by writing a 1. Writing a 0 has no effect. TXSTRT is cleared by H_RESET, S_RESET, or by setting the STOP bit.
2	TXSTRTM	Transmit Start Mask. If TXSTRTM is set, the TXSTRT bit will be masked and unable to set the INTR bit. Read/Write accessible. TXSTRTM is set to 1 by H_RESET or S_RESET and is not affected by the STOP bit.

Downloaded from Arrow.com.

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

CSR5: Extended Control and Interrupt 1

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	TOKINTD	Obsolete function. Writing has no effect. Read as undefined.
14	LTINTEN	Last Transmit Interrupt Enable. When this bit is set to 1, the LTINT bit in transmit descriptors can be used to determine when transmit interrupts occur. The Transmit Interrupt (TINT) bit will be set after a frame has been copied to the Transmit FIFO if the LTINT bit in the frame's last transmit descriptor is set. If the LTINT bit in the frame's last descriptor is 0 TINT will not be set after the frame has been copied to the Transmit FIFO. Read/Write accessible. LTINTEN is cleared by H_RESET or S_RESET and is unaffected by STOP.
13	TXDNINT	Transmission Done Interrupt. This bit is set when the transmitter has finished sending a frame. This bit is included for debugging purposes. Read/Write accessible. TXDNINT is cleared by the host by writing a 1. Writing a 0 has no effect. The state of TXDNINT is not affected by clearing any of the PCI Status register bits that get set when a data parity error (DATAPERR, bit 8), master abort (RMABORT, bit 13), or target abort (RTABORT, bit 12) occurs. TXDNINT is cleared by H_RESET, S_RESET, or by setting the STOP bit.
12	TXDNINTEN	Transmission Done Interrupt Enable. When this bit is set, the INTR bit will be set when the TXDNINT bit in INT0 is set. Read/Write accessible. TXDNINTEN is cleared by H_RESET or S_RESET and is unaffected by STOP.
11	SINT	System Interrupt is set by the Am79C976 controller when it detects a system error during a bus master transfer on the PCI bus. System errors are data parity error, master abort, or a target abort. The setting of SINT due to data parity error is not dependent on the setting of PERREN (PCI Command register, bit 6). When SINT is set, $\overline{\text{INTA}}$ is asserted if the enable bit SINTE is 1. Note that the assertion of an interrupt due to SINT is not dependent on the state of the INEA bit, since INEA is cleared by the STOP reset generated by the system error. Read/Write accessible. SINT is cleared by the host by writing a 1. Writing a 0 has no effect. The state of SINT is not affected by clearing any of the PCI Status register bits that get set when a data parity error (DATAPERR, bit 8), master abort (RMABORT, bit 13), or target abort (RTABORT, bit 12) occurs. SINT is cleared by H_RESET or S_RESET and is not affected by setting the STOP bit.
10	SINTE	System Interrupt Enable. If SINTE is set, the SINT bit will be able to set the INTR bit. Read/Write accessible. SINTE is set to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.
9-8	RES	Reserved locations. Written as zeros and read as undefined.
7	EXDINT	Obsolete function. Writing has no effect. Read as undefined.

Downloaded from Arrow.com.

6	EXDINTE	Obsolete function. Writing has no effect. Read as undefined.
5	MPPLBA	Magic Packet Physical Logical Broadcast Accept. If MPPLBA is at its default value of 0, the Am79C976 controller will only detect a Magic Packet frame if the destination address of the packet matches the content of the physical address register (PADR). If MPPLBA is set to 1, the destination address of the Magic Packet frame can be unicast, multicast, or broadcast. Note that the setting of MPPLBA only affects the address detection of the Magic Packet frame. The Magic Packet frame's data sequence must be made up of 16 consecutive physical addresses (PADR[47:0]) regardless of what kind of destination address it has. This bit is OR'ed with EMPPLBA bit (CSR116, bit 6). Read/Write accessible. MPPLBA is set to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.
4	MPINT	Magic Packet Interrupt. Magic Packet Interrupt is set by the Am79C976 controller when the device is in the Magic Packet mode and the Am79C976 controller receives a Magic Packet frame. When MPINT is set to 1, $\overline{\text{INTA}}$ is asserted if IENA (CSR0, bit 6) and the enable bit MPINTE are set to 1. Read/Write accessible. MPINT is cleared by the host by writing a 1. Writing a 0 has no affect. MPINT is cleared by H_RESET, S_RESET, or by setting the STOP bit.
3	MPINTE	Magic Packet Interrupt Enable. If MPINTE is set to 1, the MPINT bit will be able to set the INTR bit. Read/Write accessible. MPINTE is cleared to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.
2	MPEN	Magic Packet Enable. MPEN allows activation of the Magic Packet mode by the host. The Am79C976 controller will enter the Magic Packet mode when both MPEN and MPMODE are set to 1. Read/Write accessible. MPEN is cleared to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.
1	MPMODE	The Am79C976 controller will enter the Magic Packet mode when MPMODE is set to 1 and either PG is asserted or MPEN is set to 1. Read/Write accessible. MPMODE is cleared to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.
0	SPND	Suspend. Setting SPND to 1 will cause the Am79C976 controller to start requesting entrance into suspend mode. The host must poll SPND until it reads back 1 to determine that the Am79C976 controller has entered the suspend mode. Setting SPND to 0 will get the Am79C976 controller out of suspend mode. SPND can only be set to 1 if STOP (CSR0, bit 2) is set to 0. H_RESET, S_RESET or setting the STOP bit will get the Am79C976 controller out of suspend mode. Requesting entrance into the suspend mode by the host depends on the setting of the FASTSPNDE bit (CSR7, bit 15). Refer to the bit description of the FASTSPNDE bit and the Suspend section in *Detailed Functions, Buffer Management Unit* for details. In suspend mode, all of the CSR and BCR registers are accessible. As long as the Am79C976 controller is not reset while in suspend mode (by H_RESET, S_RESET or by setting the STOP bit), no re-initialization of the device is required after the device

Downloaded from Arrow.com.

comes out of suspend mode. The Am79C976 controller will continue at the transmit and receive descriptor ring locations, from where it had left, when it entered the suspend mode.

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

CSR6: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR7: Extended Control and Interrupt 2

Certain bits in CSR7 indicate the cause of an interrupt. The register is designed so that these indicator bits are cleared by writing ones to those bit locations. This means that the software can read CSR7 and write back the value just read to clear the interrupt condition.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	FASTSPNDE	Fast Suspend Enable. When FASTSPNDE is set to 1, the Am79C976 controller performs a fast suspend whenever the SPND bit is set. When a fast suspend is requested, the Am79C976 controller performs a quick entry into the suspend mode. At the time the SPND bit is set, the Am79C976 controller will complete the DMA process of any transmit and/or receive packet that had already begun DMA activity. In addition, any transmit packet that had started transmission will be fully transmitted and any receive packet that had begun reception will be fully received. However, no additional packets will be transmitted or received and no additional transmit or receive DMA activity will begin. Hence, the Am79C976 controller may enter the suspend mode with transmit and/or receive packets still in the FIFOs or the SRAM. When FASTSPNDE is 0 and the SPND bit is set, the Am79C976 controller may take longer before entering the suspend mode. At the time the SPND bit is set, the Am79C976 controller will complete the DMA process of a transmit packet if it had already begun and the Am79C976 controller will completely receive a receive packet if it had already begun. Additionally, all transmit packets stored in the transmit FIFOs and the transmit buffer area in the SRAM (if one is enabled) will be transmitted and all receive packets stored in the receive FIFOs, and the receive buffer area in the SRAM (if one is enabled) will be transferred into system memory. Since the FIFO and SRAM contents are flushed, it may take much longer before the Am79C976 controller enters the suspend mode. The amount of time that it takes depends on many factors including the size of the SRAM, bus latency, and network traffic level. When a write to CSR5 is performed with bit 0 (SPND) set to 1, the value that is simultaneously written to FASTSPNDE is used to determine which approach is used to enter suspend mode. Read/Write accessible. FASTSPNDE is cleared by H_RESET, S_RESET or by setting the STOP bit.
14	RES	Reserved location. Written as zero, read as undefined.
13	RDMD	Receive Demand, when set, causes the Buffer Management Unit to access the Receive Descriptor Ring without waiting for the chain poll-time counter to expire. Read/Write accessible. RDMD is set by writing a 1. Writing a 0 has no effect. RDMD will be cleared by the Buffer Management Unit

Downloaded from Arrow.com.

when it fetches a receive Descriptor. RDMD is cleared by H_RESET or by S_RESET. RDMD is unaffected by setting the STOP bit.

12 CHDPOLL Disable Chain Polling. If CHDPOLL is set, the Buffer Management Unit will disable chain polling. Likewise, if CHDPOLL is cleared, automatic chain polling is enabled. If CHDPOLL is set and the Buffer Management Unit is in the middle of a buffer-changing operation, setting the RDMD bit in CMD0 or CSR7 will cause a poll of the current receive descriptor, and setting the TDMD bit in CMD0 or CSR0 will cause a poll of the current transmit descriptor. If CHDPOLL is set, the RDMD bit in CSR7 can be set to initiate a manual poll of a receive or transmit descriptor if the Buffer Management Unit is in the middle of a buffer-chaining operation.

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

11 STINT Software Timer Interrupt. The Software Timer interrupt is set by the Am79C976 controller when the Software Timer counts down to 0. The Software Timer will immediately load the STVAL (BCR 31, bits 5-0) into the Software Timer and begin counting down.

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

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

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

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

9 MREINT MII Management Read Error Interrupt. The MII Read Error interrupt is set by the Am79C976 controller to indicate that the currently read register from the external PHY is invalid. The contents of BCR34 are incorrect and that the operation should be performed again. The indication of an incorrect read comes from the PHY. During the read turnaround time of the MII management frame the external PHY should drive the MDIO pin to a LOW state. If this does not happen, it indicates that the PHY and the Am79C976 controller have lost synchronization.

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

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

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

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

7 MAPINT MII Management Auto-Poll Interrupt. This bit is set when the Auto-Poll State Machine detects a change in any PHY register that is polled by the Auto-Poll State Machine.

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

Read/Write accessible. MAPINT is cleared by the host by writing a 1. Writing a 0 has no effect. MAPINT is cleared by H_RESET and

Downloaded from Arrow.com.

is not affected by S_RESET or setting the STOP bit.

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

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

5 MCCINT MII Management Command Complete Interrupt. The MII Management Command Complete Interrupt is set by the Am79C976 controller when a read or write operation to the MII Data Port (BCR34) is complete.

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

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

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

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

3 MCCIINT MII Management Command Complete Internal Interrupt. The MII Management Command Complete Interrupt is set by the Am79C976 controller when a read or write operation on the MII management port is complete from an internal operation. Examples of internal operations are Auto-Poll or MII Management Port generated MII management frames. These are normally hidden to the host.

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

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

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

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

1 MIIPDTINT MII PHY Detect Transition Interrupt. The MII PHY Detect Transition Interrupt is set by the Am79C976 controller whenever the MIIPD bit (BCR32, bit 14) transitions from 0 to 1 or vice versa.

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

Downloaded from Arrow.com.

0 MIIPDTINTE MII PHY Detect Transition Interrupt Enable. If MIIPDTINTE is set to 1, the MIIPDTINT bit will be able to set the INTR bit.

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

CSR8: Logical Address Filter 0

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

CSR9: Logical Address Filter 1

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

CSR10: Logical Address Filter 2

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

CSR11: Logical Address Filter 3

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

CSR12: Physical Address Register 0

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	PADR[15:0]	Physical Address Register, PADR[15:0]. The contents of this register are loaded from EEPROM after H_RESET or by an EEPROM read command (PREAD, BCR19, bit 14). If the EEPROM is not present, the contents of this register are undefined. This register can also be loaded from the initialization block after the INIT bit in CSR0 has been set

Downloaded from Arrow.com.

		or a direct register write has been performed on this register. Read/Write accessible. These bits are cleared by H_RESET but are unaffected by S_RESET, or STOP.

CSR13: Physical Address Register 1

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	PADR[31:16]	Physical Address Register, PADR[31:16]. The contents of this register are loaded from EEPROM after H_RESET or by an EEPROM read command (PREAD, BCR19, bit 14). If the EEPROM is not present, the contents of this register are undefined. This register can also be loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible. These bits are cleared by H_RESET but are unaffected by S_RESET, or STOP.

CSR14: Physical Address Register 2

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	PADR[47:32]	Physical Address Register, PADR[47:32].The contents of this register are loaded from EEPROM after H_RESET or by an EEPROM read command (PREAD, BCR19, bit 14). If the EEPROM is not present, the contents of this register are undefined. This register can also be loaded from the initialization block after the INIT bit in CSR0 has been set or a direct register write has been performed on this register. Read/Write accessible. These bits are cleared by H_RESET but are unaffected by S_RESET, or STOP.

CSR15: Mode

This register's fields are loaded during the Am79C976 controller initialization routine with the corresponding Initialization Block values, or when a direct register write has been performed on this register.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	PROM	Promiscuous Mode. When PROM = 1, all incoming receive frames are accepted. Read/Write accessible.
14	DRCVBC	Disable Receive Broadcast. When set, disables the Am79C976 controller from receiving broadcast messages. Used for protocols that do not support broadcast addressing, except as a function of multicast. DRCVBC is cleared by activation of H_RESET or S_RESET (broadcast messages will be received) and is unaffected by STOP. Read/Write accessible.
13	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the Am79C976 controller will be disabled. Frames addressed to the node's individual physical address will not be recognized. Read/Write accessible.
12-9	RES	Reserved locations. Written as zeros and read as undefined.
8-7	PORTSEL[1:0]	Obsolete function. Writing has no effect. Read as undefined.
6	INTL	Obsolete function. Writing has no effect. Read as undefined.

Downloaded from Arrow.com.

5 DRTY Disable Retry. When DRTY is set to 1, the Am79C976 controller will attempt only one transmission. In this mode, the device will not protect the first 64 bytes of frame data in the Transmit FIFO from being overwritten, because automatic retransmission will not be necessary. When DRTY is set to 0, the Am79C976 controller will attempt 16 transmissions before signaling a retry error.

Read/Write accessible.

4 FCOLL Force Collision. This bit allows the collision logic to be tested. The Am79C976 controller must be in internal loopback for FCOLL to be valid. If FCOLL = 1, a collision will be forced during loopback transmission attempts, which will result in a Retry Error. If FCOLL = 0, the Force Collision logic will be disabled. FCOLL is defined after the initialization block is read.

Read/Write accessible.

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

When the auto padding logic, which is enabled by the APAD_XMT bit (CSR4, bit11), adds padding to a frame, a valid FCS field is appended to the frame, regardless of the state of DXMTFCS.

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

This bit was called DTCR in the LANCE (Am7990) device.

Read/Write accessible.

2 LOOP Loopback Enable allows the Am79C976 controller to operate in full-duplex mode for test purposes. The setting of the full-duplex control bits in BCR9 have no effect when the device operates in loopback mode. When LOOP = 1, loopback is enabled. In combination with INTL and MIIILP, various loopback modes are defined as follows:.

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

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

Read/Write accessible. LOOP is cleared by H_RESET or S_RESET and is unaffected by STOP.

1 DTX Disable Transmit results in Am79C976 controller not accessing the Transmit Descriptor Ring and, therefore, no transmissions are attempted. DTX = 0, will set TXON bit (CSR0 bit 4) if STRT (CSR0 bit 1) is asserted.

Read/Write accessible.

0 DRX Disable Receiver results in the Am79C976 controller not accessing the Receive Descriptor Ring and, therefore, all receive frame data are ignored. DRX = 0, will set RXON bit (CSR0 bit 5) if STRT (CSR0 bit 1) is asserted.

Read/Write accessible.

CSR16-23: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

Downloaded from Arrow.com.

CSR24: Base Address of Receive Ring Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BADRL	Contains the lower 16 bits of the base address of the Receive Ring. Read/Write accessible. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR25: Base Address of Receive Ring Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BADRU	Contains the upper 16 bits of the base address of the Receive Ring. Read/Write accessible. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR26-29: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR30: Base Address of Transmit Ring Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BADXL	Contains the lower 16 bits of the base address of the Transmit Ring. Read/Write accessible. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR31: Base Address of Transmit Ring Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BADXU	Contains the upper 16 bits of the base address of the Transmit Ring.

Downloaded from Arrow.com.

Read/Write accessible. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR32-46: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR47: Transmit Polling Interval

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TXPOLLINT	Transmit Polling Interval. This register contains the time that the Am79C976 controller will wait between successive polling operations. The TXPOLLINT value is expressed as the two's complement of the desired interval, where each bit of TXPOLLINT represents 1 PCI clock period. TXPOLLINT[3:0] are ignored. (TXPOLLINT[16] is implied to be a one, so TXPOLLINT[15] is significant and does not represent the sign of the two's complement TXPOLLINT value.) The default value of this register is 0000h. This corresponds to a polling interval of 65,536 clock periods (2.185 ms when ERCLK = 90 MHz). Setting the INIT bit starts an initialization process that sets TXPOLLINT to its default value. If the user wants to program a value for TXPOLLINT other than the default, then he must change the value after the initialization sequence has completed. If the user does *not* use the initialization block to initialize the Am79C976 device, but instead, chooses to write directly to each of the registers that are involved in the INIT operation, then it is imperative that the user also writes an acceptable value to CSR47 as part of the alternative initialization sequence.

Read/Write accessible. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR48: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR49: Chain Polling Interval

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	CHPOLLINT	Chain Polling Interval. This register contains the time that the Am79C976 controller will wait between successive polling operations when the Buffer Management Unit is in the middle of a buffer chaining operation. The CHPOLLINT value is expressed as the two's complement of the desired interval, where each bit of CHPOLLINT approximately represents one PCI clock time period. CHPOLLINT[3:0] are ignored. (CHPOLLINT[16] is implied to be a 1, so CHPOLLINT[15] is significant and does not represent the sign of the two's complement CHPOLLINT value.) The default value of this register is 0000h. This corresponds to a polling interval of 65,536 clock periods (1.966 ms when CLK = 33 MHz). Setting the INIT bit starts an initialization process that sets CHPOLLINT to its default value. If the user wants to program a value for CHPOLLINT other than the default, then he must change the value after the initialization sequence has completed. If the user does *not* use the initialization block to initialize the Am79C976 device, but instead, chooses to write directly to each of the registers that are involved in the INIT operation, then it is imperative that the user also writes an acceptable value to CSR49 as

Downloaded from Arrow.com.

part of the alternative initialization sequence.

Read/Write accessible. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR50-57: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR58: Software Style

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-11	RES	Reserved locations. Written as zeros and read as undefined.
10	APERREN	Obsolete function. Writing has no effect. Read as undefined.
9	RES	Reserved locations. Written as zeros and read as undefined.
8	SSIZE32	Software Size 32 bits. When set, this bit indicates that the Am79C976 controller utilizes 32-bit software structures for the initialization block and the transmit and receive descriptor entries. When cleared, this bit indicates that the Am79C976 controller utilizes 16-bit software structures for the initialization block and the transmit and receive descriptor entries. In this mode, the Am79C976 controller is backwards compatible with the Am7990 LANCE and Am79C960 PCnet-ISA controllers. The value of SSIZE32 is determined by the Am79C976 controller according to the setting of the Software Style (SWSTYLE, bits 7-0 of this register). SSIZE32 is read only; write operations will be ignored. SSIZE32 will be cleared after H_RESET (since SWSTYLE defaults to 0) and is not affected by S_RESET or STOP. If SSIZE32 is reset, then bits IADR[31:24] of CSR2 will be used to generate values for the upper 8 bits of the 32-bit address bus during master accesses initiated by the Am79C976 controller. This action is required, since the 16-bit software structures specified by the SSIZE32 = 0 setting will yield only 24 bits of address for the Am79C976 controller bus master accesses. If SSIZE32 is set, then the software structures that are common to the Am79C976 controller and the host system will supply a full 32 bits for each address pointer that is needed by the Am79C976 controller for performing master accesses. The value of the SSIZE32 bit has no effect on the drive of the upper 8 address bits. The upper 8 address pins are always driven, regardless of the state of the SSIZE32 bit. Note that the setting of the SSIZE32 bit has no effect on the defined width for I/O resources. I/O resource width is determined by the state of the DWIO bit (BCR18, bit 7).
7-0	SWSTYLE	Software Style register. The value in this register determines the style of register and memory resources that shall be used by the Am79C976 controller. The Software Style selection will affect the interpretation of a few bits within the CSR space, the order of the descriptor entries and the width of the descriptors and initialization block entries. All Am79C976 controller CSR bits and BCR bits and all descriptor, buffer, and initialization block entries not cited in Table 85 are unaffected by the Software Style selection and are, therefore, always fully functional as specified in the CSR and BCR sections.

Downloaded from Arrow.com.

Read/Write accessible. The SWSTYLE register will contain the value 00h following H_RESET and will be unaffected by S_RESET or STOP.

Table 85. Software Styles

SWSTYLE [7:0]	Style Name	SSIZE32	Initialization Block Entries	Descriptor Ring Entries
00h	LANCE/ PCnet-ISA controller	0	16-bit software structures, non-burst or burst access	16-bit software structures, non-burst access only
01h	RES	1	RES	RES
02h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only
03h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst or burst access
04h	VLAN	1	Not Used	32-bit software structures, non-burst or burst access
05h	64-bit Address	1	Not Used	32-bit software structures, 32-byte descriptors, non-burst or burst access
All Other	Reserved	Undefined	Undefined	Undefined

Downloaded from Arrow.com.

CSR59-75: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR76: Receive Ring Length

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCVRL	Receive Ring Length. Contains the two's complement of the receive descriptor ring length. This register is initialized during the optional Am79C976 controller initialization routine based on the value in the RLEN field of the initialization block. However, this register can be manually altered. The actual receive ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. Read/Write accessible. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR77: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR78: Transmit Ring Length

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	XMTRL	Transmit Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the optional Am79C976 controller initialization routine based on the value in the TLEN field of the initialization block. However, this register can be manually altered. The actual transmit ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. Read/Write accessible. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR79: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR80: DMA Transfer Counter and FIFO Threshold Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-14	RES	Reserved locations. Written as zeros and read as undefined.
13-12	RCVFW[1:0]	Receive FIFO Watermark. RCVFW controls the point at which receive DMA is requested in relation to the number of received bytes in the Receive FIFO. RCVFW specifies the number of bytes which must be present (once the frame has been verified as a non-runt) before receive DMA is requested. Note however that, if the network interface is operating in half-duplex mode, in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively avoids having to react to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature is enabled or if the network interface is operating in full-duplex mode, receive DMA will be requested as soon as either the RCVFW threshold is reached, or a complete valid receive frame is detected (regardless of length). When the Full Duplex Runt Packet Accept Disable (FDRPAD) bit (BCR9, bit 2) is set and the Am79C976 controller is in full-duplex mode, in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively disables the runt packet accept feature in full duplex.

Downloaded from Arrow.com.

Table 86. Receive Watermark Programming

RCVFW[1:0]	Bytes Received
00	48
01	64
10	128
11	256

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

11-10 XMTSP[1:0] Transmit Start Point. XMTSP controls the point at which preamble transmission attempts to commence in relation to the number of bytes written to the MAC Transmit FIFO for the current transmit frame. When the entire frame is in the MAC Transmit FIFO, transmission will start regardless of the value in XMTSP. If the network interface is operating in half-duplex mode, regardless of XMTSP, the FIFO will not internally overwrite its data until at least 64 bytes (or the entire frame if shorter than 64 bytes) have been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be rewritten to the Transmit FIFO, and retries will be handled autonomously by the MAC. If the Disable Retry feature is enabled, or if the network is operating in full-duplex mode, the Am79C976 controller can overwrite the beginning of the frame as soon as the data is transmitted, because no collision handling is required in these modes.

Note that when the No Underflow (NOUFLO) bit (BCR18, bit 11) is set to 1, there is the additional restriction that the complete transmit frame must be DMA'd into the Am79C976 controller and reside within the MAC Transmit FIFO. This mode is useful in a system where high latencies cannot be avoided. See Table 87.

Read/Write accessible. XMTSP is set to a value of 01b (64 bytes) after H_RESET or S_RESET and is unaffected by STOP.

Table 87. Transmit Start Point Programming

XMTSP[1:0]	NOUFLO	Bytes Written
00	0	16
01	0	64
10	0	128
11	0	Full Frame
XX	1	Full Frame

9-8 XMTFW[1:0] Transmit FIFO Watermark. XMTFW specifies the point at which transmit DMA is requested, based upon the number of bytes that could be written to the Transmit FIFO without FIFO overflow. Transmit DMA is requested at any time when the number of bytes specified by XMTFW could be written to the FIFO without causing Transmit FIFO overflow, and the internal state machine has reached a point where the Transmit FIFO is checked to determine if DMA servicing is required. See Table 88.

Table 88. Transmit Watermark Programming

XMTFW[1:0]	Bytes Available
00	16
01	64
10	128
11	256

Read/Write accessible. XMTFW is set to a value of 00b (16 bytes) after H_RESET or S_RESET and is unaffected by STOP.

7-0 DMATC[7:0] Obsolete function. Writing has no effect. Read as undefined.

Downloaded from Arrow.com.

CSR81-87: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR88: Chip ID Register Lower

Bit	Name	Description
31-16	RES	Reserved locations. Read as undefined.
15-12	PARTIDL	Lower 4 bits of the Am79C976 controller part number, i.e. 1000b (8h). PARTIDU and PARTIDL comprise the 16-bit code for the Am79C976 controller, which is 0010 0110 0010 1000b (2628h). This register is exactly the same as the Device ID register in the JTAG description. However, this part number is different from that stored in the Device ID register in the PCI configuration space. PARTIDL is read only. Write operations are ignored.
11-1	MANFID	Manufacturer ID. The 11-bit manufacturer code for AMD is 00000000001b. This code is per the JEDEC Publication 106-A. Note that this code is not the same as the Vendor ID in the PCI configuration space. MANFID is read only. Write operations are ignored.
0	ONE	Always a logic 1. ONE is read only. Write operations are ignored.

CSR89: Chip ID Register Upper

Bit	Name	Description
31-16	RES	Reserved locations. Read as undefined.
15-12	VER	Version. This 4-bit pattern is silicon-revision dependent. VER is read only. Write operations are ignored.
11-0	PARTIDU	Upper 12 bits of the Am79C976 controller part number, i.e., 0010 0110 0010b (262h). PARTIDU is read only. Write operations are ignored.

CSR90-99: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR100: Bus Timeout

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MERRTO	Obsolete function. Writing has no effect. Read as undefined.

CSR101-111: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR112: Missed Frame Count

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MFC	Obsolete function. Replaced by MIB counter. Writing has no effect. Read as undefined.

CSR113: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR114: Receive Collision Count

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCC	Obsolete function. Replaced by MIB counter. Writing has no effect. Read as undefined.

CSR115: Reserved

Bit	Name	Description

Downloaded from Arrow.com.

31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR116: OnNow Power Mode Register

Note: *Bits 10-0 in this register are programmable through the EEPROM.*

Bit	Name	Description
31-11	RES	Reserved locations. Written as zeros and read as undefined.
10	PME_EN_OVR	PME_EN Overwrite. When this bit is set and the MPMAT or LCDET bit is set, the PME pin will always be asserted regardless of the state of PME_EN bit. Read/Write accessible. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
9	LCDET	Link Change Detected. This bit is set when the MII auto-polling logic detects a change in link status and the LCMODE bit is set. This bit can be cleared to 0 either by writing 1 to CSR116, bit 9 or by writing 1 to STAT0, bit 10. LCDET is cleared when power is initially applied (POR). Read/Write accessible.
8	LCMODE	Link Change Wake-up Mode. When this bit is set to 1, the LCDET bit gets set when the MII auto polling logic detects a Link Change. Read/Write accessible. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7	PMAT	Pattern Matched. This bit is set when the PMMODE bit is set and an OnNow pattern match occurs. This bit can be cleared to 0 either by writing 0 to CSR116, bit 7 or by writing 1 to STAT0, bit 12. PMAT is cleared when power is initially applied (POR). Read/Write accessible.
6	EMPPLBA	Magic Packet Physical Logical Broadcast Accept. If both EMPPLBA and MPPLBA (CSR5, bit 5) are at their default value of 0, the Am79C976 controller will only detect a Magic Packet frame if the destination address of the packet matches the content of the physical address register (PADR). If either EMPPLBA or MPPLBA is set to 1, the destination address of the Magic Packet frame can be unicast, multicast, or broadcast. Note that the setting of EMPPLBA and MPPLBA only affects the address detection of the Magic Packet frame. The Magic Packet frame's data sequence must be made up of 16 consecutive physical addresses (PADR[47:0]) regardless of what kind of destination address it has. Read/Write accessible. EMPPLBA is set to 0 by H_RESET or S_RESET and is not affected by setting the STOP bit.
5	MPMAT	Magic Packet Match. This bit is set when PCnet-FAST+ detects a Magic Packet while it is in the Magic Packet mode. This bit can be cleared to 0 either by writing 0 to CSR116, bit 5 or by writing 1 to STAT0, bit 11. MPMAT is cleared when power is initially applied (POR). Read/Write accessible.
4	MPPEN	Magic Packet Pin Enable. When this bit is set, the device enters the Magic Packet mode when the PG input goes LOW or MPEN bit (CSR5, bit 2) gets set to 1. This bit is OR'ed with MPEN (CSR5, bit 2). Read/Write accessible. Cleared by H_RESET and is not affected

Downloaded from Arrow.com.

by S_RESET or setting the STOP bit.

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

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

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

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

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

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

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

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

CSR117-121: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR122: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR123: Reserved

Bit	Name	Description
31-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR124: Test Register 1

This register is used to place the Am79C976 controller into various test modes. The Runt Packet Accept is the only user accessible test mode. All other test modes are for AMD internal use only.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-4	RES	Reserved locations. Written as zeros and read as undefined.
3	RPA	Runt Packet Accept. This bit forces the Am79C976 controller to accept runt packets (packets shorter than 64 bytes). The minimum packet size that can be received is 12 bytes. Read/Write accessible. RPA is cleared by H_RESET or S_RESET and is not affected by STOP.
2-0	RES	Reserved locations. Written as zeros and read as undefined.

CSR125: MAC Enhanced Configuration Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	IPG	Inter Packet Gap. This value indicates the minimum number of network bit times after the end of a frame that the transmitter will wait before it starts transmitting another frame. In half-duplex mode the end of the frame is determined by CRS, while in full-duplex mode the end of the frame is determined by TX_EN. The IPG value can be adjusted to compensate for delays through the external PHY device. IPG should be programmed to the nearest nibble. The two least significant bits are ignored. For

Downloaded from Arrow.com.

example, programming IPG to 63h has the same effect as programming it to 60h.

***CAUTION**: Use this parameter with care. By lowering the IPG below the IEEE 802.3 standard 96 bit times, the Am79C976 controller can interrupt normal network behavior.*

Read/Write accessible. IPG is set to 60h (96 Bit times) by H_RESET or S_RESET and is not affected by STOP.

7-0 IFS1

InterFrameSpacingPart1. Changing IFS1 allows the user to program the value of the InterFrame-SpacePart1 timing. The Am79C976 controller sets the default value at 60 bit times (3ch). See the subsection on *Medium Allocation* in the section *Media Access Management* for more details. The equation for setting IFS1 when IPG ≥ 96 bit times is:

IFS1 = IPG - 36 bit times

IPG should be programmed to the nearest nibble. The two least significant bits are ignored. For example, programming IPG to 63h has the same effect as programming it to 60h.

Read/Write accessible. IFS1 is set to 3ch (60 bit times) by H_RESET or S_RESET and is not affected by STOP.

Downloaded from Arrow.com.

Bus Configuration Registers

The Bus Configuration Registers (BCRs) are included for compatibility with older PCnet Family software All BCR functions can be accessed more efficiently through the memory-mapped registers.

BCRs are used to program the configuration of the bus interface and other special features of the Am79C976 controller that are not related to the IEEE 8802-3 MAC functions. The BCRs are accessed by first setting the appropriate RAP value and then by performing a slave access to the BDP. See Table 89.

All BCR registers are 16 bits in width in Word I/O mode (DWIO = 0, BCR18, bit 7) and 32 bits in width in DWord I/O mode (DWIO = 1). The upper 16 bits of all BCR registers is undefined when in DWord I/O mode. These bits should be written as zeros and should be treated as undefined when read. The default value given for any BCR is the value in the register after H_RESET.

Some of these values may be changed shortly after H_RESET when the contents of the external EEPROM is automatically read in. None of the BCR register values are affected by the assertion of the STOP bit or S_RESET.

Note that several registers have no default value. BCR0, BCR1, BCR3, BCR8, BCR10-17, and BCR21 are reserved and have undefined values. BCR2 and BCR34 are not observable without first being programmed through the EEPROM read operation or a user register write operation.

BCR0, BCR1, BCR16, BCR17, and BCR21 are registers that are used by other devices in the PCnet family. Writing to these registers have no effect on the operation of the Am79C976 controller.

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

Table 89. BCR Registers

RAP	Mnemonic	Default	Name	Programmability	
				User	EEPROM
0	MSRDA	0005h	Reserved	No	No
1	MSWRA	0005h	Reserved	No	No
2	MC	0000h	Miscellaneous Configuration	Yes	Yes
3	Reserved	N/A	Reserved	No	No
4	LED0	00C0h	LED0 Status	Yes	Yes
5	LED1	0094h	LED1 Status	Yes	Yes
6	LED2	1080h	LED2 Status	Yes	Yes
7	LED3	0081h	LED3 Status	Yes	Yes
8	Reserved	N/A	Reserved	No	No
9	FDC	0004h	Full-Duplex Control	Yes	Yes
10-15	Reserved	N/A	Reserved	No	No
16	IOBASEL	N/A	Reserved	No	No
17	IOBASEU	N/A	Reserved	No	No
18	BSBC	9000h	Burst and Bus Control	Yes	Yes
19	EECAS	0000h	EEPROM Control and Status	Yes	No
20	SWS	0000h	Software Style	Yes	No
21	Reserved	N/A	Reserved	No	No
22	PCILAT	1818h	PCI Latency	Yes	Yes
23	PCISID	0000h	PCI Subsystem ID	No	Yes
24	PCISVID	0000h	PCI Subsystem Vendor ID	No	Yes

Downloaded from Arrow.com.

25	SRAMSIZE	0000h	SRAM Size	Yes	Yes
26	SRAMBND	0000h	SRAM Boundary	Yes	Yes
27	Reserved	N/A	Reserved	No	No
28	EBADDRL	N/A	Expansion Bus Address Lower	Yes	No
29	EBADDRU	N/A	Expansion Bus Address Upper	Yes	No
30	EBDR	N/A	Expansion Bus Data Port	Yes	No
31	STVAL	FFFFh	Software Timer Value	Yes	No
32	MIICAS	0400h	MII Control and Status	Yes	Yes
33	MIIADDR	N/A	MII Address	Yes	Yes
34	MIIMDR	N/A	MII Management Data	Yes	No
35	PCIVID	1022h	PCI Vendor ID	No	Yes
36	PMC_A	C802h	PCI Power Management Capabilities (PMC) Alias Register	No	Yes
37	DATA0	0000h	PCI DATA Register Zero Alias Register	No	Yes
38	DATA1	0000h	PCI DATA Register One Alias Register	No	Yes
39	DATA2	0000h	PCI DATA Register Two Alias Register	No	Yes
40	DATA3	0000h	PCI DATA Register Three Alias Register	No	Yes
41	DATA4	0000h	PCI DATA Register Four Alias Register	No	Yes
42	DATA5	0000h	PCI DATA Register Five Alias Register	No	Yes
43	DATA6	0000h	PCI DATA Register Six Alias Register	No	Yes
44	DATA7	0000h	PCI DATA Register Seven Alias Register	No	Yes
45	PMR1	N/A	Pattern Matching Register 1	Yes	No
46	PMR2	N/A	Pattern Matching Register 2	Yes	No
47	PMR3	N/A	Pattern Matching Register 3	Yes	No

BCR0: Master Mode Read Active

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MSRDA	Reserved locations. After H_RESET, the value in this register will be 0005h. The setting of this register has no effect on any Am79C976 controller function. It is only included for software compatibility with other PCnet family devices. Read always. MSRDA is read only. Write operations have no effect.

BCR1: Master Mode Write Active

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MSWRA	Reserved locations. After H_RESET, the value in this register will be 0005h. The setting of this register has no effect on any Am79C976 controller function. It is only included for software compatibility with other PCnet family devices. Read always. MSWRA is read only. Write operations have no effect.

Downloaded from Arrow.com.

BCR2: Miscellaneous Configuration

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-13	RES	Reserved locations. Written and read as zeros.
12	LEDPE	LED Program Enable. When LEDPE is set to 1, programming of the LED0 (BCR4), LED1 (BCR5), LED2 (BCR6), and LED3 (BCR7) registers is enabled. When LEDPE is cleared to 0, programming of LED0 (BCR4), LED1 (BCR5), LED2 (BCR6), and LED3 (BCR7) registers is disabled. Writes to those registers will be ignored. Read/Write accessible. LEDPE is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
11-9	RES	Reserved locations. Written and read as zeros.
8	APROMWE	Address PROM Write Enable. The Am79C976 controller contains a shadow RAM on board for storage of the first 16 bytes loaded from the serial EEPROM. Accesses to Address PROM I/O Resources will be directed toward this RAM. When APROMWE is set to 1, then write access to the shadow RAM will be enabled. Read/Write accessible. APROMWE is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
7	INTLEVEL	Interrupt Level. This bit allows the interrupt output signals to be programmed for level or edge-sensitive applications. When INTLEVEL is cleared to 0, the $\overline{\text{INTA}}$ pin is configured for level-sensitive applications. In this mode, an interrupt request is signaled by a low level driven on the $\overline{\text{INTA}}$ pin by the Am79C976 controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is tri-stated by the Am79C976 controller and allowed to be pulled to a high level by an external pull-up device. This mode is intended for systems which allow the interrupt signal to be shared by multiple devices. When INTLEVEL is set to 1, the $\overline{\text{INTA}}$ pin is configured for edge-sensitive applications. In this mode, an interrupt request is signaled by a high level driven on the $\overline{\text{INTA}}$ pin by the Am79C976 controller. When the interrupt is cleared, the $\overline{\text{INTA}}$ pin is driven to a low level by the Am79C976 controller. This mode is intended for systems that do not allow interrupt channels to be shared by multiple devices. INTLEVEL should not be set to 1 when the Am79C976 controller is used in a PCI bus application. Read/Write accessible. INTLEVEL is cleared to 0 by H_RESET and is unaffected by S_RESET or by setting the STOP bit.
6-4	RES	Reserved locations. Written as zeros and read as undefined.
3	EADISEL	Obsolete function. Writing has no effect. Read as undefined.
2	RES	Reserved location. Written and read as zeros.
1	ASEL	Obsolete function. Writing has no effect. Read as undefined.
0	RES	Reserved location. Written and read as zeros.

BCR4: LED0 Status

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

Note: *When LEDPE (BCR2, bit 12) is set to 1, programming of the LED0 Status register is enabled.*

Downloaded from Arrow.com.

When LEDPE is cleared to 0, programming of the LED0 register is disabled. Writes to this register will be ignored.

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of 1 in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (bits 8 and 6-0). This bit is read only; writes have no effect. LEDOUT is unaffected by H_RESET, S_RESET, or STOP.
14	LEDPOL	LED Polarity. When this bit has the value 0, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false (i.e., the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit). When this bit has the value 1, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true, and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e., the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit.). The setting of this bit will not effect the polarity of the LEDOUT bit for this register. Read/Write accessible. LEDPOL is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then the LED output value will be governed by the LEDOUT and LEDPOL values. Read/Write accessible. LEDDIS is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
12	100E	100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C976 controller is operating at 100 Mbps mode. Read/Write accessible. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
11-10	RES	Reserved locations. Written and read as zeros.
9	MPSE	Magic Packet Status Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when Magic Packet frame mode is enabled and a Magic Packet frame is detected on the network. Read/Write accessible. MPSE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
8	FDLSE	Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LEDOUT signal when the Am79C976 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C976 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal.

Downloaded from Arrow.com.

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

7 PSE — Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher.

Read/Write accessible. PSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

6 LNKSE — Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register when in Link Pass state.

Read/Write accessible. LNKSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

5 RCVME — Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network that has passed the address match function for this node. All address matching modes are included: physical, logical filtering, broadcast and promiscuous.

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

4 XMTE — Transmit Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is transmit activity on the network.

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

3 RES — Reserved location. Written and read as zeros.

2 RCVE — Receive Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network.

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

1 SFBDE — Start Frame/Byte Delimiter Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the RXD[3:0] pins are presenting the least significant nibble of valid frame data.

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

0 COLE — Collision Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is collision activity on the network.

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

BCR5: LED1 Status

BCR5 controls the function(s) that the $\overline{\text{LED1}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR5 defaults to transmit or receive activity (XMTE = 1 and RCVE = 1) with pulse stretcher enabled (PSE = 1) and is fully programmable.

Note: *When LEDPE (BCR2, bit 12) is set to 1, programming of the LED1 Status register is enabled. When LEDPE is cleared to 0, programming of the LED1 register is disabled. Writes to this register will be ignored.*

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of 1 in this bit indicates that the OR of the enabled signals is true.

Downloaded from Arrow.com.

		The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (bits 8 and 6-0). This bit is read only, writes have no effect. LEDOUT is unaffected by H_RESET, S_RESET, or STOP.
14	LEDPOL	LED Polarity. When this bit has the value 0, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true, and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false (i.e., the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit). When this bit has the value 1, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true, and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e., the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit). The setting of this bit will not effect the polarity of the LEDOUT bit for this register. Read/Write accessible. LEDPOL is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then the LED output value will be governed by the LEDOUT and LEDPOL values. Read/Write accessible. LEDDIS is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
12	100E	100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C976 controller is operating at 100 Mbps mode. Read/Write accessible. 100E is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
11-10	RES	Reserved locations. Written and read as zeros.
9	MPSE	Magic Packet Status Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when Magic Packet mode is enabled and a Magic Packet frame is detected on the network. Read/Write accessible. MPSE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
8	FDLSE	Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LEDOUT signal when the Am79C976 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C976 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal. Read/Write accessible. FDLSE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7	PSE	Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher. Read/Write accessible. PSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

Downloaded from Arrow.com.

6	LNKSE	Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register in Link Pass state. Read/Write accessible. LNKSE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
5	RCVME	Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network that has passed the address match function for this node. All address matching modes are included: physical, logical filtering, broadcast, and promiscuous. Read/Write accessible. RCVME is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
4	XMTE	Transmit Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is transmit activity on the network. Read/Write accessible. XMTE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.
3	RES	Reserved location. Written and read as zeros.
2	RCVE	Receive Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network. Read/Write accessible. RCVE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.
1	SFBDE	Start Frame/Byte Delimiter Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the RXD[3:0] pins are presenting the least significant nibble of valid frame data. Read/Write accessible. RCVE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
0	COLE	Collision Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is collision activity on the network. Read/Write accessible. COLE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR6: LED2 Status

BCR6 controls the function(s) that the $\overline{\text{LED2}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR5 defaults to 100 Mb/s speed indication (100E = 1) with pulse stretcher enabled (PSE = 1).

Note: *When LEDPE (BCR2, bit 12) is set to 1, programming of the LED2 Status register is enabled. When LEDPE is cleared to 0, programming of the LED2 register is disabled. Writes to this register will be ignored.*

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of 1 in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (bits 8 and 6-0). This bit is read only; writes have no effect. LEDOUT is unaffected by H_RESET, S_RESET, or STOP.
14	LEDPOL	LED Polarity. When this bit has the value 0, the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true. The LED pin will be disabled

Downloaded from Arrow.com.

and allowed to float high whenever the OR of the enabled signals is false (i.e., the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit).

When this bit has the value 1, the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true. The LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e., the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit).

The setting of this bit will not effect the polarity of the LEDOUT bit for this register.

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

13 LEDDIS — LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then the LED output value will be governed by the LEDOUT and LEDPOL values.

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

12 100E — 100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C976 controller is operating at 100 Mbps mode.

Read/Write accessible. 100E is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

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

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

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

8 FDLSE — Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LEDOUT signal when the Am79C976 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C976 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal.

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

7 PSE — Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher.

Read/Write accessible. PSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

6 LNKSE — Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register in Link Pass state.

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

5 RCVME — Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network that has passed the address match function for this node. All address matching modes are included:

Downloaded from Arrow.com.

		physical, logical filtering, broadcast, and promiscuous. Read/Write accessible. RCVME is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
4	XMTE	Transmit Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is transmit activity on the network. Read/Write accessible. XMTE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
3	RES	Reserved location. Written and read as zeros.
2	RCVE	Receive Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network. Read/Write accessible. RCVE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
1	SFBDE	Start Frame/Byte Delimiter Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the RXD[3:0] pins are presenting the least significant nibble of valid frame data. Read/Write accessible. RCVE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
0	COLE	Collision Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is collision activity on the network. Read/Write accessible. COLE is cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR7: LED3 Status

BCR7 controls the function(s) that the $\overline{\text{LED3}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR7 defaults to collision indication (COLE = 1) with pulse stretcher enabled (PSE = 1) and is fully programmable.

Note: *When LEDPE (BCR2, bit 12) is set to 1, programming of the LED3 Status register is enabled. When LEDPE is cleared to 0, programming of the LED3 register is disabled. Writes to this register will be ignored.*

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of 1 in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (bits 8 and 6-0). This bit is read only; writes have no effect. LEDOUT is unaffected by H_RESET, S_RESET, or STOP.
14	LEDPOL	LED Polarity. When this bit has the value 0, the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true. The LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false (i.e., the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit). When this bit has the value 1, the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true. The LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e., the LED output will be a Totem Pole output and the output value will be the

Downloaded from Arrow.com.

same polarity as the LEDOUT status bit).

The setting of this bit will not effect the polarity of the LEDOUT bit for this register.

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

13 LEDDIS — LED Disable. This bit is used to disable the LED output. When LEDDIS has the value 1, then the LED output will always be disabled. When LEDDIS has the value 0, then the LED output value will be governed by the LEDOUT and LEDPOL values.

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

12 100E — 100 Mbps Enable. When this bit is set to 1, a value of 1 is passed to the LEDOUT bit in this register when the Am79C976 controller is operating at 100 Mbps mode.

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

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

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

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

8 FDLSE — Full-Duplex Link Status Enable. Indicates the Full-Duplex Link Test Status. When this bit is set, a value of 1 is passed to the LEDOUT signal when the Am79C976 controller is functioning in a Link Pass state and full-duplex operation is enabled. When the Am79C976 controller is not functioning in a Link Pass state with full-duplex operation being enabled, a value of 0 is passed to the LEDOUT signal.

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

7 PSE — Pulse Stretcher Enable. When this bit is set, the LED illumination time is extended for each new occurrence of the enabled function for this LED output. A value of 0 disables the pulse stretcher.

Read/Write accessible. PSE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

6 LNKSE — Link Status Enable. When this bit is set, a value of 1 will be passed to the LEDOUT bit in this register in Link Pass state.

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

5 RCVME — Receive Match Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network that has passed the address match function for this node. All address matching modes are included: physical, logical filtering, broadcast, and promiscuous.

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

4 XMTE — Transmit Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is transmit activity on the network.

Read/Write accessible. XMTE is cleared by H_RESET and is not

Downloaded from Arrow.com.

affected by S_RESET or setting the STOP bit.

3 RES — Reserved location. Written and read as zeros.

2 RCVE — Receive Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is receive activity on the network.

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

1 SFBDE — Start Frame/Byte Delimiter Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when the RXD[3:0] pins are presenting the least significant nibble of valid frame data.

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

0 COLE — Collision Status Enable. When this bit is set, a value of 1 is passed to the LEDOUT bit in this register when there is collision activity on the network.

Read/Write accessible. COLE is set to 1 by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR9: Full-Duplex Control

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-3	RES	Reserved locations. Written as zeros and read as undefined.
2	FDRPAD	Full-Duplex Runt Packet Accept Disable. When FDRPAD is set to 1 and full-duplex mode is enabled, the Am79C976 controller will only receive frames that meet the minimum Ethernet frame length of 64 bytes. Receive DMA will not start until at least 64 bytes or a complete frame have been received. When FDRPAD is cleared to 0, the Am79C976 controller will accept any frame of 12 bytes or greater, and receive DMA will start according to the programming of the receive FIFO watermark. Read/Write accessible. FDRPAD is set to 1 by H_RESET and is not affected by S_RESET or by setting the STOP bit.
1	RES	Reserved locations. Written as zeros and read as undefined.
0	FDEN	Full-Duplex Enable. FDEN controls whether full-duplex operation is enabled. When FDEN is cleared and the Auto-Negotiation is disabled, full-duplex operation is not enabled and the Am79C976 controller will always operate in the half-duplex mode. When FDEN is set, the Am79C976 controller will operate in full-duplex mode when the MII port is enabled. ***Do not set this bit when Auto-Negotiation is enabled***. Read/Write accessible. FDEN is reset to 0 by H_RESET, and is unaffected by S_RESET and the STOP bit.

BCR16: I/O Base Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-5	IOBASEL	Reserved locations. After H_RESET, the value of these bits will be undefined. The settings of these bits will have no effect on any Am79C976 controller function. It is only included for software compatibility with other PCnet family devices. Read/Write accessible. IOBASEL is not affected by S_RESET or STOP.
4-0	RES	Reserved locations. Written as zeros, read as undefined.

Downloaded from Arrow.com.

BCR17: I/O Base Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	IOBASEU	Reserved locations. After H_RESET, the value in this register will be undefined. The settings of this register will have no effect on any Am79C976 controller function. It is only included for software compatibility with other PCnet family devices. Read/Write accessible. IOBASEU is not affected by S_RESET or STOP.

BCR18: Burst and Bus Control Register

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	ROMTMG	Expansion ROM Timing. The value of ROMTMG is used to tune the timing for all accesses to the external Flash/EPROM. ROMTMG defines the amount of time that a valid address is driven on the ERA[19:0] pins. The register value specifies delay in number of ROMCLK cycles, where ROMCLK is an internal clock signal that runs at one fourth the speed of ERCLK. **Note**: Programming ROMTNG with a value of 0 is not permitted. To ensure adequate expansion ROM setup time, ROMTMG should be set to 1 plus tACC / (ROMCLK period), where tACC is the access time of the expansion ROM device (Flash or EPROM). (The extra ROMCLK cycle is added to account for the ERA[19:0] output delay from ROMCLK plus the ERD[7:0] setup time to ROMCLK.) This field is an alias of CTRL0, bits 11-8. Read/Write accessible. ROMTMG is set to the value of 1001b by H_RESET and is not affected by S_RESET or STOP. The default value allows the use of an Expansion ROM with an access time of 350 ns if ERCLK is running at 90 MHz.
11	NOUFLO	No Underflow on Transmit. When the NOUFLO bit is set to 1, the Am79C976 controller will not start transmitting the preamble for a packet until the Transmit Start Point (CTRL1, bits 16-17) requirement has been met *and* the complete packet has been copied into the transmit FIFO. When the NOUFLO bit is cleared to 0, the Transmit Start Point is the only restriction on when preamble transmission begins for transmit packets. Setting the NOUFLO bit guarantees that the Am79C976 controller will never suffer transmit underflows, because the arbiter that controls transfers to and from the SSRAM guarantees a worst case latency on transfers to and from the MAC and Bus Transmit FIFOs such that it will never underflow if the complete packet has been copied into the Am79C976 controller before packet transmission begins. Read/Write accessible. NOUFLO is cleared to 0 after H_RESET or S_RESET and is unaffected by STOP.
10	RES	Reserved location. Written as zeros and read as undefined.
9	MEMCMD	Obsolete function. Writing has no effect. Read as undefined.
8	EXTREQ	Obsolete function. Writing has no effect. Read as undefined.
7	DWIO	Double Word I/O. When set, this bit indicates that the Am79C976 controller is programmed for DWord I/O (DWIO) mode. When

Downloaded from Arrow.com.

cleared, this bit indicates that the Am79C976 controller is programmed for Word I/O (WIO) mode. This bit affects the I/O Resource Offset map and it affects the defined width of the Am79C976 controllers I/O resources. See the DWIO and WIO sections for more details.

The initial value of the DWIO bit is determined by the programming of the EEPROM.

The value of DWIO can be altered automatically by the Am79C976 controller. Specifically, the Am79C976 controller will set DWIO if it detects a DWord write access to offset 10h from the Am79C976 controller I/O base address (corresponding to the RDP resource).

Once the DWIO bit has been set to a 1, only a H_RESET or an EEPROM read can reset it to a 0. (Note that the EEPROM read operation will only set DWIO to a 0 if the appropriate bit inside of the EEPROM is set to 0.)

DWIO is read only, write operations have no effect. DWIO is cleared by H_RESET and is not affected S_RESET or by setting the STOP bit.

6	BREADE	Obsolete function. Writing has no effect. Read as undefined.
5	BWRITE	Obsolete function. Writing has no effect. Read as undefined.
4-3	TSTSHDW	Reserved locations. Written and read as zeros.
2-0	LINBC	Reserved locations. Written and read as zeros.

BCR19: EEPROM Control and Status

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	PVALID	EEPROM Valid status bit. PVALID is read only; write operations have no effect. A value of 1 in this bit indicates that a PREAD operation has occurred, and that (1) there is an EEPROM connected to the Am79C976 controller interface pins and (2) the contents read from the EEPROM have passed the checksum verification operation. A value of 0 in this bit indicates a failure in reading the EEPROM. The checksum for the EEPROM contents is incorrect or no EEPROM is connected to the interface pins. PVALID is set to 0 during H_RESET and is unaffected by S_RESET or the STOP bit. However, following the H_RESET operation, an automatic, sequential read of the EEPROM will be performed. Just as is true for the normal PREAD command, at the end of this automatic, sequential read operation, the PVALID bit may be set to 1. Therefore, H_RESET will set the PVALID bit to 0 at first, but the automatic EEPROM read operation may later set PVALID to a 1. If PVALID becomes 0 following an EEPROM read operation (either automatically generated after H_RESET, or requested through PREAD), then all EEPROM-programmable BCR locations will be reset to their H_RESET values. The content of the Address PROM locations, however, will not be cleared. If no EEPROM is present at the EESK, EEDI, and EEDO pins, then all attempted PREAD commands will terminate early and PVALID will *not* be set. This applies to the automatic read of the EEPROM after H_RESET, as well as to host-initiated PREAD commands.
14	PREAD	EEPROM Read command bit. When this bit is set to a 1 by the host, the PVALID bit (BCR19, bit 15) will immediately be reset to a 0, and then the Am79C976 controller will perform a sequential

Downloaded from Arrow.com.

read operation of the EEPROM through the interface. The EEPROM data that is fetched during the read will be stored in the appropriate internal registers on board the Am79C976 controller. Upon completion of the EEPROM read operation, the Am79C976 controller will assert the PVALID bit. EEPROM contents will be indirectly accessible to the host through read accesses to the Address PROM (offsets 0h through Fh) and through read accesses to other EEPROM programmable registers. Note that read accesses from these locations will not actually access the EEPROM itself, but instead will access the Am79C976 controller's internal copy of the EEPROM contents. Write accesses to these locations may change the Am79C976 controller register contents, but the EEPROM locations will not be affected. EEPROM locations may be accessed directly through BCR19.

At the end of the sequential read operation, the PREAD bit will automatically be reset to a 0 by the Am79C976 controller and PVALID will be set, provided that an EEPROM existed on the interface pins and that the checksum for the EEPROM contents was correct.

Note that when PREAD is set to a 1, then the Am79C976 controller will no longer respond to any accesses directed toward it, until the PREAD operation has completed successfully. The Am79C976 controller will terminate these accesses with the assertion of DEVSEL and STOP while TRDY is not asserted, signaling to the initiator to disconnect and retry the access at a later time.

If a PREAD command is given to the Am79C976 controller but no EEPROM is attached to the interface pins, the PREAD bit will be cleared to a 0, and the PVALID bit will remain reset with a value of 0. This applies to the automatic, sequential read of the EEPROM after H_RESET as well as to host initiated PREAD commands. EEPROM programmable locations on board the Am79C976 controller will be set to their default values by such an aborted PREAD operation. For example, if the aborted PREAD operation immediately followed the H_RESET operation, then the final state of the EEPROM programmable locations will be equal to the H_RESET programming for those locations.

If a PREAD command is given to the Am79C976 controller and the auto-detection pin (EESK/LED1/SFBD) indicates that no EEPROM is present, then the EEPROM read operation will still be attempted.

Note that at the end of the H_RESET operation, a read of the EEPROM will be performed automatically. This H_RESET-generated EEPROM read function will not proceed if the auto-detection pin (EESK/LED1) indicates that no EEPROM is present.

Read/Write accessible. PREAD is set to 0 during H_RESET and is unaffected by S_RESET or the STOP bit.

13 EEDET

EEPROM Detect. This bit indicates the sampled value of the EESK/LED1 pin at the end of H_RESET. This value indicates whether or not an EEPROM is present at the EEPROM interface. If this bit is a 1, it indicates that an EEPROM is present. If this bit is a 0, it indicates that an EEPROM is not present.

EEDET is read only; write operations have no effect. The value of this bit is determined at the end of the H_RESET operation. It is unaffected by S_RESET or the STOP bit.

Downloaded from Arrow.com.

Table 90 indicates the possible combinations of EEDET and the existence of an EEPROM and the resulting operations that are possible on the EEPROM interface.

Table 90. EEDET Setting

EEDET Value (BCR19[13])	EEPROM Connected?	Result if PREAD is Set to 1	Result of Automatic EEPROM Read Operation Following H_RESET
0	No	EEPROM read operation is attempted. Entire read sequence will occur; checksum failure will result; PVALID is reset to 0.	First two EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to 0.
0	Yes	EEPROM read operation is attempted. Entire read sequence will occur; checksum operation will pass; PVALID is set to 1.	First two EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to 0.
1	No	EEPROM read operation is attempted. Entire read sequence will occur; checksum failure will result; PVALID is reset to 0.	EEPROM read operation is attempted. Entire read sequence will occur; checksum failure will result; PVALID is reset to 0.
1	Yes	EEPROM read operation is attempted. Entire read sequence will occur; checksum operation will pass; PVALID is set to 1.	EEPROM read operation is attempted. Entire read sequence will occur; checksum operation will pass; PVALID is set to 1.

12-5 RES — Reserved locations. Written as zeros; read as undefined.

4 EEN — EEPROM Port Enable. When this bit is set to a 1, it causes the values of ECS, ESK, and EDI to be driven onto the EECS, EESK, and EEDI pins, respectively. If EEN = 0 and no EEPROM read function is currently active, then EECS will be driven LOW. When EEN = 0 and no EEPROM read function is currently active, EESK and EEDI pins will be driven by the LED registers BCR5 and BCR4, respectively. See Table 91.

Read/Write accessible. EEN is set to 0 by H_RESET and is unaffected by S_RESET or STOP.

Table 91. Interface Pin Assignment

$\overline{\text{RST}}$ Pin	*PREAD or Auto Read in Progress	EEN	EECS	EESK	EEDI
Low	X	X	0	Tri-State	Tri-State
High	1	X	Active	Active	Active
High	0	1	From ECS Bit of BCR19	From ESK Bit of BCR19	From EEDI Bit of BCR19
High	0	0	0	LED1	LED0

** PREAD and Auto Read required an EEPROM with support for sequential read (automatic address increment).*

3 RES — Reserved location. Written as zero and read as undefined.

2 ECS — EEPROM Chip Select. This bit is used to control the value of the EECS pin of the interface when the EEN bit is set to 1 and the PREAD bit is set to 0. If EEN = 1 and PREAD = 0 and ECS is set to a 1, then the EECS pin will be forced to a HIGH level at the rising edge of the next clock following bit programming. If EEN = 1 and PREAD = 0 and ECS is set to a 0, then the EECS pin will be forced to a LOW level at the rising edge of the next clock following bit programming. ECS has no effect on the output value of the EECS pin unless the PREAD bit is set to 0 and the EEN bit is set to 1.

Read/Write accessible. ECS is set to 0 by H_RESET and is not affected by S_RESET or STOP.

Downloaded from Arrow.com.

1	ESK	EEPROM Serial Clock. This bit and the EDI/EDO bit are used to control host access to the EEPROM. Values programmed to this bit are placed onto the EESK pin at the rising edge of the next clock following bit programming, except when the PREAD bit is set to 1 or the EEN bit is set to 0. If both the ESK bit and the EDI/EDO bit values are changed during one BCR19 write operation, while EEN = 1, then setup and hold times of the EEDI pin value with respect to the EESK signal edge are not guaranteed. ESK has no effect on the EESK pin unless the PREAD bit is set to 0 and the EEN bit is set to 1. Read/Write accessible. ESK is reset to 0 by H_RESET and is not affected by S_RESET or STOP.
0	EDI/EDO	EEPROM Data In/EEPROM Data Out. Data that is written to this bit will appear on the EEDI output of the interface, except when the PREAD bit is set to 1 or the EEN bit is set to 0. Data that is read from this bit reflects the value of the EEDO input of the interface. EDI/EDO has no effect on the EEDI pin unless the PREAD bit is set to 0 and the EEN bit is set to 1. Read/Write accessible. EDI/EDO is reset to 0 by H_RESET and is not affected by S_RESET or STOP.

BCR20: Software Style

This register is an alias of the location CSR58. Accesses to and from this register are equivalent to accesses to CSR58.

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-11	RES	Reserved locations. Written as zeros and read as undefined.
10	APERREN	Obsolete function. Writing has no effect. Read as undefined.
9	RES	Reserved locations. Written as zeros; read as undefined.
8	SSIZE32	Software Size 32 bits. When set, this bit indicates that the Am79C976 controller utilizes 32-bit software structures for the initialization block and the transmit and receive descriptor entries. When cleared, this bit indicates that the Am79C976 controller utilizes 16-bit software structures for the initialization block and the transmit and receive descriptor entries. In this mode, the Am79C976 controller is backwards compatible with the Am7990 LANCE and Am79C960 PCnet-ISA controllers. The value of SSIZE32 is determined by the Am79C976 controller according to the setting of the Software Style (SWSTYLE, bits 7-0 of this register). SSIZE32 is read only; write operations will be ignored. SSIZE32 will be cleared after H_RESET (since SWSTYLE defaults to 0) and is not affected by S_RESET or STOP. If SSIZE32 is reset, then bits IADR[31:24] of CSR2 will be used to generate values for the upper 8 bits of the 32-bit address bus during master accesses initiated by the Am79C976 controller. This action is required, since the 16-bit software structures specified by the SSIZE32 = 0 setting will yield only 24 bits of address for Am79C976 controller bus master accesses. If SSIZE32 is set, then the software structures that are common to the Am79C976 controller and the host system will supply a full 32 bits for each address pointer that is needed by the Am79C976 controller for performing master accesses.

Downloaded from Arrow.com.

The value of the SSIZE32 bit has no effect on the drive of the upper 8 address bits. The upper 8 address pins are always driven, regardless of the state of the SSIZE32 bit.

Note that the setting of the SSIZE32 bit has no effect on the defined width for I/O resources. I/O resource width is determined by the state of the DWIO bit (BCR18, bit 7).

7-0 SWSTYLE Software Style register. The value in this register determines the style of register and memory resources that shall be used by the Am79C976 controller. The Software Style selection will affect the interpretation of a few bits within the CSR space, the order of the descriptor entries and the width of the descriptors and initialization block entries.

All Am79C976 controller CSR bits and all descriptor, buffer, and initialization block entries not cited in the Table 92 are unaffected by the Software Style selection and are, therefore, always fully functional as specified in the CSR and BCR sections.

Read/Write accessible. The SWSTYLE register will contain the value 00h following H_RESET and will be unaffected by S_RESET or STOP.

Downloaded from Arrow.com.

Table 92. Software Styles

SWSTYLE [7:0]	Style Name	SSIZE32	Initialization Block Entries	Descriptor Ring Entries
00h	LANCE/ PCnet-ISA controller	0	16-bit software structures, non-burst or burst access	16-bit software structures, non-burst access only
01h	RES	1	RES	RES
02h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst access only
03h	PCnet-PCI controller	1	32-bit software structures, non-burst or burst access	32-bit software structures, non-burst or burst access
04h	VLAN	1	Not used	32-bit software structures, non-burst or burst access
05h	64-bit address	1	Not used	32-bit software structures, 32-byte descriptors, non-burst or burst access
All Other	Reserved	Undefined	Undefined	Undefined

BCR22: PCI Latency Register

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	MAX_LAT	Maximum Latency. Specifies the maximum arbitration latency the Am79C976 controller can sustain without causing problems to the network activity. The register value specifies the time in units of 1/4 microseconds. MAX_LAT is aliased to the PCI configuration space register MAX_LAT (offset 3Fh). The host will use the value in the register to determine the setting of the Am79C976 Latency Timer register. Read/Write accessible. MAX_LAT is set to the value of 18h by H_RESET which results in a default maximum latency of 6 microseconds. MAX_LAT is not affected by S_RESET or STOP.
7-0	MIN_GNT	Minimum Grant. Specifies the minimum length of a burst period the Am79C976 controller needs to keep up with the network activity. The length of the burst period is calculated assuming a clock rate of 33 MHz. The register value specifies the time in units of 1/4 µs. MIN_GNT is aliased to the PCI configuration space register MIN_GNT (offset 3Eh). The host will use the value in the register to determine the setting of the Am79C976 Latency Timer register. Read/Write accessible. MIN_GNT is set to the value of 18h by H_RESET which results in a default minimum grant of 6 µs. MIN_GNT is not affected by S_RESET or STOP.

BCR23: PCI Subsystem Vendor ID Register

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SVID	Subsystem Vendor ID. SVID is used together with SID (BCR24, bits 15-0) to uniquely identify the add-in board or subsystem the Am79C976 controller is used in. Subsystem Vendor IDs can be obtained from the PCI SIG. A value of 0 (the default) indicates that the Am79C976 controller does not support subsystem identification.

Downloaded from Arrow.com.

SVID is aliased to the PCI configuration space register Subsystem Vendor ID (offset 2Ch).

SVID is read only. Write operations are ignored. SVID is cleared to 0 by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR24: PCI Subsystem ID Register

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SID	Subsystem ID. SID is used together with SVID (BCR23, bits 15-0) to uniquely identify the add-in board or subsystem the Am79C976 controller is used in. The value of SID is up to the system vendor. A value of 0 (the default) indicates that the Am79C976 controller does not support subsystem identification. SID is aliased to the PCI configuration space register Subsystem ID (offset 2Eh). SID is read only. Write operations are ignored. SID is cleared to 0 by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR25: SRAM Size Register

Bit	Name	Description

Note: *Bits 7-0 in this register are programmable through the EEPROM.*

31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SRAM_SIZE	SRAM Size. Specifies the total size of the SSRAM buffer in units of 512-byte pages. For example, assume that the external memory consists of one 64K X 32 bit SSRAM, for a total of 256K bytes. In this case SRAM_SIZE should be set to 512 (256K divided by 512). This field must be initialized to the appropriate value, either by the EEPROM or by the host CPU. SRAM_SIZE must be set to a value less than or equal to 2000h. Values larger than 2000h will cause incorrect behavior. **Note**: The minimum allowed number of pages is eight for normal network operation. The Am79C976 controller will not operate correctly with less than the eight pages of memory. When the minimum number of pages is used, these pages must be allocated four each for transmit and receive. CAUTION: Programming SRAM_BND and SRAM_SIZE to the same value will cause data corruption. Read/Write accessible. SRAM_SIZE is set to 000000b during H_RESET and is unaffected by S_RESET or STOP.

BCR26: SRAM Boundary Register

Bit	Name	Description

Note: Bits 7-0 in this register are programmable through the EEPROM.

31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SRAM_BND	SRAM Boundary. Specifies the size of the transmit buffer portion of the SRAM in units of 512-byte pages. For example if SRAM_BND is set to 10, then 5120 bytes of the SRAM will be allocated for the transmit buffer and the rest will be allocated for the receive buffer. The transmit buffer in the SRAM begins at address 0 and ends at the address (SRAM_BND*512)-1. Therefore, the receive buffer always begins on a 512-byte boundary. SRAM_BND must be initialized to an appropriate value, either by the EEPROM or by the host CPU. SRAM_BND must be set to a value less than or equal to IFFCh.

Downloaded from Arrow.com.

Values larger than IFFCh will cause incorrect behavior.

Note: The minimum allowed number of pages is four and the maximum is SRAM_SIZE-4. The Am79C976 controller will not operate correctly with less than four pages of memory per queue. See Table 93 for SRAM_BND programming details.

Table 93. SRAM_BND Programming

SRAM Addresses	SRAM_BND 11:0]
Minimum SRAM_BND Address	004h
Maximum SRAM_BND Address	SRAM_SIZE - 4

***CAUTION**: Programming SRAM_BND and SRAM_SIZE to the same value will cause data corruption.*

Read/Write accessible. SRAM_BND is set to 00000000b during H_RESET and is unaffected by S_RESET or STOP.

BCR27: SRAM Interface Control Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	PTR TST	Reserved. Reserved for manufacturing tests. Written as zero and read as undefined. **Note**: Use of this bit will cause data corruption and erroneous operation. Read/Write accessible. PTR_TST is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit.
14	LOLATRX	Obsolete function. Writing has no effect. Read as undefined.
13-6	RES	Reserved locations. Written as zeros and read as undefined.
5-3	EBCS	Obsolete function. Writing has no effect. Read as undefined.
2-0	CLK_FAC	Obsolete function. Writing has no effect. Read as undefined.

BCR28: Expansion Bus Port Address Lower (Used for Flash/EPROM and SRAM Accesses)

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	EPADDRL	Expansion Port Address Lower. This address is used to provide addresses for the Flash port accesses. Flash accesses are started when a read or write is performed on BCR30. During Flash accesses all bits in EPADDR are valid. Read accessible always; write accessible only when STOP is set or when SRAM_SIZE (BCR25, bits 7-0) is 0. EPADDRL is undefined after H_RESET and is unaffected by S_RESET or STOP.

BCR29: Expansion Port Address Upper (Used for Flash/EPROM Accesses)

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	FLASH	Obsolete function. Read only. Always returns logic 1.
14	LAAINC	Lower Address Auto Increment. When the LAAINC bit is set to 1, the Expansion Port Lower Address will automatically increment by one after a read or write access to EBDATA (BCR30). When EBADDRL reaches FFFFh and LAAINC is set to 1, the Expansion Port Lower Address (EPADDRL) will roll over to 0000h. When the LAAINC bit is set to 0, the Expansion Port Lower Address will not be affected in any way after an access to EBDATA (BCR30) and must be programmed. Read accessible always; write accessible only when the STOP bit is set. LAINC is 0 after H_RESET and is unaffected by S_RESET or the STOP bit.

Downloaded from Arrow.com.

13-8	RES	Reserved locations. Written as zeros and read as undefined.
7-0	EPADDRU	Expansion Port Address Upper. This upper portion of the Expansion Bus address is used to provide addresses for Flash/EPROM port accesses. Read accessible always; write accessible only when the STOP bit is set or when SRAM SIZE (BCR25, bits 7-0) is 0. EPADDRU is undefined after H_RESET and is unaffected by S_RESET or the STOP bit.

BCR30: Expansion Bus Data Port Register

Bit	Name	Description
31-8	RES	Reserved locations. Written as zeros and read as undefined.
7-0	EBDATA	Expansion Bus Data Port. EBDATA is the data port for operations on the Expansion Port involving Flash accesses. Flash read cycles are performed when BCR30 is read. Upon completion of the read cycle, the 8-bit result for Flash access is stored in EBDATA[7:0]. Flash write cycles are performed when BCR30 is written and the FLASH bit (BCR29, bit 15) is set to 1. Read and write accessible. EBDATA is undefined after H_RESET, and is unaffected by S_RESET and the STOP bit.

BCR31: Software Timer Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	STVAL	Software Timer Value. STVAL controls the maximum time for the Software Timer to count before generating the STINT (CSR7, bit 11) interrupt. The Software Timer is a free-running timer that is started upon the first write to STVAL. After the first write, the Software Timer will continually count and set the STINT interrupt at the STVAL period. The STVAL value is interpreted as an unsigned number with a resolution of 10.24µs. For instance, if STVAL is set to 48,828 (0BEBCh), the Software Timer period will be 0.5 s. Setting STVAL to a value of 0 will result in erratic behavior. Read and write accessible. STVAL is set to FFFFh after H_RESET and is unaffected by S_RESET and the STOP bit.

BCR32: MII Control and Status Register

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	ANTST	Reserved for manufacturing tests. Written as 0 and read as undefined. **Note**: Use of this bit will cause data corruption and erroneous operation. Read/Write accessible. ANTST is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit.
14	MIIPD	MII PHY Detect. MIIPD reflects the quiescent state of the MDIO pin. MIIPD is continuously updated whenever there is no management operation in progress on the MII interface. When a management operation begins on the interface, the state of MIIPD is preserved until the operation ends, when the quiescent state is again monitored and continuously updates the MIIPD bit. When the MDIO pin is at a quiescent LOW state, MIIPD is cleared to 0. When the MDIO pin is at a quiescent HIGH state, MIIPD is set to 1. Any transition on the MIIPD bit will set the MIIPDTINT bit (CSR7, bit 1).

Downloaded from Arrow.com.

MIIPD is read only. Write operations are ignored.

13-12 FMDC — Fast Management Data Clock. When FMDC is set to 2h the MII Management Data Clock will run at 10 MHz max. The Management Data Clock will no longer be IEEE 802.3u-compliant and setting this bit should be used with care. The accompanying external PHY must also be able to accept management frames at the new clock rate. When FMDC is set to 1h, the MII Management Data Clock will run at 5 MHz max. The Management Data Clock will no longer be IEEE 802.3u-compliant and setting this bit should be used with care. The accompanying external PHY must also be able to accept management frames at the new clock rate. When FMDC is set to 0h, the MII Management Data Clock will run at 2.5 MHz max and will be fully compliant to IEEE 802.3u standards. See Table 94.

Table 94. FMDC Values

FMDC	Fast Management Data Clock
00	2.5 MHz max
01	5 MHz max
10	10 MHz max
11	Reserved

Read/Write accessible. FMDC is set to 0 during H_RESET, and is unaffected by S_RESET and the STOP bit

11 APEP — MII Auto-Poll External PHY. when APEP is set to 1, the Am79C976 controller will poll the MII status register in the external PHY. This feature allows the software driver or upper layers to see any changes in the status of the external PHY. An interrupt, when enabled, is generated when the contents of the new status is different from the previous status.

Read/Write accessible. APEP is set to 0 during H_RESET and is unaffected by S_RESET and the STOP bit.

10-8 APDW — MII Auto-Poll Dwell Time. APDW determines the dwell time between MII Management Frames accesses when Auto-Poll is turned on. The MDC clock is ERCLK divided by 40 and will be slower than 2.5MHz. The actual interval between polls will be the time shown in Table 95 multiplied by 100 / (ERCLK frequency in MHz).

Table 95. APDW Values

APDW	Auto-Poll™ Dwell Time
000	Continuous (27 µs @ 2.5 MHz)
001	Every 64 MDC cycles (77 µs @ 2.5 MHz)
010	Every 128 MDC cycles (129 µs @ 2.5 MHz)
011	Every 256 MDC cycles (232 µs @ 2.5 MHz)
100	Every 512 MDC cycles (436 µs @ 2.5 MHz)
101	Every 1024 MDC cycles (845 µs @ 2.5 MHz)
110-111	Reserved

Read/Write accessible. APDW is set to 100b after H_RESET and is unaffected by S_RESET and the STOP bit.

7 DISPM — Disable Port Manager. (The corresponding bit in older PCnet family devices is called Disable Auto-Negotiation Auto Setup or DANAS. The name has been changed, but not the function.)

When DISPM is set, the Am79C976 controller after a H_RESET or S_RESET will remain dormant and not automatically start up the Auto-Negotiation section or the enhanced automatic port selection section. Instead, the Am79C976 controller will wait for the software driver to set up the Auto-Negotiation portions of the device. The MII programming in BCR33 and BCR34 is still valid. The Am79C976 controller will not generate any management frames unless Auto-Poll is enabled.

Read/write accessible. DISPM is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit.

Downloaded from Arrow.com.

6	XPHYRST	External PHY Reset. When XPHYRST is set, the Am79C976 controller after an H_RESET or S_RESET will issue an MII management frame that will reset the external PHY. This bit is needed when there is no way to guarantee the state of the external PHY. This bit must be reprogrammed after every H_RESET. Read/Write accessible. XPHYRST is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit. XPHYRST is only valid when the internal Network Port Manager is scanning for a network port.
5	XPHYANE	External PHY Auto-Negotiation Enable. This bit will force the external PHY into enabling Auto-Negotiation. When set to 0 the Am79C976 controller will send a MII management frame disabling Auto-Negotiation. Read/Write accessible. XPHYANE is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit. XPHYANE is only valid when the internal Network Port Manager is scanning for a network port.
4	XPHYFD	External PHY Full Duplex. When set, this bit will force the external PHY into full duplex when Auto-Negotiation is not enabled. Read/Write accessible. XPHYFD is set to 0 by H_RESET, and is unaffected by S_RESET and the STOP bit. XPHYFD is only valid when the internal Network Port Manager is scanning for a network port.
3	XPHYSP	External PHY Speed. When set, this bit will force the external PHY into 100 Mbps mode when Auto-Negotiation is not enabled. Read/Write accessible. XPHYSP is set to 0 by H_RESET, and is unaffected by S_RESET and the STOP bit. XPHYSP is only valid when the internal Network Port Manager is scanning for a network port.
2	RES	Reserved location. Written as zeros and read as undefined.
1	MIIILP	Media Independent Interface Internal Loopback. When set, this bit will cause the internal portion of the MII data port to loop back on itself. The interface is mapped in the following way. The TXD[3:0] nibble data path is looped back onto the RXD[3:0] nibble data path. TX_CLK is looped back as RX_CLK. TX_EN is looped back as RX_DV. CRS is correctly OR'd with TX_EN and RX_DV and always encompasses the transmit frame. TX_ER is looped back as RX_ER. However, TX_ER will not get asserted by the Am79C976 controller to signal an error. The TX_ER function is reserved for future use. Read/Write accessible. MIIILP is set to 0 by H_RESET and is unaffected by S_RESET and the STOP bit.
0	RES	Reserved location. Written as zeros and read as undefined.

BCR33: MII Address Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-5	PHYAD	MII Management Frame PHY Address. PHYAD contains the 5-bit PHY Address field that is used in the management frame that gets clocked out via the MII management port pins (MDC and MDIO) whenever a read or write transaction occurs to BCR34. The PHY address 1Fh is not valid. Read/Write accessible. PHYAD is undefined after H_RESET and is unaffected by S_RESET and the STOP bit.

Downloaded from Arrow.com.

		The PHYAD field is loaded from bits [9:5] of the AUTOPOLL0 register when AUTOPOLL0 is loaded from EEPROM. See "AUTOPOLL0: Auto-Poll0 Register" on page 125
4-0	REGAD	MII Management Frame Register Address. REGAD contains the 5-bit Register Address field that is used in the management frame that gets clocked out via the MII management port pins (MDC and MDIO) whenever a read or write transaction occurs to BCR34. Read/Write accessible. REGAD is undefined after H_RESET and is unaffected by S_RESET and the STOP bit.

BCR34: MII Management Data Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MIIMD	MII Management Data. MIIMD is the data port for operations on the MII management interface (MDIO and MDC). The Am79C976 device builds management frames using the PHYAD and REGAD values from BCR33. The operation code used in each frame is based upon whether a read or write operation has been performed to BCR34. Read cycles on the MII management interface are invoked when BCR34 is read. Upon completion of the read cycle, the 16-bit result of the read operation is stored in MIIMD. Write cycles on the MII management interface are invoked when BCR34 is written. The value written to MIIMD is the value used in the data field of the management write frame. Read/Write accessible. MIIMD is undefined after H_RESET and is unaffected by S_RESET and the STOP bit.

BCR35: PCI Vendor ID Register

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

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	VID	Vendor ID. The PCI Vendor ID register is a 16-bit register that identifies the manufacturer of the Am79C976 controller. AMD's Vendor ID is 1022h. Note that this Vendor ID is not the same as the Manufacturer ID in CSR88 and CSR89. The Vendor ID is assigned by the PCI Special Interest Group. The Vendor ID is not normally programmable, but the Am79C976 controller allows this due to legacy operating systems that do not look at the PCI Subsystem Vendor ID and the Vendor ID to uniquely identify the add-in board or subsystem that the Am79C976 controller is used in. ***Note:*** *If the operating system or the network operating system supports PCI Subsystem Vendor ID and Subsystem ID, use those to identify the add-in board or subsystem and program the VID with the default value of 1022h*. VID is aliased to the PCI configuration space register Vendor ID (offset 00h). VID is read only. Write operations are ignored. VID is set to 1022h by H_RESET and is not affected by S_RESET or by setting the STOP bit.

BCR36: PCI Power Management Capabilities (PMC) Alias Register

This register is an alias of the PMC register located at offset 42h of the PCI Configuration Space. It is included for compatibility with older PCnet devices. Since the PMC register is read only, in older PCnet devices BCR36 provides a means of programming PMC through the EEPROM.

In the Am79C976 controller there is a single PMC register that can be accessed through three different memory spaces. It can be accessed as read-only through PCI Configuration Space (at offset 42h), read-only through BCR36, or read-write through the mem-

Downloaded from Arrow.com.

ory-mapped PMC Alias Register at offset 1B8h. It can be loaded from the EEPROM through offset 1B8h in the Am79C976 controller's memory-mapped I/O space.

For the definition of the bits in this register, refer to the PMC register definition. BCR36 is read only. It is set to 0C802h by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR37: PCI DATA Register Zero (DATA0) Alias Register

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

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

BCR38: PCI DATA Register One (DATA1) Alias Register

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

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D1_SCALE	These bits correspond to the DATA_SCALE field of the PMCSR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. D1_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7-0	DATA1	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. DATA1 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR39: PCI DATA Register Two (DATA2) Alias Register

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

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D2_SCALE	These bits correspond to the DATA_SCALE field of the PMCSR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field. D2_SCALE is read only. Cleared by H_RESET and is not affected

Downloaded from Arrow.com.

by S_RESET or setting the STOP bit.

7-0	DATA2	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. DATA2 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR40: PCI DATA Register Three (DATA3) Alias Register

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

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

BCR41: PCI DATA Register Four (DATA4) Alias Register

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

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

BCR42: PCI DATA Register Five (DATA5) Alias Register

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

Bit	Name	Description
15-10	RES	Reserved locations. Written as zeros and read as undefined.
9-8	D5_SCALE	These bits correspond to the DATA_SCALE field of the PMC-SR (offset Register 44 of the PCI configuration space, bits 14-13). Refer to the description of DATA_SCALE for the meaning of this field.

Downloaded from Arrow.com.

		D5_SCALE is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.
7-0	DATA5	These bits correspond to the PCI DATA register (offset Register 47 of the PCI configuration space, bits 7-0). Refer to the description of DATA register for the meaning of this field. DATA5 is read only. Cleared by H_RESET and is not affected by S_RESET or setting the STOP bit.

BCR43: PCI DATA Register Six (DATA6) Alias Register

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

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

BCR44: PCI DATA Register Seven (DATA7) Alias Register

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

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

BCR45: OnNow Pattern Matching Register 1

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

When BCR45 is written and the PMAT_MODE bit is 0, the Pattern Match logic is disabled and accesses to the PMR are possible. Bits 6-0 of BCR45 specify the address of the PMR word to be accessed. Following the write to BCR45, the PMR word may be read by reading BCR45, BCR46 and BCR47 in any order. To write to PMR word, the write to BCR45 must be followed by a write to BCR46 and a write to BCR47 in that order to

Downloaded from Arrow.com.

complete the operation. The RAM will not actually be written until the write to BCR47 is complete. The write to BCR47 causes all 5 bytes (four bytes of BCR46-47 and the upper byte of the BCR45) to be written to whatever PMR word is addressed by bits 6:0 of BCR45.

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

BCR46: OnNow Pattern Matching Register 2

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

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

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

BCR47: OnNow Pattern Matching Register 3

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

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

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

Downloaded from Arrow.com.

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

Initialization Block

Note: When SSIZE32 (BCR20, bit 8) is set to 0, the software structures are defined to be 16 bits wide. The base address of the initialization block must be aligned to a word boundary, i.e., CSR1, bit 0 must be cleared to 0. When SSIZE32 is set to 0, the initialization block looks like Table 96.

Note: *The Am79C976 controller performs DWord accesses to read the initialization block. This statement is always true, regardless of the setting of the SSIZE32 bit.*

When SSIZE32 (BCR20, bit 8) is set to 1, the software structures are defined to be 32 bits wide. The base address of the initialization block must be aligned to a DWord boundary, i.e., CSR1, bits 1 and 0 must be cleared to 0. When SSIZE32 is set to 1, the initialization block looks like Table 97.

Table 96. Initialization Block (SSIZE32 = 0)

Address	Bits 15-13	Bit 12	Bits 11-8	Bits 7-4	Bits 3-0
IADR+00h	MODE 15-00				
IADR+02h	PADR 15-00				
IADR+04h	PADR 31-16				
IADR+06h	PADR 47-32				
IADR+08h	LADRF 15-00				
IADR+0Ah	LADRF 31-16				
IADR+0Ch	LADRF 47-32				
IADR+0Eh	LADRF 63-48				
IADR+10h	RDRA 15-00				
IADR+12h	RLEN	0	RES	RDRA 23-16	
IADR+14h	TDRA 15-00				
IADR+16h	TLEN	0	RES	TDRA 23-16	

Table 97. Initialization Block (SSIZE32 = 1)

Address	Bits 31-28	Bits 27-24	Bits 23-20	Bits 19-16	Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0
IADR+00h	TLEN	RES	RLEN	RES	MODE			
IADR+04h	PADR 31-00							
IADR+08h	RES				PADR 47-32			
IADR+0Ch	LADRF 31-00							
IADR+10h	LADRF 63-32							
IADR+14h	RDRA 31-00							
IADR+18h	TDRA 31-00							

Downloaded from Arrow.com.

RLEN and TLEN

When SSIZE32 (BCR20, bit 8) is set to 0, the software structures are defined to be 16 bits wide, and the RLEN and TLEN fields in the initialization block are each three bits wide. The values in these fields determine the number of transmit and receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is shown in Table 98. If a value other than those listed in Table 98 is desired, CSR76 and CSR78 can be written after initialization is complete.

When SSIZE32 (BCR20, bit 8) is set to 1, the software structures are defined to be 32 bits wide, and the RLEN and TLEN fields in the initialization block are each 4 bits wide. The values in these fields determine the number of transmit and receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is shown in Table 99.

If a value other than those listed in Table 99 is desired, CSR76 and CSR78 can be written after initialization is complete.

Table 98. R/TLEN Decoding (SSIZE32 = 0)

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

RDRA and TDRA

RDRA and TDRA indicate where the transmit and receive descriptor rings begin. Each DRE must be located at a 16-byte address boundary when SSIZE32 is set to 1 (BCR20, bit 8). Each DRE must be located at an 8-byte address boundary when SSIZE32 is set to 0 (BCR20, bit 8).

Table 99. R/TLEN Decoding (SSIZE32 = 1)

R/TLEN	Number of DREs
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
1X1X	512
11XX	512

LADRF

The Logical Address Filter (LADRF) is a 64-bit mask that is used to accept incoming Logical Addresses. If the first bit in the incoming address (as transmitted on the wire) is a 1, it indicates a logical address. If the first bit is a 0, it is a physical address and is compared against the physical address that was loaded through the initialization block.

A logical address is passed through the CRC generator, producing a 32-bit result. The high order 6 bits of the CRC is used to select one of the 64 bit positions in the Logical Address Filter. If the selected filter bit is set, the address is accepted and the frame is placed into memory.

The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the node's responsibility to determine if the message is actually intended for the node by comparing the destination address of the stored message with a list of acceptable logical addresses.

If the Logical Address Filter is loaded with all zeros and promiscuous mode is disabled, all incoming logical addresses except broadcast will be rejected. If the DRCVBC bit (CSR15, bit 14) is set as well, the broadcast packets will be rejected. See Figure 47.

PADR

This 48-bit value represents the unique node address assigned by the ISO 8802-3 (IEEE/ANSI 802.3) and used for internal address comparison. PADR[0] is compared with the first bit in the destination address of the incoming frame. It must be 0 since only the destination address of a unicast frames is compared to PADR. The six hex-digit nomenclature used by the ISO 8802-3 (IEEE/ANSI 802.3) maps to the Am79C976 PADR register as follows: the first byte is compared with PADR[7:0], with PADR[0] being the least significant bit of the byte. The second ISO 8802-3 (IEEE/ANSI 802.3) byte is compared with PADR[15:8], again from the least significant bit to the most significant bit, and so on. The sixth byte is compared with PADR[47:40], the least significant bit being PADR[40].

Mode

The mode register field of the initialization block is copied into CSR15 and interpreted according to the description of CSR15.

Downloaded from Arrow.com.

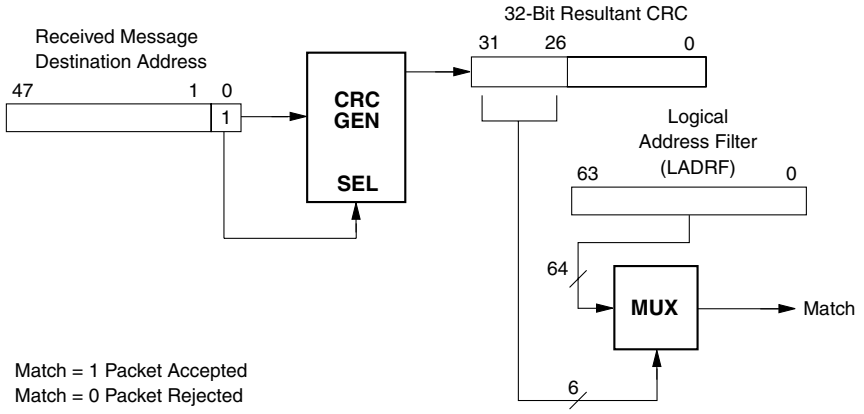


Figure 47. Address Match Logic

Receive Descriptors

When SWSTYLE (BCR20, bits 7-0) is set to 0, then the software structures are defined to be 16 bits wide, and receive descriptors look like Table 100.

When SWSTYLE is set to 2, then the software structures are defined to be 32 bits wide, and receive descriptors look like Table 101.

When SWSTYLE is set to 3, then the software structures are defined to be 32 bits wide, and receive descriptors look like Table 102.

When SWSTYLE is set to 4, then the software structures are defined to be 32 bits wide, and receive descriptors look like Table 103.

When SWSTYLE is set to 5, then the software structures are defined to be 32 bits wide, and receive descriptors look like Table 104. Also, when SWSTYLE is 5, the Am79C976 controller uses 64-bit addressing for software structures that are located above the 32-bit address boundary.

Table 100. Receive Descriptor (SWSTYLE = 0)

Offset	15	14	13	12	11	10	9	8	7-0
00h	RBADR[15:0]								
02h	OWN	ERR	FRAM	OFLO	CRC		STP	ENP	RBADR[23:16]
04h	BCNT								
06h	0	0	0	0	MCNT				

Table 101. Receive Descriptor (SWSTYLE = 2)

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19-16	15-0
00h	RBADR[31:0]													
04h	OWN	ERR	FRAM	OFLO	CRC		STP	ENP		PAM	LAFM	BAM	RES	BCNT
08h	RES	RFRTAG[14:0]												MCNT
0Ch	USER SPACE													

Downloaded from Arrow.com.

Table 102. Receive Descriptor (SWSTYLE = 3)

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19-18	17-16	15-0
00h		RFRTAG[14:0]													MCNT
04h	OWN	ERR	FRAM	OFLO	CRC		STP	ENP		PAM	LAFM	BAM			BCNT
08h	RBADR[31:0]														
0Ch	USER SPACE														

Table 103. Receive Descriptor (SWSTYLE = 4)

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19-18	17-16	15-0
00h	TCI[15:0]														MCNT
04h	OWN	ERR	FRAM	OFLO	CRC		STP	ENP		PAM	LAFM	BAM	TT		BCNT
08h	RBADR[31:0]														
0Ch	USER SPACE														

Table 104. Receive Descriptor (SWSTYLE = 5)

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19-18	17-16	15-0
00h	RFRTAG[31:0]														
04h	TCI[15:0]														MCNT
08h	OWN	ERR	FRAM	OFLO	CRC		STP	ENP		PAM	LAFM	BAM	TT		BCNT
0Ch	RBADR[31:0]														
10h	RBADR[63:32]														
14h	USER SPACE														
18h	USER SPACE														
1Ch	USER SPACE														

Downloaded from Arrow.com.

The following tables describe the bits of the receive descriptors in more detail.

Table 105. Receive Descriptor, SWSTYLE = 0

Offset	Bit	Name	Description
0	15-0	RBADR	Receive Buffer Address. This field contains the address of the receive buffer that is associated with this descriptor.
2	15	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after it has emptied the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after filling the buffer that the description points to. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
	14	ERR	Error Summary. ERR is the OR of FRAM, OFLO, and CRC. ERR is set by the Am79C976 controller and cleared by the host.
	13	FRAM	Framing error indicates that the incoming frame contains a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non- integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is set by the Am79C976 controller and cleared by the host.
	12	OFLO	Overflow error indicates that the receiver has lost all or part of the incoming frame, due to an inability to move data from the receive FIFO into a memory buffer before the internal FIFO overflowed. OFLO is set by the Am79C976 controller and cleared by the host.
	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 set by the Am79C976 controller and cleared by the host. CRC will also be set when Am79C976 receives an RX_ER indication from the external PHY through the MII.
	10		Reserved.
	9	STP	Start of Packet indicates that this is the first buffer used by the Am79C976 controller for this frame. If STP and ENP are both set to 1, the frame fits into a single buffer. Otherwise, the frame is spread over more than one buffer. When LAPPEN (CSR3, bit 5) is cleared to 0, STP is set by the Am79C976 controller and cleared by the host. When LAPPEN is set to 1, STP must be set by the host.
	8	ENP	End of Packet indicates that this is the last buffer used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the Am79C976 controller and cleared by the host.
	7-0	RBADR[23:16]	Receive Buffer Address (high order bits)
4	15-0	BCNT	Buffer Byte Count is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and unchanged by the Am79C976 controller.
6	15-12	ZEROS	This field is reserved. Am79C976 controller will write zeros to these locations.
	11-0	MCNT	Message Byte Count is the number of bytes of the received message written to the receive buffer. This is the actual frame length (including FCS) unless stripping is enabled and the length field is < 46 bytes. In this case, MCNT is 14 + length_field. MCNT can take values in the range 15 to 59 and values greater than or equal to 64. MCNT is expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the Am79C976 controller and cleared by the host.

Downloaded from Arrow.com.

Table 106. Receive Descriptor, SWSTYLE = 2

Offset	Bit	Name	Description
0	31-0	RBADR[31:0]	Receive Buffer Address. This field contains the address of the receive buffer that is associated with this descriptor.
4	31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after it has emptied the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after filling the buffer that the descriptor points to. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
4	30	ERR	Error Summary. ERR is the OR of FRAM, OFLO, and CRC. ERR is set by the Am79C976 controller and cleared by the host.
4	29	FRAM	Framing error indicates that the incoming frame contains a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non-integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is set by the Am79C976 controller and cleared by the host.
4	28	OFLO	Overflow error indicates that the receiver has lost all or part of the incoming frame, due to an inability to move data from the receive FIFO into a memory buffer before the internal FIFO overflowed. OFLO is set by the Am79C976 controller and cleared by the host.
4	27	CRC	CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is set by the Am79C976 controller and cleared by the host. CRC will also be set when Am79C976 controller receives an RX_ER indication from the external PHY through the MII.
4	26		Reserved.
4	25	STP	Start of Packet indicates that this is the first buffer used by the Am79C976 controller for this frame. If STP and ENP are both set to 1, the frame fits into a single buffer. Otherwise, the frame is spread over more than one buffer. When LAPPEN (CSR3, bit 5) is cleared to 0, STP is set by the Am79C976 controller and cleared by the host. When LAPPEN is set to 1, STP must be set by the host.
4	24	ENP	End of Packet indicates that this is the last buffer used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the Am79C976 controller and cleared by the host.
4	23		Reserved.
4	22	PAM	Physical Address Match is set by the Am79C976 controller when it accepts the received frame due to a match of the frame's destination address with the content of the physical address register. PAM is valid only when ENP is set. PAM is set by the Am79C976 controller and cleared by the host. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).

Downloaded from Arrow.com.

Offset	Bit	Name	Description
4	21	LAFM	Logical Address Filter Match is set by the Am79C976 controller when it accepts the received frame based on the value in the logical address filter register. LAFM is valid only when ENP is set. LAFM is set by the Am79C976 controller and cleared by the host. Note that if DRCVBC (CSR15, bit 14) is cleared to 0, only BAM, but not LAFM will be set when a Broadcast frame is received, even if the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter. If DRCVBC is set to 1 and the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter, LAFM will be set on the reception of a Broadcast frame. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
4	20	BAM	Broadcast Address Match is set by the Am79C976 controller when it accepts the received frame, because the frame's destination address is of the type 'Broadcast.' BAM is valid only when ENP is set. BAM is set by the Am79C976 controller and cleared by the host. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
4	19-16		Reserved.
4	15-0	BCNT	Buffer Byte Count is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and unchanged by the Am79C976 controller.
8	31		Reserved.
	30-16	RFRTAG[14:0]	Receive Frame Tag. Indicates the Receive Frame Tag applied from the EADI interface. This field is user defined and has a default value of all zeros. When RXFRTG (CSR7, bit 14) is set to 0, RFRTAG will be read as all zeros. See the section on Receive Frame Tagging for details.
	15-0	MCNT	Message Byte Count is the number of bytes of the received message written to the receive buffer. This is the actual frame length (including FCS) unless stripping is enabled and the length field is < 46 bytes. In this case, MCNT is 14 + length_field. MCNT can take values in the range 15 to 59 and values greater than or equal to 64. MCNT is expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the Am79C976 controller and cleared by the host.
0Ch	31:0	USER SPACE	User Space. Reserved for user defined data.

Downloaded from Arrow.com.

Table 107. Receive Descriptor, SWSTYLE = 3

Offset	Bit	Name	Description
0	31		Reserved
	30-16	RFRTAG[14:0]	Receive Frame Tag. Indicates the Receive Frame Tag applied from the EADI interface. This field is user defined and has a default value of all zeros. When RXFRTG (CSR7, bit 14) is set to 0, RFRTAG will be read as all zeros. See the section on Receive Frame Tagging for details.
	15-0	MCNT	Message Byte Count is the number of bytes of the received message written to the receive buffer. This is the actual frame length (including FCS) unless stripping is enabled and the length field is < 46 bytes. In this case, MCNT is 14 + length_field. MCNT can take values in the range 15 to 59 and values greater than or equal to 64. MCNT is expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the Am79C976 controller and cleared by the host.
4	31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after it has emptied the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after filling the buffer that the descriptor points to. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
4	30	ERR	Error Summary. ERR is the OR of FRAM, OFLO, and CRC. ERR is set by the Am79C976 controller and cleared by the host.
4	29	FRAM	Framing error indicates that the incoming frame contains a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non-integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is set by the Am79C976 controller and cleared by the host.
4	28	OFLO	Overflow error indicates that the receiver has lost all or part of the incoming frame, due to an inability to move data from the receive FIFO into a memory buffer before the internal FIFO overflowed. OFLO is set by the Am79C976 controller and cleared by the host.
4	27	CRC	CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is set by the Am79C976 controller and cleared by the host. CRC will also be set when Am79C976 controller receives an RX_ER indication from the external PHY through the MII.
4	26		Reserved.
4	25	STP	Start of Packet indicates that this is the first buffer used by the Am79C976 controller for this frame. If STP and ENP are both set to 1, the frame fits into a single buffer. Otherwise, the frame is spread over more than one buffer. When LAPPEN (CSR3, bit 5) is cleared to 0, STP is set by the Am79C976 controller and cleared by the host. When LAPPEN is set to 1, STP must be set by the host.
4	24	ENP	End of Packet indicates that this is the last buffer used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the Am79C976 controller and cleared by the host.
4	23		Reserved.

Downloaded from Arrow.com.

Offset	Bit	Name	Description
4	22	PAM	Physical Address Match is set by the Am79C976 controller when it accepts the received frame due to a match of the frame's destination address with the content of the physical address register. PAM is valid only when ENP is set. PAM is set by the Am79C976 controller and cleared by the host. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
4	21	LAFM	Logical Address Filter Match is set by the Am79C976 controller when it accepts the received frame based on the value in the logical address filter register. LAFM is valid only when ENP is set. LAFM is set by the Am79C976 controller and cleared by the host. Note that if DRCVBC (CSR15, bit 14) is cleared to 0, only BAM, but not LAFM will be set when a Broadcast frame is received, even if the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter. If DRCVBC is set to 1 and the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter, LAFM will be set on the reception of a Broadcast frame. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
4	20	BAM	Broadcast Address Match is set by the Am79C976 controller when it accepts the received frame, because the frame's destination address is of the type 'Broadcast.' BAM is valid only when ENP is set. BAM is set by the Am79C976 controller and cleared by the host. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
4	19-16		Reserved.
4	15-0	BCNT	Buffer Byte Count is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and unchanged by the Am79C976 controller.
8	31-0	RBADR[31:0]	Receive Buffer Address. This field contains the address of the receive buffer that is associated with this descriptor.
0Ch	31:0	USER SPACE	User Space. Reserved for user defined data.

Table 108. Receive Descriptor, SWSTYLE = 4

Offset	Bit	Name	Description
0	31-16	TCI[15:0]	VLAN Tag Control Information copied from the received frame.
	15-0	MCNT	Message Byte Count is the number of bytes of the received message written to the receive buffer. This is the actual frame length (including FCS) unless stripping is enabled and the length field is < 46 bytes. In this case, MCNT is 14 + length_field. MCNT can take values in the range 15 to 59 and values greater than or equal to 64. MCNT is expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the Am79C976 controller and cleared by the host.
4	31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after it has emptied the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after filling the buffer that the descriptor points to. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.

Downloaded from Arrow.com.

Offset	Bit	Name	Description
4	30	ERR	Error Summary. ERR is the OR of FRAM, OFLO, and CRC. ERR is set by the Am79C976 controller and cleared by the host.
4	29	FRAM	Framing error indicates that the incoming frame contains a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non-integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is set by the Am79C976 controller and cleared by the host.
4	28	OFLO	Overflow error indicates that the receiver has lost all or part of the incoming frame, due to an inability to move data from the receive FIFO into a memory buffer before the internal FIFO overflowed. OFLO is set by the Am79C976 controller and cleared by the host.
4	27	CRC	CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is set by the Am79C976 controller and cleared by the host. CRC will also be set when Am79C976 controller receives an RX_ER indication from the external PHY through the MII.
4	26		Reserved.
4	25	STP	Start of Packet indicates that this is the first buffer used by the Am79C976 controller for this frame. If STP and ENP are both set to 1, the frame fits into a single buffer. Otherwise, the frame is spread over more than one buffer. When LAPPEN (CSR3, bit 5) is cleared to 0, STP is set by the Am79C976 controller and cleared by the host. When LAPPEN is set to 1, STP must be set by the host.
4	24	ENP	End of Packet indicates that this is the last buffer used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the Am79C976 controller and cleared by the host.
4	23		Reserved.
4	22	PAM	Physical Address Match is set by the Am79C976 controller when it accepts the received frame due to a match of the frame's destination address with the content of the physical address register. PAM is valid only when ENP is set. PAM is set by the Am79C976 controller and cleared by the host. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
4	21	LAFM	Logical Address Filter Match is set by the Am79C976 controller when it accepts the received frame based on the value in the logical address filter register. LAFM is valid only when ENP is set. LAFM is set by the Am79C976 controller and cleared by the host. Note that if DRCVBC (CSR15, bit 14) is cleared to 0, only BAM, but not LAFM will be set when a Broadcast frame is received, even if the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter. If DRCVBC is set to 1 and the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter, LAFM will be set on the reception of a Broadcast frame. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).

Downloaded from Arrow.com.

Offset	Bit	Name	Description
4	20	BAM	Broadcast Address Match is set by the Am79C976 controller when it accepts the received frame, because the frame's destination address is of the type 'Broadcast.' BAM is valid only when ENP is set. BAM is set by the Am79C976 controller and cleared by the host. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
4	19-18	TT[1:0]	VLAN Tag Type. Indicates what type of VLAN tag, if any, is included in the received frame. 00 = Reserved 01 = Frame is Untagged 10 = Frame is Priority-tagged 11 = Frame is VLAN-tagged
4	17-16		Reserved.
4	15-0	BCNT	Buffer Byte Count is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and unchanged by the Am79C976 controller.
8	31-0	RBADR[31:0]	Receive Buffer Address. This field contains the address of the receive buffer that is associated with this descriptor.
0Ch	31:0	USER SPACE	User Space. Reserved for user defined data.

Table 109. Receive Descriptor, SWSTYLE = 5

Offset	Bit	Name	Description
0	31-0	RFRTAG[31:0]	Receive Frame Tag. Indicates the Receive Frame Tag applied from the EADI interface. This field is user defined and has a default value of all zeros. When RXFRTG (CSR7, bit 14) is set to 0, RFRTAG will be read as all zeros. See the section on Receive Frame Tagging for details.
4	31-16	TCI[15:0]	VLAN Tag Control Information copied from the received frame.
	15-0	MCNT	Message Byte Count is the number of bytes of the received message written to the receive buffer. This is the actual frame length (including FCS) unless stripping is enabled and the length field is < 46 bytes. In this case, MCNT is 14 + length_field. MCNT can take values in the range 15 to 59 and values greater than or equal to 64. MCNT is expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the Am79C976 controller and cleared by the host.
8	31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after it has emptied the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after filling the buffer that the descriptor points to. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
8	30	ERR	Error Summary. ERR is the OR of FRAM, OFLO, and CRC. ERR is set by the Am79C976 controller and cleared by the host.
8	29	FRAM	Framing error indicates that the incoming frame contains a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non-integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is set by the Am79C976 controller and cleared by the host.

Downloaded from Arrow.com.

Offset	Bit	Name	Description
8	28	OFLO	Overflow error indicates that the receiver has lost all or part of the incoming frame, due to an inability to move data from the receive FIFO into a memory buffer before the internal FIFO overflowed. OFLO is set by the Am79C976 controller and cleared by the host.
8	27	CRC	CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is set by the Am79C976 controller and cleared by the host. CRC will also be set when Am79C976 controller receives an RX_ER indication from the external PHY through the MII.
8	26		Reserved.
8	25	STP	Start of Packet indicates that this is the first buffer used by the Am79C976 controller for this frame. If STP and ENP are both set to 1, the frame fits into a single buffer. Otherwise, the frame is spread over more than one buffer. When LAPPEN (CSR3, bit 5) is cleared to 0, STP is set by the Am79C976 controller and cleared by the host. When LAPPEN is set to 1, STP must be set by the host.
8	24	ENP	End of Packet indicates that this is the last buffer used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the Am79C976 controller and cleared by the host.
8	23		Reserved.
8	22	PAM	Physical Address Match is set by the Am79C976 controller when it accepts the received frame due to a match of the frame's destination address with the content of the physical address register. PAM is valid only when ENP is set. PAM is set by the Am79C976 controller and cleared by the host. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
8	21	LAFM	Logical Address Filter Match is set by the Am79C976 controller when it accepts the received frame based on the value in the logical address filter register. LAFM is valid only when ENP is set. LAFM is set by the Am79C976 controller and cleared by the host. Note that if DRCVBC (CSR15, bit 14) is cleared to 0, only BAM, but not LAFM will be set when a Broadcast frame is received, even if the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter. If DRCVBC is set to 1 and the Logical Address Filter is programmed in such a way that a Broadcast frame would pass the hash filter, LAFM will be set on the reception of a Broadcast frame. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
8	20	BAM	Broadcast Address Match is set by the Am79C976 controller when it accepts the received frame, because the frame's destination address is of the type 'Broadcast.' BAM is valid only when ENP is set. BAM is set by the Am79C976 controller and cleared by the host. This bit does not exist when the Am79C976 controller is programmed to use 16-bit software structures for the descriptor ring entries (BCR20, bits 7-0, SWSTYLE is cleared to 0).
8	19-18	TT[1:0]	VLAN Tag Type. Indicates what type of VLAN tag, if any, is included in the received frame. 00 = Reserved 01 = Frame is Untagged 10 = Frame is Priority-tagged 11 = Frame is VLAN-tagged
8	17-16		Reserved.

Downloaded from Arrow.com.

Offset	Bit	Name	Description
8	15-0	BCNT	Buffer Byte Count is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and unchanged by the Am79C976 controller.
0Ch	31-0	RBADR[31:0]	Receive Buffer Address. This field contains the low order bits of the address of the receive buffer that is associated with this descriptor.
10h	31-0	RBADR[63:32]	Receive Buffer Address. This field contains the high order bits of the address of the receive buffer that is associated with this descriptor.
14h	31:0	USER SPACE	User Space. Reserved for user defined data.
18h	31:0	USER SPACE	User Space. Reserved for user defined data.
1Ch	31:0	USER SPACE	User Space. Reserved for user defined data.

Downloaded from Arrow.com.

Transmit Descriptors

When SWSTYLE (BCR20, bits 7-0) is set to 0, the software structures are defined to be 16 bits wide, and transmit descriptors look like Table 110.

When SWSTYLE is set to 2, the software structures are defined to be 32 bits wide, and transmit descriptors look like Table 111.

When SWSTYLE is set to 3, then the software structures are defined to be 32 bits wide, and transmit descriptors look like Table 112.

When SWSTYLE is set to 4, then the software structures are defined to be 32 bits wide, and transmit descriptors look like Table 113.

When SWSTYLE is set to 5, then the software structures are defined to be 32 bits wide, and transmit descriptors look like Table 114. Also when SWSTYLE is 5, the Am79C976 controller uses 64-bit addressing for software structures that are located above the 32-bit address boundary.

Table 110. Transmit Descriptor (SWSTYLE = 0)

Offset	15	14	13	12	11	10	9	8	7-0
00h	TBADR[15:0]								
02h	OWN		ADD_ FCS	LTINT			STP	ENP	TBADR[23:16]
04h	BCNT								
06h	Reserved								

Table 111. Transmit Descriptor (SWSTYLE = 2)

Offset	31	30	29	28	27	26	25	24	23	22-16	15-12	11-4	3-0
00h	TBADR[31:0]												
04h	OWN		ADD_ FCS	LTINT			STP	ENP			BCNT		
08h	Reserved												
0Ch	USER SPACE												

Table 112. Transmit Descriptor (SWSTYLE = 3)

Offset	31	30	29	28	27	26	25	24	23	22-16	15-0
00h	Reserved										
04h	OWN		ADD_ FCS	LTINT			STP	ENP			BCNT
08h	TBADR[31:0]										
0Ch	USER SPACE										

Table 113. Transmit Descriptor (SWSTYLE = 4)

Offset	31	30	29	28	27-26	25	24	23	22	21-18	17-16	15-0
00h	OWN		ADD_ FCS	LTINT		STP	ENP		KILL			BCNT
04h											TCC[1:0]	TCI[15:0]
08h	TBADR[31:0]											
0Ch	USER SPACE											

Downloaded from Arrow.com.

Table 114. Transmit Descriptor (SWSTYLE = 5)

Offset	31	30	29	28	27-26	25	24	23	22	21-18	17-16	15-0
00h	OWN		ADD_ FCS	LTINT		STP	ENP		KILL			BCNT
04h											TCC[1:0]	TCI[15:0]
08h	TBADR[31:0]											
0Ch	TBADR[63:32]											
10h	Reserved											
14h	USER SPACE											
18h	USER SPACE											
1Ch	USER SPACE											

The following tables describe the transmit descriptor bits in more detail.

Table 115. Transmit Descriptor, SWSTYLE = 0

Offset	Bit	Name	Description
0	15-0	TBADR[15:0]	Transmit Buffer Address. This field contains the high order bits of the address of the Transmit buffer that is associated with this descriptor.
2	15	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after transmitting the contents of the buffer. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
2	14		Reserved location.
2	13	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. This bit should be set with the ENP bit. However, for backward compatibility, it is recommended that this bit be set for every descriptor of the intended frame. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS is cleared to 0, FCS generation is controlled by DXMTFCS. When APAD_XMT (CSR4, bit 11) is set to 1, the setting of ADD_FCS has no effect. ADD_FCS is set by the host, and is not changed by the Am79C976 controller. This is a reserved bit in the C-LANCE (Am79C90) controller.
2	12	LTINT	Last Transmit Interrupt. When enabled by the LTINTEN bit (CSR5, bit 14), LTINT is used to suppress interrupts after selected frames have been copied to the transmit FIFO. When LTINT is cleared to 0 and ENP is set to 1, the Am79C976 controller will not set TINT (CSR0, bit 9) after the corresponding frame has been copied to the transmit FIFO. TINT will only be set when the last descriptor of a frame has both LTINT and ENP set to 1. When LTINTEN is cleared to 0, the LTINT bit is ignored.
2	11-10		Reserved.
2	9	STP	Start of Packet indicates that this is the first buffer to be used by the Am79C976 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 Am79C976 controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is set by the host and is not changed by the Am79C976 controller.
2	8	ENP	End of Packet indicates that this is the last buffer to be used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the host and is not changed by the Am79C976 controller.

Downloaded from Arrow.com.

Offset	Bit	Name	Description
2	7-0	TBADR[23:16]	Transmit Buffer Address (high order bits)
4	15-0	BCNT	Buffer Byte Count is the usable length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the Am79C976 controller. This field is written by the host and is not changed by the Am79C976 controller. There are no minimum buffer size restrictions.
6	15-0		Reserved.

Table 116. Transmit Descriptor, SWSTYLE = 2

Offset	Bit	Name	Description
0	31-0	TBADR[31:0]	Transmit Buffer Address. This field contains the address of the Transmit buffer that is associated with this descriptor.
4	31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after transmitting the contents of the buffer. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
4	30		Reserved location.
4	29	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. This bit should be set with the ENP bit. However, for backward compatibility, it is recommended that this bit be set for every descriptor of the intended frame. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS is cleared to 0, FCS generation is controlled by DXMTFCS. When APAD_XMT (CSR4, bit 11) is set to 1, the setting of ADD_FCS has no effect. ADD_FCS is set by the host, and is not changed by the Am79C976 controller. This is a reserved bit in the C-LANCE (Am79C90) controller.
4	28	LTINT	Last Transmit Interrupt. When enabled by the LTINTEN bit (CSR5, bit 14), LTINT is used to suppress interrupts after selected frames have been copied to the transmit FIFO. When LTINT is cleared to 0 and ENP is set to 1, the Am79C976 controller will not set TINT (CSR0, bit 9) after the corresponding frame has been copied to the transmit FIFO. TINT will only be set when the last descriptor of a frame has both LTINT and ENP set to 1. When LTINTEN is cleared to 0, the LTINT bit is ignored.
4	27-26		Reserved.
4	25	STP	Start of Packet indicates that this is the first buffer to be used by the Am79C976 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 Am79C976 controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is set by the host and is not changed by the Am79C976 controller.
4	24	ENP	End of Packet indicates that this is the last buffer to be used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the host and is not changed by the Am79C976 controller.
4	23-16		Reserved.
4	15-0	BCNT	Buffer Byte Count is the usable length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the Am79C976 controller. This field is written by the host and is not changed by the Am79C976 controller. There are no minimum buffer size restrictions.
8	31-0		Reserved
0Ch	31-0	USER SPACE	User Space. Reserved for user defined data.

Downloaded from Arrow.com.

Table 117. Transmit Descriptor, SWSTYLE = 3

Offset	Bit	Name	Description
0	31-0		Reserved.
4	31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after transmitting the contents of the buffer. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
4	30		Reserved location.
4	29	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. This bit should be set with the ENP bit. However, for backward compatibility, it is recommended that this bit be set for every descriptor of the intended frame. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS is cleared to 0, FCS generation is controlled by DXMTFCS. When APAD_XMT (CSR4, bit 11) is set to 1, the setting of ADD_FCS has no effect. ADD_FCS is set by the host, and is not changed by the Am79C976 controller. This is a reserved bit in the C-LANCE (Am79C90) controller.
4	28	LTINT	Last Transmit Interrupt. When enabled by the LTINTEN bit (CSR5, bit 14), LTINT is used to suppress interrupts after selected frames have been copied to the transmit FIFO. When LTINT is cleared to 0 and ENP is set to 1, the Am79C976 controller will not set TINT (CSR0, bit 9) after the corresponding frame has been copied to the transmit FIFO. TINT will only be set when the last descriptor of a frame has both LTINT and ENP set to 1. When LTINTEN is cleared to 0, the LTINT bit is ignored.
4	27-26		Reserved.
4	25	STP	Start of Packet indicates that this is the first buffer to be used by the Am79C976 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 Am79C976 controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is set by the host and is not changed by the Am79C976 controller.
4	24	ENP	End of Packet indicates that this is the last buffer to be used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the host and is not changed by the Am79C976 controller.
4	23-16		Reserved.
4	15-0	BCNT	Buffer Byte Count is the usable length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the Am79C976 controller. This field is written by the host and is not changed by the Am79C976 controller. There are no minimum buffer size restrictions.
8	31-0	TBADR[31:0]	Transmit Buffer Address. This field contains the address of the Transmit buffer that is associated with this descriptor.
0Ch	31-0	USER SPACE	User Space. Reserved for user defined data.

Downloaded from Arrow.com.

Table 118. Transmit Descriptor, SWSTYLE = 4

Offset	Bit	Name	Description
0	31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after transmitting the contents of the buffer. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
0	30		Reserved location.
0	29	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. This bit should be set with the ENP bit. However, for backward compatibility, it is recommended that this bit be set for every descriptor of the intended frame. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS is cleared to 0, FCS generation is controlled by DXMTFCS. When APAD_XMT (CSR4, bit 11) is set to 1, the setting of ADD_FCS has no effect. ADD_FCS is set by the host, and is not changed by the Am79C976 controller. This is a reserved bit in the C-LANCE (Am79C90) controller.
0	28	LTINT	Last Transmit Interrupt. When enabled by the LTINTEN bit (CSR5, bit 14), LTINT is used to suppress interrupts after selected frames have been copied to the transmit FIFO. When LTINT is cleared to 0 and ENP is set to 1, the Am79C976 controller will not set TINT (CSR0, bit 9) after the corresponding frame has been copied to the transmit FIFO. TINT will only be set when the last descriptor of a frame has both LTINT and ENP set to 1. When LTINTEN is cleared to 0, the LTINT bit is ignored.
0	27-26		Reserved.
0	25	STP	Start of Packet indicates that this is the first buffer to be used by the Am79C976 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 Am79C976 controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is set by the host and is not changed by the Am79C976 controller.
0	24	ENP	End of Packet indicates that this is the last buffer to be used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the host and is not changed by the Am79C976 controller.
0	23		Reserved.
0	22	KILL	This bit causes the transmission of the corresponding frame to be aborted. If the transmitter has not started sending the frame at the time that the descriptor processing logic encounters the KILL bit, no portion of the frame will be sent. If part of the frame has been sent, the frame will be truncated, and an FCS field containing the inverse of the correct CRC will be appended.
0	21-16		Reserved.
0	15-0	BCNT	Buffer Byte Count is the usable length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the Am79C976 controller. This field is written by the host and is not changed by the Am79C976 controller. There are no minimum buffer size restrictions.
4	31-18		Reserved.
4	17-16	TCC[1:0]	VLAN Tag Control Command. This field contains a command that causes the transmitter to add, modify, or delete a VLAN tag or to transmit the frame unaltered. 00 = Transmit the data in the buffer unaltered 01 = Delete the VLAN tag (the 13th through 16th bytes of the frame) 10 = Insert a VLAN tag containing the TCI field from the descriptor 11 = Replace the TCI field of the frame with TCI data from the descriptor

Downloaded from Arrow.com.

Offset	Bit	Name	Description
4	15-0	TCI[15:0]	Tag Control Information. If the contents of the TCC field is 10 or 11, the controller will transmit the contents of the TCI field as bytes 15 and 16 of the outgoing frame.
8	31-0	TBADR[31:0]	Transmit Buffer Address. This field contains the address of the Transmit buffer that is associated with this descriptor.
0Ch	31-0	USER SPACE	User Space. Reserved for user defined data.

Table 119. Transmit Descriptor, SWSTYLE = 5

Offset	Bit	Name	Description
0	31	OWN	This bit indicates whether the descriptor entry is owned by the host (OWN = 0) or by the Am79C976 controller (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The Am79C976 controller clears the OWN bit after transmitting the contents of the buffer. Both the Am79C976 controller and the host must not alter a descriptor entry after it has relinquished ownership.
0	30		Reserved location.
0	29	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. This bit should be set with the ENP bit. However, for backward compatibility, it is recommended that this bit be set for every descriptor of the intended frame. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS is cleared to 0, FCS generation is controlled by DXMTFCS. When APAD_XMT (CSR4, bit 11) is set to 1, the setting of ADD_FCS has no effect. ADD_FCS is set by the host, and is not changed by the Am79C976 controller. This is a reserved bit in the C-LANCE (Am79C90) controller.
0	28	LTINT	Last Transmit Interrupt. When enabled by the LTINTEN bit (CSR5, bit 14), LTINT is used to suppress interrupts after selected frames have been copied to the transmit FIFO. When LTINT is cleared to 0 and ENP is set to 1, the Am79C976 controller will not set TINT (CSR0, bit 9) after the corresponding frame has been copied to the transmit FIFO. TINT will only be set when the last descriptor of a frame has both LTINT and ENP set to 1. When LTINTEN is cleared to 0, the LTINT bit is ignored.
0	27-26		Reserved.
0	25	STP	Start of Packet indicates that this is the first buffer to be used by the Am79C976 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 Am79C976 controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is set by the host and is not changed by the Am79C976 controller.
0	24	ENP	End of Packet indicates that this is the last buffer to be used by the Am79C976 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the host and is not changed by the Am79C976 controller.
0	23		Reserved.
0	22	KILL	This bit causes the transmission of the corresponding frame to be aborted. If the transmitter has not started sending the frame at the time that the descriptor processing logic encounters the KILL bit, no portion of the frame will be sent. If part of the frame has been sent, the frame will be truncated, and an FCS field containing the inverse of the correct CRC will be appended.
0	21-16		Reserved.

Downloaded from Arrow.com.

Offset	Bit	Name	Description
0	15-0	BCNT	Buffer Byte Count is the usable length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the Am79C976 controller. This field is written by the host and is not changed by the Am79C976 controller. There are no minimum buffer size restrictions.
4	31-18		Reserved.
4	17-16	TCC[1:0]	VLAN Tag Control Command. This field contains a command that causes the transmitter to add, modify, or delete a VLAN tag or to transmit the frame unaltered. 00 = Transmit the data in the buffer unaltered 01 = Delete the VLAN tag (the 13th through 16th bytes of the frame) 10 = Insert a VLAN tag containing the TCI field from the descriptor 11 = Replace the TCI field of the frame with TCI data from the descriptor
4	15-0	TCI[15:0]	Tag Control Information. If the contents of the TCC field are 10 or 11, the controller will transmit the contents of the TCI field as bytes 15 and 16 of the outgoing frame.
8	31-0	TBADR[31:0]	Transmit Buffer Address. This field contains the low order bits of the address of the Transmit buffer that is associated with this descriptor.
0Ch	31-0	TBADR[63:32]	Transmit Buffer Address. This field contains the high order bits of the address of the Transmit buffer that is associated with this descriptor.
10h	31-0		Reserved.
14h	31-0	USER SPACE	User Space. Reserved for user defined data.
18h	31-0	USER SPACE	User Space. Reserved for user defined data.
1Ch	31-0	USER SPACE	User Space. Reserved for user defined data.

Downloaded from Arrow.com.

REGISTER SUMMARY

PCI Configuration Registers

Offset	Name	Width in Bit	Access Mode	Default Value
00h	PCI Vendor ID	16	RO	1022h
02h	PCI Device ID	16	RO	2000h
04h	PCI Command	16	RW	0000h
06h	PCI Status	16	RW	0290h
08h	PCI Revision ID	8	RO	5Xh
09h	PCI Programming IF	8	RO	00h
0Ah	PCI Sub-Class	8	RO	00h
0Bh	PCI Base-Class	8	RO	02h
0Ch	PCI Cache Line Size	8	R/W	00h
0Dh	PCI Latency Timer	8	RW	00h
0Eh	PCI Header Type	8	RO	00h
0Fh	Reserved	8	RO	00h
10h	PCI I/O Base Address	32	RW	0000 0001h
14h	PCI Memory Mapped I/O Base Address	32	RW	0000 0000h
18h - 2Bh	Reserved	8	RO	00h
2Ch	PCI Subsystem Vendor ID	16	RO	00h
2Eh	PCI Subsystem ID	16	RO	00h
30h	PCI Expansion ROM Base Address	32	RW	0000 0000h
31h - 33h	Reserved	8	RO	00h
34h	Capabilities Pointer	8	RO	44h
35h-3Bh	Reserved	8	RO	00h
3Ch	PCI Interrupt Line	8	RW	00h
3Dh	PCI Interrupt Pin	8	RO	01h
3Eh	PCI MIN_GNT	8	RO	18h
3Fh	PCI MAX_LAT	8	RO	18h
44h	PCI Capability Identifier	8	RO	01h
45h	PCI Next Item Pointer	8	RO	00h
46h	PCI Power Management Capabilities	16	RO	C802h
48h	PCI Power Management Control/Status	16	RO	00h
4Ah	PCI PMCSR Bridge Support Extensions	8	RO	00h
4Bh	PCI Data	8	RO	00h
4Ch - FFh	Reserved	8	RO	00h

***Note**: RO = read only, RW = read/write*

Downloaded from Arrow.com.

Memory-Mapped Registers

Register	Offset, hex	Width, bits	Reset Value	Reset Type	Notes
AP_VALUE0	0A8	16	0	H	Auto-poll register 0 value
AP_VALUE1	0AA	16	0	H	Auto-poll register 1 value
AP_VALUE2	0AC	16	0	H	Auto-poll register 2 value
AP_VALUE3	0AE	16	0	H	Auto-poll register 3 value
AP_VALUE4	0B0	16	0	H	Auto-poll register 4 value
AP_VALUE5	0B2	16	0	H	Auto-poll register 5 value
AUTOPOLL0	88	16	8100	E,H	Auto-poll Register 1
AUTOPOLL1	8A	16	0	E,H	Auto-poll Register 2
AUTOPOLL2	8C	16	0	E,H	Auto-poll Register 3
AUTOPOLL3	8E	16	0	E,H	Auto-poll Register 4
AUTOPOLL4	90	16	0	E,H	Auto-poll Register 5
AUTOPOLL5	92	16	0	E,H	Auto-poll Register 6
BADR	120	64	0	H	Base Address of Receive Descriptor Ring
BADX	100	64	0	H	Base Address of Transmit Descriptor Ring
CHIPID	0F0	32	X262 8003	H	Chip ID. Read only
CHPOLLTIME	18A	16	0	E,H	Chain Poll Timer Register
CMD0	48	32	0	E,H	Command 0
CMD2	50	32	0	E,H	Command 2
CMD3	54	32	0	E,H	Command 3
CMD7	64	32	0	P	Command 7
CTRL0	68	32	0000 0900	E,H	Control 0
CTRL1	6C	32	0001 0001	E,H	Control 1
CTRL2	70	32	0000 0004	E,H	Control 2
CTRL3	74	32	0	H	Control 3
DATAMBIST	1A0	64	0	H	MBIST Access Register
EEPROM_ACC	17C	16	0	H	EEPROM Access Register
FLASH_ADDR	198	32	0	H	Flash Address Register
FLASH_DATA	19C	16	0	H	Flash Data Register
FLOW	0C8	32	0	E,H	Flow Control Register
IFS1	18C	8	3C	E,H	Inter-Frame Spacing, Part 1 Register
INT0	38	32	0		Interrupt 0. Read only or write 1 to clear. All bits cleared by H_RESET. Bits 0, 5, 6, & 8 also cleared when RUN is cleared.
INTEN0	40	32	0	E,H	Interrupt Enable 0.
IPG	18D	8	60	E,H	Inter-Packet Gap Register
LADRF	168	64	0	H	Logical Address Filter Register

Downloaded from Arrow.com.

Register	Offset, hex	Width, bits	Reset Value	Reset Type	Notes
LED0	0E0	16	00C0	E,H	LED 0 Control
LED1	0E2	16	0094	E,H	LED 1 Control
LED2	0E4	16	1080	E,H	LED 2 Control
LED3	0E6	16	0081	E,H	LED 3 Control
MAX_LAT_A	1B1	8	18h	E,H	PCI Maximum Latency Shadow Register
MIN_GNT_A	1B0	8	18h	E,H	PCI Minimum Grant Shadow Register
PADR	160	48	0	H	Physical Address Register
PAUSE_CNT	0DE	32	0	H	Pause count. Read only.
PCIDATA0	1BC	16	0	E,H	PCI Data Register 0 Alias Register
PCIDATA1	1BE	16	0	E,H	PCI Data Register 1 Alias Register
PCIDATA2	1C0	16	0	E,H	PCI Data Register 2 Alias Register
PCIDATA3	1C2	16	0	E,H	PCI Data Register 3 Alias Register
PCIDATA4	1C4	16	0	E,H	PCI Data Register 4 Alias Register
PCIDATA5	1C6	16	0	E,H	PCI Data Register 5 Alias Register
PCIDATA6	1C8	16	0	E,H	PCI Data Register 6 Alias Register
PCIDATA7	1CA	16	0	E,H	PCI Data Register 7 Alias Register
PHY_ACCESS	0D0	32	0	H	PHY Access
PMAT0	190	32	0	H	OnNow Pattern Register 0
PMAT1	194	16	0	H	OnNow Pattern Register 1
PMC_A	1B8	16	c802h	E,H	PCI Power Management Capabilities Shadow Register
RCV_PROTECT	0DC	16	40	H	Receive Protect Register.
RCV_RING_LEN	150	16	0	H	Receive Descriptor Ring Length. Two's-complement.
ROM_CFG	18E	16	0	H	ROMBASE Configuration Register
SID_A	1B4	16	0	E,H	PCI Subsystem ID Shadow Register
SRAM_BND	17A	16	0	E,H	SSRAM Boundary
SRAM_SIZE	178	16	0	E,H	SSRAM Size
STAT0	30	32	0		Status 0. Read only or write 1 to clear. Bits 12-10 are reset by POR. All others are reset by H_RESET.
STVAL	0D8	16	FFFF	H	Software Timer Value
SVID_A	1B6	16	0	E,H	Subsystem Vendor ID Shadow Register
VID_A	1B2	16	1022	E,H	PCI Vendor ID Shadow Register
XMT_RING_LEN	140	16	0	H	Transmit Descriptor Ring Length. Two's-complement.
XMTPOLLTIME	188	16	0	E,H	Transmit Polling Interval Register

Note: *H = H_RESET, E = EE_RESET, P = Power on Reset*

Downloaded from Arrow.com.

Control and Status Registers

RAP Addr	Symbol	Default Value	Comments	Use
00	CSR0	uuuu 0004	Am79C976 Controller Status and Control Register	R
01	CSR1	uuuu uuuu	Lower IADR: maps to location 16	S
02	CSR2	uuuu uuuu	Upper IADR: maps to location 17	S
03	CSR3	uuuu 0600	Interrupt Masks and Deferral Control	S
04	CSR4	uuuu 0004	Test and Features Control	R
05	CSR5	uuuu 0000	Extended Control and Interrupt 1	R
06	CSR6	uuuu uuuu	Reserved	
07	CSR7	0uuu 0000	Extended Control and Interrupt 2	R
08	CSR8	uuuu uuuu	LADRF0: Logical Address Filter — LADRF[15:0]	S
09	CSR9	uuuu uuuu	LADRF1: Logical Address Filter — LADRF[31:16]	S
10	CSR10	uuuu uuuu	LADRF2: Logical Address Filter — LADRF[47:32]	S
11	CSR11	uuuu uuuu	LADRF3: Logical Address Filter — LADRF[63:48]	S
12	CSR12	uuuu uuuu	PADR0: Physical Address Register — PADR[15:0]	S
13	CSR13	uuuu uuuu	PADR1: Physical Address Register — PADR[31:16]	S
14	CSR14	uuuu uuuu	PADR2: Physical Address Register — PADR[47:32]	S
15	CSR15	see register description	MODE: Mode Register	S
16	CSR16	uuuu uuuu	Reserved	
17	CSR17	uuuu uuuu	Reserved	
18	CSR18	uuuu uuuu	Reserved	
19	CSR19	uuuu uuuu	Reserved	
20	CSR20	uuuu uuuu	Reserved	
21	CSR21	uuuu uuuu	Reserved	
22	CSR22	uuuu uuuu	Reserved	
23	CSR23	uuuu uuuu	Reserved	
24	CSR24	uuuu uuuu	BADRL: Base Address of RCV Ring Lower	S
25	CSR25	uuuu uuuu	BADRU: Base Address of RCV Ring Upper	S
26	CSR26	uuuu uuuu	Reserved	
27	CSR27	uuuu uuuu	Reserved	
28	CSR28	uuuu uuuu	Reserved	
29	CSR29	uuuu uuuu	Reserved	
30	CSR30	uuuu uuuu	BADXL: Base Address of XMT Ring Lower	S
31	CSR31	uuuu uuuu	BADXU: Base Address of XMT Ring Upper	S
32	CSR32	uuuu uuuu	Reserved	
33	CSR33	uuuu uuuu	Reserved	

Downloaded from Arrow.com.

RAP Addr	Symbol	Default Value	Comments	Use
34	CSR34	uuuu uuuu	Reserved	
35	CSR35	uuuu uuuu	Reserved	
36	CSR36	uuuu uuuu	Reserved	
37	CSR37	uuuu uuuu	Reserved	
38	CSR38	uuuu uuuu	Reserved	
39	CSR39	uuuu uuuu	Reserved	
40	CSR40	uuuu uuuu	Reserved	
41	CSR41	uuuu uuuu	Reserved	
42	CSR42	uuuu uuuu	Reserved	
43	CSR43	uuuu uuuu	Reserved	
44	CSR44	uuuu uuuu	Reserved	
45	CSR45	uuuu uuuu	Reserved	
46	CSR46	uuuu uuuu	Reserved	
47	CSR47	uuuu uuuu	TXPOLLINT: Transmit Polling Interval	S
48	CSR48	uuuu uuuu	Reserved	
49	CSR49	uuuu uuuu	CHPOLLINT: Chain Polling Interval	
50	CSR50	uuuu uuuu	Reserved	
51	CSR51	uuuu uuuu	Reserved	
52	CSR52	uuuu uuuu	Reserved	
53	CSR53	uuuu uuuu	Reserved	
54	CSR54	uuuu uuuu	Reserved	
55	CSR55	uuuu uuuu	Reserved	
56	CSR56	uuuu uuuu	Reserved	
57	CSR57	uuuu uuuu	Reserved	
58	CSR58	see register description	SWS: Software Style	S
59	CSR59	uuuu uuuu	Reserved	T
60	CSR60	uuuu uuuu	Reserved	
61	CSR61	uuuu uuuu	Reserved	
62	CSR62	uuuu uuuu	Reserved	
63	CSR63	uuuu uuuu	Reserved	
64	CSR64	uuuu uuuu	Reserved	
65	CSR65	uuuu uuuu	Reserved	
66	CSR66	uuuu uuuu	Reserved	
67	CSR67	uuuu uuuu	Reserved	
68	CSR68	uuuu uuuu	Reserved	
69	CSR69	uuuu uuuu	Reserved	

Downloaded from Arrow.com.

RAP Addr	Symbol	Default Value	Comments	Use
70	CSR70	uuuu uuuu	Reserved	
71	CSR71	uuuu uuuu	Reserved	
72	CSR72	uuuu uuuu	Reserved	
73	CSR73	uuuu uuuu	Reserved	
74	CSR74	uuuu uuuu	Reserved	
75	CSR75	uuuu uuuu	Reserved	
76	CSR76	uuuu uuuu	RCVRL: RCV Ring Length	S
77	CSR77	uuuu uuuu	Reserved	
78	CSR78	uuuu uuuu	XMTRL: XMT Ring Length	S
79	CSR79	uuuu uuuu	Reserved	
80	CSR80	uuuu 1400	DMATCFW: DMA Transfer Counter and FIFO Threshold Control	S
81	CSR81	uuuu uuuu	Reserved	
82	CSR82	uuuu uuuu	Reserved	
83	CSR83	uuuu uuuu	Reserved	
84	CSR84	uuuu uuuu	Reserved	
85	CSR85	uuuu uuuu	Reserved	
86	CSR86	uuuu uuuu	Reserved	
87	CSR87	uuuu uuuu	Reserved	
88	CSR88	X262 8003	Chip ID Register Lower	T
89	CSR89	uuuu X262	Chip ID Register Upper	T
90	CSR90	uuuu uuuu	Reserved	
91	CSR91	uuuu uuuu	Reserved	
92	CSR92	uuuu uuuu	Reserved	
93	CSR93	uuuu uuuu	Reserved	
94	CSR94	uuuu uuuu	Reserved	
95	CSR95	uuuu uuuu	Reserved	
96	CSR96	uuuu uuuu	Reserved	
97	CSR97	uuuu uuuu	Reserved	
98	CSR98	uuuu uuuu	Reserved	
99	CSR99	uuuu uuuu	Reserved	
100	CSR100	uuuu uuuu	Bus Timeout	0
101	CSR101	uuuu uuuu	Reserved	
102	CSR102	uuuu uuuu	Reserved	
103	CSR103	uuuu 0000	Reserved	
104	CSR104	uuuu uuuu	Reserved	
105	CSR105	uuuu uuuu	Reserved	
106	CSR106	uuuu uuuu	Reserved	

Downloaded from Arrow.com.

RAP Addr	Symbol	Default Value	Comments	Use
107	CSR107	uuuu uuuu	Reserved	
108	CSR108	uuuu uuuu	Reserved	
109	CSR109	uuuu uuuu	Reserved	
110	CSR110	uuuu uuuu	Reserved	
111	CSR111	uuuu uuuu	Reserved	
112	CSR112	uuuu uuuu	Missed Frame Count	O
113	CSR113	uuuu uuuu	Reserved	
114	CSR114	uuuu uuuu	Receive Collision Count	O
115	CSR115	uuuu uuuu	Reserved	
116	CSR116	0000 0000	OnNow Power Mode Register	S
117	CSR117	uuuu uuuu	Reserved	
118	CSR118	uuuu uuuu	Reserved	
119	CSR119	uuuu 0000	Reserved	
120	CSR120	uuuu uuuu	Reserved	
121	CSR121	uuuu uuuu	Reserved	
122	CSR122	uuuu uuuu	Reserved	
123	CSR123	uuuu uuuu	Reserved	
124	CSR124	uuuu 0000	Test Register 1	T
125	CSR125	uuuu 603C	MAC Enhanced Configuration Control	T
126	CSR126	uuuu uuuu	Reserved	
127	CSR127	uuuu uuuu	Reserved	

Note:
u = undefined value, R = Running register, S = Setup register, T = Test register; all default values are in hexadecimal format.

O = Obsolete Register

Downloaded from Arrow.com.

Bus Configuration Registers

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

RAP	Mnemonic	Default	Name	Programmability	
				User	EEPROM
0	MSRDA	0005h	Reserved	No	No
1	MSWRA	0005h	Reserved	No	No
2	MC	0000h	Miscellaneous Configuration	Yes	Yes
3	Reserved	N/A	Reserved	No	No
4	LED0	00C0h	LED0 Status	Yes	Yes
5	LED1	0094h	LED1 Status	Yes	Yes
6	LED2	1080h	LED2 Status	Yes	Yes
7	LED3	0081h	LED3 Status	Yes	Yes
8	Reserved	N/A	Reserved	No	No
9	FDC	0004h	Full-Duplex Control	Yes	Yes
10-15	Reserved	N/A	Reserved	No	No
16	IOBASEL	N/A	Reserved	No	No
17	IOBASEU	N/A	Reserved	No	No
18	BSBC	9000h	Burst and Bus Control	Yes	Yes
19	EECAS	0000h	EEPROM Control and Status	Yes	No
20	SWS	0000h	Software Style	Yes	No
21	Reserved	N/A	Reserved	No	No
22	PCILAT	1818h	PCI Latency	Yes	Yes
23	PCISID	0000h	PCI Subsystem ID	No	Yes
24	PCISVID	0000h	PCI Subsystem Vendor ID	No	Yes
25	SRAMSIZE	0000h	SRAM Size	Yes	Yes
26	SRAMBND	0000h	SRAM Boundary	Yes	Yes
27	Reserved	N/A	Reserved	No	No
28	EBADDRL	N/A	Expansion Bus Address Lower	Yes	No
29	EBADDRU	N/A	Expansion Bus Address Upper	Yes	No
30	EBDATA	N/A	Expansion Bus Data Port	Yes	No
31	STVAL	FFFFh	Software Timer Value	Yes	No
32	MIICAS	0400h	MII Control and Status	Yes	Yes
33	MIIADDR	N/A	MII Address	Yes	Yes
34	MIIMDR	N/A	MII Management Data	Yes	No
35	PCIVID	1022h	PCI Vendor ID	No	Yes
36	PMC_A	C802h	PCI Power Management Capabilities (PMC) Alias Register	No	Yes
37	DATA0	0000h	PCI DATA Register Zero Alias Register	No	Yes
38	DATA1	0000h	PCI DATA Register One Alias Register	No	Yes
39	DATA2	0000h	PCI DATA Register Two Alias Register	No	Yes
40	DATA3	0000h	PCI DATA Register Three Alias Register	No	Yes
41	DATA4	0000h	PCI DATA Register Four Alias Register	No	Yes
42	DATA5	0000h	PCI DATA Register Five Alias Register	No	Yes
43	DATA6	0000h	PCI DATA Register Six Alias Register	No	Yes
44	DATA7	0000h	PCI DATA Register Seven Alias Register	No	Yes
45	PMR1	N/A	Pattern Matching Register 1	Yes	No
46	PMR2	N/A	Pattern Matching Register 2	Yes	No
47	PMR3	N/A	Pattern Matching Register 3	Yes	No

Downloaded from Arrow.com.

REGISTER BIT CROSS REFERENCE

Table 120 shows the location and default value for each programmable bit or field. The Offset column gives the offset in hexadecimal of the register that contains the bit. The codes in the Reset Type column have the following meanings:

H	Hardware Reset. The bit is set to its default value when the RST pin is asserted (low).
E	EEPROM Reset. The bit is set to its default value when before the EEPROM is read or after an EEPROM read error is detected
P	Power On Reset. The bit is set to its default value when power is first applied to the device.

In many cases, a particular bit can be accessed through more than one register. For these bits, the columns, “Alternate Reg.”, “Bit Num”, and “Alternate Bit Name," give the alternate access paths.

Table 120. Register Bit Cross Reference

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
ANTST	TEST0	8	1A8	0	E,H	BCR32	15	ANTST
AP_PHY0_ ADDR	AUTOPOLL0	4:0	88	0	E,H			
AP_PHY1_ ADDR	AUTOPOLL1	4:0	8A	0	E,H			
AP_PHY1_ DFLT	AUTOPOLL1	5	8A	0	E,H			
AP_PHY2_ ADDR	AUTOPOLL2	4:0	8C	0	E,H			
AP_PHY2_ DFLT	AUTOPOLL2	5	8C	0	E,H			
AP_PHY3_ ADDR	AUTOPOLL3	4:0	8E	0	E,H			
AP_PHY3_ DFLT	AUTOPOLL3	5	8E	0	E,H			
AP_PHY4_ ADDR	AUTOPOLL4	4:0	90	0	E,H			
AP_PHY4_ DFLT	AUTOPOLL4	5	90	0	E,H			
AP_PHY5_ ADDR	AUTOPOLL5	4:0	92	0	E,H			
AP_PHY5_ DFLT	AUTOPOLL5	5	92	0	E,H			
AP_PRE_ SUP1	AUTOPOLL1	6	8A	0	E,H			
AP_PRE_ SUP2	AUTOPOLL2	6	8C	0	E,H			

Downloaded from Arrow.com.

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
AP_PRE_SUP3	AUTOPOLL3	6	8E	0	E,H			
AP_PRE_SUP4	AUTOPOLL4	6	90	0	E,H			
AP_PRE_SUP5	AUTOPOLL5	6	92	0	E,H			
AP_REG0_ADDR	AUTOPOLL0	12:8	88	1	E,H			
AP_REG0_EN	AUTOPOLL0	15	88	1	E,H			
AP_REG1_ADDR	AUTOPOLL1	12:8	8A	0	E,H			
AP_REG1_EN	AUTOPOLL1	15	8A	0	E,H			
AP_REG2_ADDR	AUTOPOLL2	12:8	8C	0	E,H			
AP_REG2_EN	AUTOPOLL2	15	8C	0	E,H			
AP_REG3_ADDR	AUTOPOLL3	12:8	8E	0	E,H			
AP_REG3_EN	AUTOPOLL3	15	8E	0	E,H			
AP_REG4_ADDR	AUTOPOLL4	12:8	90	0	E,H			
AP_REG4_EN	AUTOPOLL4	15	90	0	E,H			
AP_REG5_ADDR	AUTOPOLL5	12:8	92	0	E,H			
AP_REG5_EN	AUTOPOLL5	15	92	0	E,H			
AP_VALUE0	AP_VALUE0	15:0	0A8	0	H			
AP_VALUE1	AP_VALUE1	15:0	0AA	0	H			
AP_VALUE2	AP_VALUE2	15:0	0AC	0	H			
AP_VALUE3	AP_VALUE3	15:0	0AE	0	H			
AP_VALUE4	AP_VALUE4	15:0	0B0	0	H			
AP_VALUE5	AP_VALUE5	15:0	0B2	0	H			
APAD_XMT	CMD2	6	50	0	E,H	CSR4	11	APAD_XMT
APDW	CTRL2	2:0	70	4	E,H	BCR32	10:8	APDW
APEP	CMD3	24	54	0	E,H	BCR32	11	APEP
APINT0	INT0	20	38	0	H			
APINT0EN	INTEN0	20	40	0	E,H			
APINT1	INT0	21	38	0	H			
APINT1EN	INTEN0	21	40	0	E,H			
APINT2	INT0	22	38	0	H			
APINT2EN	INTEN0	22	40	0	E,H			

Downloaded from Arrow.com.

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
APINT3	INT0	24	38	0	H			
APINT3EN	INTEN0	24	40	0	E,H			
APINT4	INT0	25	38	0	H			
APINT4EN	INTEN0	25	40	0	E,H			
APINT5	INT0	26	38	0	H			
APINT5EN	INTEN0	26	40	0	E,H			
APROMWE	CMD2	27	50	0	E,H	BCR2	8	APROMWE
ASTRP_RCV	CMD2	13	50	0	E,H	CSR4	10	ASTRP_RCV
AUTONEG_ COMPLETE	STAT0	4	30	0	H			
BADR	BADR	63:32	120	0	H			
BADR	BADR	31:16	120	0	H	CSR25	15:0	BADRU
BADR	BADR	15:0	120	0	H	CSR24	15:0	BADRL
BADX	BADX	63:32	100	0	H			
BADX	BADX	31:16	100	0	H	CSR31	15:0	BADXU
BADX	BADX	15:0	100	0	H	CSR30	15:0	BADXL
BFD_SCALE_ DOWN	TEST0	9	1A8	0	E,H			
BSWP	CTRL0	24	68	0	E,H	CSR3	2	BSWP
BURST_ALIGN	CTRL0	4	68	0	E,H			
BURST_LIMIT	CTRL0	3:0	68	0	E,H			
CBIO_EN	TEST0	17	1A8	0	E,H			
CHDPOLL	CMD2	1	50	0	E,H	CSR7	12	CHDPOLL
CHIPID	CHIPID	31:0	0F0	X262 8003h	H			
CHPOLLTIME	CHPOLLTIME	15:0	18A	0	E,H	CSR49	15:0	CHPOLLINT
DISABLE_MWI	CMD3	27	54	0	E,H			
DIS_ READ_WAIT	CMD3	29	54	0	E,H			
DIS_ WRITE_WAIT	CMD3	28	54	0	E,H			
DISPM	CMD3	14	54	0	E,H	BCR32	7	DISPM
DM_DIR	DATAMBIST	56	1A0	0	H			
DM_DATA	DATAMBIST	31:0	1A0	0	H			
DM_BACKG	DATAMBIST	53:52	1A0	0	H			
DM_ADDR	DATAMBIST	51:32	1A0	0	H			
DM_DONE	DATAMBIST	63	1A0	0	H			
DM_ERROR	DATAMBIST	62	1A0	0	H			

Downloaded from Arrow.com.

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
DM_FAIL_ STATE	DATAMBIST	55:54	1A0	0	H			
DM_FAIL_ STOP	DATAMBIST	59	1A0	0	H			
DM_TEST_ FAIL	DATAMBIST	58	1A0	0	H			
DM_RESUME	DATAMBIST	60	1A0	0	H			
DM_START	DATAMBIST	61	1A0	0	H			
DRCVBC	CMD2	17	50	0	E,H	CSR15	14	DRCVBC
DRCVPA	CMD2	18	50	0	E,H	CSR15	13	DRCVPA
DRTY	CMD2	5	50	0	E,H	CSR15	5	DRTY
DWIO	CMD2	28	50	0	E,H	BCR18	7	DWIO
DXMT2PD	CMD2	10	50	0	E,H	CSR3	4	DXMT2PD
DXMTFCS	CMD2	8	50	0	E,H	CSR15	3	DXMTFCS
ECS	EEPROM_ ACC	2	17C	0	H	BCR19	2	ECS
EDI/EDO	EEPROM_ ACC	0	17C	0	H	BCR19	0	EDI/EDO
EEBUSY_T	TEST0	13	1A8	0	E,H			
EEDET	EEPROM_ ACC	13	17C	0	H	BCR19	13	EEDET
EEN	EEPROM_ ACC	4	17C	0	H	BCR19	4	EEN
ESK	EEPROM_ ACC	1	17C	0	H	BCR19	1	ESK
EMBA	CMD2	11	50	0	E,H	CSR3	3	EMBA
EXLOOP	CMD2	3	50	0	E,H			
FCCMD	FLOW_ CONTROL	16	0C8	0	E,H			
FCOLL	CMD2	12	50	0	E,H	CSR15	4	FCOLL
FCPEN	FLOW_ CONTROL	17	0C8	0	E,H			
FDRPA	CMD2	20	50	0	E,H	BCR9	2	FDRPA
FIXP	FLOW_ CONTROL	18	0C8	0	E,H			
FL_ADDR	FLASH_ ADDR	23:0	198	0	H	BCR28 BCR29	15:0 7:0	EPADDRL EPADDRU
FL_DATA	FLASH_DATA	7:0	19C	0	H	BCR30	7:0	EBDATA[7:0]
FMDC	CTRL2	9:8	70	0	E,H	BCR32	13:12	FMDC
FORCE_FD	CMD3	12	54	0	E,H	BCR9	0	FDEN

Downloaded from Arrow.com.

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
FORCE_LS	CMD3	11	54	0	E,H			
FORCE_FS	CTRL2	18:16	70	0	E,H			
FPA	FLOW_ CONTROL	20	0C8	0	E,H			
FULL_DUPLEX	STAT0	6	30	0	H			
IFS1	IFS1	7:0	18C	3CH	E,H	CSR125	7:0	IFS1
INIT_MIB	CMD3	25	54	0	E,H			
INLOOP	CMD2	4	50	0	E,H			
INTLEVEL	CMD3	13	54	0	E,H	BCR2	7	INTLEVEL
INTR	INT0	31	38	0	H	CSR0	7	INTR
INTREN	CMD0	1	48	0	E,H			
IPG	IPG	7:0	18D	60H	E,H	CSR125	15:8	IPG
JUMBO	CMD3	21	54	0	E,H			
LAAINC	FLASH_ ADDR	31	198	0	H	BCR29	14	LAAINC
LADRF	LADRF	63:48	168	0	H	CSR11	15:0	LADRF3
LADRF	LADRF	47:32	168	0	H	CSR10	15:0	LADRF2
LADRF	LADRF	31:16	168	0	H	CSR9	15:0	LADRF1
LADRF	LADRF	15:0	168	0	H	CSR8	15:0	LADRF0
LAPPEN	CMD2	2	50	0	E,H	CSR3	5	LAPPEN
LC_DET	STAT0	10	30	0	P	CSR116	9	LCDET
LCINT	INT0	27	38	0	H			
LCINTEN	INTEN0	27	40	0	E,H			
LCMODE_EE	CMD3	5	54	0	E,H	CSR116	8	LCMODE
LCMODE_SW	CMD7	0	64	0	P			
LED0	LED0	15:0	0E0	00C0h	E,H	BCR4	15:0	LED0
LED1	LED1	15:0	0E2	0094h	E,H	BCR5	15:0	LED1
LED2	LED2	15:0	0E4	1080h	E,H	BCR6	15:0	LED2
LED3	LED3	15:0	0E6	0081h	E,H	BCR7	15:0	LED3
LEDCNTTST	TEST0	5	1A8	0	E,H			
LEDPE	CMD2	29	50	0	E,H	BCR2	12	LEDPE
LINK_STAT	STAT0	5	30	0	H			
LTINTEN	CMD2	9	50	0	E,H	CSR5	14	LTINTEN
MAX_LAT_A	MAX_LAT	7:0	1B1	18H	E,H	BCR22	15:8	MAX_LAT
MCCIINT	INT0	18	38	0	H	CSR7	3	MCCIINT
MCCIINTEN	INTEN0	18	40	0	E,H	CSR7	2	MCCIINTE

Downloaded from Arrow.com.

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
MCCINT	INT0	17	38	0	H	CSR7	5	MCCINT
MCCINTEN	INTEN0	17	40	0	E,H	CSR7	4	MCCINTE
MFSM_RESET	TEST0	10	1A8	0	E,H			
MIIPD	STAT0	3	30	0	H	BCR32	14	MIIPD
MIIPDTINT	INT0	19	38	0	H	CSR7	1	MIIPDTINT
MIIPDTINTEN	INTEN0	19	40	0	E,H	CSR7	0	MIIPDTINTE
MIN_GNT_A	MIN_GNT	7:0	1B0	18H	E,H	BCR22	7:0	MIN_GNT
MP_DET	STAT0	11	30	0	P	CSR116	5	MPMAT
MPEN_EE	CMD3	6	54	0	E,H			
MPEN_SW	CMD7	1	64	0	P			
MPINT	INT0	13	38	0	H	CSR5	4	MPINT
MPINTEN	INTEN0	13	40	0	E,H	CSR5	3	MPINTE
MPPEN_EE	CMD3	8	54	0	E,H	CSR116	4	MPPEN
MPPEN_SW	CMD7	2	64	0	P			
MPPLBA	CMD3	9	54	0	E,H	CSR5	5	MPPLBA
MREINT	INT0	16	38	0	H	CSR7	9	MREINT
MREINTEN	INTEN0	16	40	0	E,H	CSR7	8	MREINTE
NOUFLO	CMD2	30	50	0	E,H	BCR18	11	NOUFLO
NPA	FLOW_ CONTROL	19	0C8	0	E,H			
PADR	PADR	47:32	160	0	H	CSR14	15:0	PADR2
PADR	PADR	31:16	160	0	H	CSR13	15:0	PADR1
PADR	PADR	15:0	160	0	H	CSR12	15:0	PADR0
PAUSE_CNT	PAUSE_CNT	15:0	0DE	0	H			
PAUSE_LEN	FLOW_ CONTROL	15:0	0C8	0	E,H			
PAUSE_PEND	STAT0	14	30	0	H			
PAUSING	STAT0	13	30	0	H			
PCIDATA0	PCIDATA0	9:0	1BC	0	E,H	BCR37	9:0	PCIDATA0
PCIDATA1	PCIDATA1	9:0	1BE	0	E,H	BCR38	9:0	PCIDATA1
PCIDATA2	PCIDATA2	9:0	1C0	0	E,H	BCR39	9:0	PCIDATA2
PCIDATA3	PCIDATA3	9:0	1C2	0	E,H	BCR40	9:0	PCIDATA3
PCIDATA4	PCIDATA4	9:0	1C4	0	E,H	BCR41	9:0	PCIDATA4
PCIDATA5	PCIDATA5	9:0	1C6	0	E,H	BCR42	9:0	PCIDATA5
PCIDATA6	PCIDATA6	9:0	1C8	0	E,H	BCR43	9:0	PCIDATA6
PCIDATA7	PCIDATA7	9:0	1CA	0	E,H	BCR44	9:0	PCIDATA7

Downloaded from Arrow.com.

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
PHY_ADDR	PHY_ ACCESS	25:21	0D0	0	H			
PHY_BLK_RD_ CMD	PHY_ ACCESS	29	0D0	0	H			
PHY_CMD_ DONE	PHY_ ACCESS	31	0D0	0	H			
PHY_DATA	PHY_ ACCESS	15:0	0D0	0	H			
PHY_NBLK_ RD_CMD	PHY_ ACCESS	28	0D0	0	H			
PHY_PRE_ SUP	PHY_ ACCESS	27	0D0	0	H			
PHY_REG_ ADDR	PHY_ ACCESS	20:16	0D0	0	H			
RST_POL	CMD3	0	54	0	E,H	CSR116	0	RST_POL
PHY_WR_ CMD	PHY_ ACCESS	30	0D0	0	H			
PMAT_DET	STAT0	12	30	0	P	CSR116	7	PMAT
PMAT_MODE	CMD7	3	64	0	P	BCR45	7	PMAT_MODE
PMC	PMC	15:0	1B8	C802H	E,H	BCR36	15:0	PMC
PME_EN_OVR	CMD3	4	54	0	E,H	CSR116	10	PME_EN_OV R
PMR_ADDR	PMAT0	6:0	190	0	H	BCR45	6:0	PMR_ADDR
PMR_B0	PMAT0	15:8	190	0	H	BCR45	15:8	PMR_B0
PMR_B1	PMAT0	23:16	190	0	H	BCR46	7:0	PMR_B1
PMR_B2	PMAT0	31:24	190	0	H	BCR46	15:8	PMR_B2
PMR_B3	PMAT1	7:0	194	0	H	BCR47	7:0	PMR_B3
PMR_B4	PMAT1	15:8	194	0	H	BCR47	15:8	PMR_B4
PREAD	EEPROM_ ACC	14	17C	0	H	BCR19	14	PREAD
PREFETCH_ DIS	CMD3	30	54	0	E,H			
PROM	CMD2	16	50	0	E,H	CSR15	15	PROM
PVALID	EEPROM_ ACC	15	17C	0	H	BCR19	15	PVALID
RCV_ PROTECT	RCV_ PROTECT	15:0	0DC	64	H			
RCV_RING_ LEN	RCV_RING_ LEN	15:0	150	0	H	CSR76	15:0	RCVRL
RCVFW	CTRL1	1:0	6C	1	E,H	CSR80	13:12	RCVFW[1:0]
RDMD	CMD0	12	48	0	E,H	CSR7	13	RDMD

Downloaded from Arrow.com.

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
RST_PHY	CMD3	26	54	0	E,H			
REX_RTRY	CMD3	18	54	0	E,H			
REX_UFLO	CMD3	17	54	0	E,H			
RINT	INT0	0	38	0	H,R	CSR0	10	RINT
RINTEN	INTEN0	0	40	0	E,H	CSR3	10	RINTM
ROMBASE[0]	ROM_CFG	0	18E	0	H			
ROMBASE [23:11]	ROM_CFG	15:3	18E	0	H			
ROMTMG	CTRL0	11:8	68	9	E,H	BCR18	15:12	ROMTMG
RPA	CMD2	19	50	0	E,H	CSR124	3	RPA
RTRY_LCOL	CMD3	16	54	0	E,H			
RTYTST_ BUMP	TEST0	4	1A8	0	E,H			
RTYTST_OUT	TEST0	3	1A8	0	E,H			
RTYTST_ RANGEN	TEST0	2	1A8	0	E,H			
RTYTST_SLOT	TEST0	1	1A8	0	E,H			
RUN	CMD0	0	48	0	E,H			
RUNNING	STAT0	0	30	0	H			
RWU_DRIVER	CMD3	3	54	0	E,H	CSR116	3	RWU_DRIVER
RWU_GATE	CMD3	2	54	0	E,H	CSR116	2	RWU_GATE
RWU_POL	CMD3	1	54	0	E,H	CSR116	1	RWU_POL
RX_FAST_ SPND	CMD0	5	48	0	E,H			
RX_SPND	CMD0	3	48	0	E,H			
RX_ SUSPENDED	STAT0	2	30	0	H			
RXFRTGEN	CMD3	10	54	0	E,H			
SERRLEVEL	TEST0	0	1A8	0	E,H			
SID	SID	15:0	1B4	0	E,H	BCR24	15:0	SID
SINT	INT0	12	38	0	H	CSR5	11	SINT
SINTEN	INTEN0	12	40	0	E,H	CSR5	10	SINTE
SLOTMOD	CTRL1	25:24	6C	0	E,H			
SPEED	STAT0	9:7	30	0	H			
SPNDINT	INT0	14	38	0	H			
SPNDINTEN	INTEN0	14	40	0	E,H			
SRAM_BND	SRAM_BND	15:0	17A	0	E,H	BCR26	7:0	SRAM_BND
SRAM_SIZE	SRAM_SIZE	15:0	178	0	E,H	BCR25	7:0	SRAM_SIZE

Downloaded from Arrow.com.

Bit Name	Register	Bit Num	Offset (hex)	Default Value	Reset Type	Alternate		
						Register	Bit Num	Bit Name
SRAM_TYPE	CTRL0	17:16	68	?	E,H			
STINT	INT0	4	38	0	H	CSR7	11	STINT
STINTEN	INTEN0	4	40	0	E,H	CSR7	10	STINTE
STVAL	STVAL	15:0	0D8	FFFFH	H	BCR31	15:0	STVAL
SVID	SVID	15:0	1B6	0	E,H	BCR23	15:0	SVID
SWSTYLE	CTRL3	7:0	74	0	H	CSR58	7:0	SWSTYLE
TDMD	CMD0	8	48	0	E,H	CSR0	3	TDMD
TINT	INT0	8	38	0	H,R	CSR0	9	TINT
TINTEN	INTEN0	8	40	0	E,H	CSR3	9	TINTM
TSEL	TEST0	11	1A8	0	E,H			
TX_FAST_ SPND	CMD0	4	48	0	E,H			
TX_SPND	CMD0	2	48	0	E,H			
TX_ SUSPENDED	STAT0	1	30	0	H			
TXDNINT	INT0	6	38	0	H,R			
TXDNINTEN	INTEN0	6	40	0	E,H	CSR5	12	TXDNINTEN
TXDPOLL	CMD2	0	50	0	E,H	CSR4	12	TXDPOLL
TXSTRTINT	INT0	5	38	0	H,R	CSR4	3	TXSTRT
TXSTRTINTEN	INTEN0	5	40	0	E,H	CSR4	2	TXSTRTM
UINT	INT0	7	38	0	H	CSR4	6	UINT
UINTCMD	CMD0	6	48	0	E,H	CSR4	7	UINTCMD
VID	VID	15:0	1B2	1022H	E,H	BCR35	15:0	VID
VLONLY	CMD3	19	54	0	E,H			
VSIZE	CMD3	20	54	0	E,H			
XMT_RING_ LEN	XMT_RING_ LEN	15:0	140	0	H	CSR78	15:0	XMTRL
XMTFW	CTRL1	9:8	6C	0	E,H	CSR80	9:8	XMTFW[1:0]
XMTPOLLTIME	XMTPOLL TIME	15:0	188	0	E,H	CSR47	15:0	TXPOLLINT
XMTSP	CTRL1	17:16	6C	1	E,H	CSR80	11:10	XMTSP[1:0]
XPHYANE	CTRL2	5	70	0	E,H	BCR32	5	XPHYANE
XPHYFD	CTRL2	4	70	0	E,H	BCR32	4	XPHYFD
XPHYRST	CTRL2	6	70	0	E,H	BCR32	6	XPHYRST
XPHYSP	CTRL2	3	70	0	E,H	BCR32	3	XPHYSP

Downloaded from Arrow.com.

REGISTER PROGRAMMING SUMMARY

Programmable Registers

Table 121. Control and Status Registers

Register	Contents			
CSR0	Status and control bits: (DEFAULT = 0004)			
	8000 -- 4000 -- 2000 -- 1000 --	0800 -- 0400 RINT 0200 TINT 0100 IDON	0080 INTR 0040 IENA 0020 RXON 0010 TXON	0008 TDMD 0004 STOP 0002 STRT 0001 INIT
CSR1	Lower IADR			
CSR2	Upper IADR			
CSR3	Interrupt masks and Deferral Control: (DEFAULT = 0600)			
	8000 -- 4000 -- 2000 -- 1000 --	0800 0400 RINTM 0200 TINTM 0100 IDONM	0080 -- 0040 0020 LAPPEN 0010 DXMT2PD	0008 EMBA 0004 BSWP 0002 -- 0001 --
CSR4	Interrupt masks, configuration and status bits: (DEFAULT = 0004)			
	8000 -- 4000 -- 2000 -- 1000 TXDPOLL	0800 APAD_XMT 0400 ASTRP_RCV 0200 0100	0080 UINTCMD 0040 UINT 0020 0010	0008 TXSTRT 0004 TXSTRTM 0002 -- 0001 --
CSR5	Extended Interrupt masks, configuration and status bits: (DEFAULT = 0XXX)			
	8000 4000 LTINTEN 2000 TXDNINT 1000 TXDNINTEN	0800 SINT 0400 SINTE 0200 -- 0100 --	0080 0040 0020 MPPLBA 0010 MPINT	0008 MPINTE 0004 MPEN 0002 MPMODE 0001 SPND
CSR7	Extended Interrupt masks, configuration and status bits: (DEFAULT = 0000)			
	8000 FASTSPNDE 4000 2000 RDMD 1000 CHDPOLL	0800 STINT 0400 STINTE 0200 MREINT 0100 MREINTE	0080 MAPINT 0040 MAPINTE 0020 MCCINT 0010 MCCINTE	0008 MCCIINT 0004 MCCIINTE 0002 MIIPDTINT 0001 MIIPDTNTE
CSR8 - CSR11	Logical Address Filter			
CSR12 - CSR14	Physical Address Register			
CSR15	MODE: (DEFAULT = 0)			
	8000 PROM 4000 DRCVBC 2000 DRCVPA 1000 --	0800 -- 0400 -- 0200 -- 0100 --	0080 -- 0040 0020 DRTY 0010 FCOLL	0008 DXMTFCS 0004 LOOP 0002 DTX 0001 DRX
CSR47	TXPOLLINT: Transmit Polling Interval			
CSR49	RXPOLLINT: Chain Polling Interval			
CSR58	Software Style (mapped to BCR20) bits [7:0] = SWSTYLE, Software Style Register.			

Downloaded from Arrow.com.

Register	Contents			
	8000--	0800 --	0080 SWSTYLE	0008 SWSTYLE
	4000--	0400 --	0040 SWSTYLE	0004 SWSTYLE
	2000--	0200 --	0020 SWSTYLE	0002 SWSTYLE
	1000--	0100 SSIZE32	0010 SWSTYLE	0001 SWSTYLE
CSR76	RCVRL: RCV Descriptor Ring length			
CSR78	XMTRL: XMT Descriptor Ring length			
CSR80	FIFO threshold and DMA burst control (DEFAULT = 1400)			
	8000 Reserved 4000 Reserved bits [13:12] = RCVFW, Receive FIFO Watermark 0000 Request DMA when 48 bytes are present 1000 Request DMA when 64 bytes are present 2000 Request DMA when 128 bytes are present 3000 Reserved bits [11:10] = XMTSP, Transmit Start Point 0000 Start transmission after 16 bytes have been written 0400 Start transmission after 64 bytes have been written 0800 Start transmission after 128 bytes have been written 0C00 Start transmission after the full packet has been written bits [9:8] = XMTFW, Transmit FIFO Watermark 0000 Start DMA when 16 write cycles can be made 0100 Start DMA when 64 write cycles can be made 0200 Start DMA when 128 write cycles can be made 0300 Start DMA when 256 write cycles can be made bits [7:0] = Reserved			
CSR88~89	Chip ID (Contents = v2628003; v = Version Number)			
CSR116	OnNow Power Mode Register			
	8000 --	0800 --	0080 PMAT	0008 RWU_DRIVER
	4000 --	0400 PM_EN_OVR	0040 EMPPLBA	0004 RWU_GATE
	2000 --	0200 LCDET	0020 MPMAT	0002 RWU_POL
	1000 --	0100 LCMODE	0010 MPPEN	0001 RST_POL
CSR124	BMU Test Register (DEFAULT = 0000)			
	8000 --	0800 --	0080 --	0008 RPA
	4000 --	0400 --	0040 --	0004 --
	2000 --	0200 --	0020 --	0002 --
	1000 --	0100 --	0010 --	0001 --
CSR125	MAC Enhanced Configuration Control (DEFAULT = 603c bits [15:8] = IPG, InterPacket Gap (Default=60xx, 96 bit times) bits [8:0] = IFS1, InterFrame Space Part 1 (Default=xx3c, 60 bit times)			

Downloaded from Arrow.com.

Table 122. Bus Configuration Registers

RAP Addr	Register	Contents			
2	MC	Miscellaneous Configuration bits: (DEFAULT = 0)			
		8000 --	0800 --	0080 INTLEVEL	0008 --
		4000 --	0400 --	0040 --	0004 --
		2000 --	0200 --	0020 --	0002 --
		1000 LEDPE	0100 APROMWE	0010 --	0001 --
4	LED0	Programs the function and width of the LED0 signal. (DEFAULT = 00C0)			
		8000 LEDOUT	0800 --	0080 PSE	0008 --
		4000 LEDPOL	0400 --	0040 LNKSE	0004 RCVE
		2000 LEDDIS	0200 MPSE	0020 RCVME	0002 SFBDE
		1000 100E	0100 FDLSE	0010 XMTE	0001 COLE
5	LED1	Programs the function and width of the LED1 signal. (DEFAULT = 0094)			
		8000 LEDOUT	0800 --	0080 PSE	0008 --
		4000 LEDPOL	0400 --	0040 LNKSE	0004 RCVE
		2000 LEDDIS	0200 MPSE	0020 RCVME	0002 SFBDE
		1000 100E	0100 FDLSE	0010 XMTE	0001 COLE
6	LED2	Programs the function and width of the LED2 signal. (DEFAULT = 1080)			
		8000 LEDOUT	0800 --	0080 PSE	0008 --
		4000 LEDPOL	0400 --	0040 LNKSE	0004 RCVE
		2000 LEDDIS	0200 MPSE	0020 RCVME	0002 SFBDE
		1000 100E	0100 FDLSE	0010 XMTE	0001 COLE
7	LED3	Programs the function and width of the LED3 signal. (DEFAULT = 0081)			
		8000 LEDOUT	0800 --	0080 PSE	0008 --
		4000 LEDPOL	0400 --	0040 LNKSE	0004 RCVE
		2000 LEDDIS	0200 MPSE	0020 RCVME	0002 SFBDE
		1000 100E	0100 FDLSE	0010 XMTE	0001 COLE
9	FDC	Full-Duplex Control. (DEFAULT= 0004)			
		8000 --	0800 --	0080 --	0008 --
		4000 --	0400 --	0040 --	0004 FDRPAD
		2000 --	0200 --	0020 --	0002 --
		1000 --	0100 --	0010 --	0001 FDEN
18	BSBC	Burst Size and Bus Control (DEFAULT = 9000)			
		8000 ROMTMG3	0800 NOUFLO	0080 DWIO	0008 --
		4000 ROMTMG2	0400 --	0040	0004 --
		2000 ROMTMG1	0200	0020	0002 --
		1000 ROMTMG0	0100	0010 --	0001 --
19	EECAS	EEPROM Control and Status (DEFAULT = 0000)			
		8000 PVALID	0800 --	0080 --	0008 --
		4000 PREAD	0400 --	0040 --	0004 ECS
		2000 EEDET	0200 --	0020 --	0002 ESK
		1000 --	0100 --	0010 EEN	0001 EDI/EDO
20	SWSTYLE	Software Style (DEFAULT = 0000, maps to CSR 58)			
22	PCILAT	PCI Latency (DEFAULT = 1818)			
		bits [15:8] = MAX_LAT bits [7:0] = MIN_GNT			
25	SRAM_SIZE	SRAM Size (DEFAULT = 0000)			

Downloaded from Arrow.com.

RAP Addr	Register	Contents
		bits [15:0] = SRAM_SIZE
26	SRAM_BND	SRAM Boundary (DEFAULT = 0000)
		bits [15:0] = SRAM_BND
28	EPADDRL	Expansion Port Address Lower
29	EPADDRU	Expansion Port Address Upper
		8000 4000 LAINC 2000 -- 1000 -- 0800 0400 -- 0200 -- 0100 -- 0080 EPADDRU7 0040 EPADDRU6 0020 EPADDRU5 0010 EPADDRU4 0008 EPADDRU3 0004 EPADDRU2 0002 EPADDRU1 0001 EPADDRU0
30	EBDATA	Expansion Bus Data Port
31	STVAL	Software Timer Interrupt Value (DEFAULT = FFFF)
32	MIICAS	MII Status and Control (DEFAULT = 0400)
		8000 *ANTST* 4000 MIIPD 2000 FMDC1 1000 FMDC0 0800 APEP 0400 APDW2 0200 APDW1 0100 APDW0 0080 DISPM 0040 XPHYRST 0020 XPHYANE 0010 XPHYFD 0008 XPHYSP 0004 -- 0002 MIIILP 0001 --
33	MIIADDR	MII Address
		bits [9:5] = PHYAD, Physical Layer Device Address bits [4:0] = REGAD, MII/Auto-Negotiation Register Address
34	MIIMDR	MII Data Port
35	PCI ID	PCI Vendor ID Register (DEFAULT = 1022h)
36	PMC_A	PCI Power Management Capabilities (DEFAULT = C802)
37	DATA 0	PCI Data Register Zero Alias Register (DEFAULT = 0000)
38	DATA 1	PCI Data Register One Alias Register (DEFAULT = 0000)
39	DATA 2	PCI Data Register Two Alias Register (DEFAULT = 0000)
40	DATA 3	PCI Data Register Three Alias Register (DEFAULT = 0000)
41	DATA 4	PCI Data Register Four Alias Register (DEFAULT = 0000)
42	DATA 5	PCI Data Register Five Alias Register (DEFAULT = 0000)
43	DATA 6	PCI Data Register Six Alias Register (DEFAULT = 0000)
44	DATA 7	PCI Data Register Seven Alias Register (DEFAULT = 0000)
45	PMR 1	OnNow Pattern Matching Register 1
46	PMR 2	OnNow Pattern Matching Register 2
47	PMR 3	OnNow Pattern Matching Register 3

Downloaded from Arrow.com.

ABSOLUTE MAXIMUM RATINGS

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

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

Supply voltage . . . . . with respect to V_{SSB}, V_{SS} . . . . . –0.3 V to 3.63 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Temperature (T) . . . . . 0°C to +70°C

Supply Voltages (VDD, AVDD) . . . . . +3.3 V ±10%

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

Operating ranges define those limits between which the functionality of the device is guaranteed.

Downloaded from Arrow.com.

DC CHARACTERISTICS OVER COMMERCIAL OPERATING RANGES

Unless specified otherwise

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Units
Digital I/O (Non-PCI Pins)					
V_{IH}	Input HIGH Voltage		2.0		V
V_{IL}	Input LOW Voltage			0.8	V
V_{OL}	Output LOW Voltage	I_{OL1} = 6 mA I_{OL2} = 12 mA (Note 1)		0.4	V
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH1}= -4 mA	2.4		V
I_{OZ}	Output Leakage Current (Note 3)	0 V <V_{OUT} <V_{DD}	-10	10	µA
I_{IX}	Input Leakage Current (Note 4)	0 V <V_{IN} <V_{DD}	-10	10	µA
I_{IL}	Input LOW Current (Note 5)	V_{IN} = 0 V; V_{DD} = 3.6 V	-200	-10	µA
I_{IH}	Input HIGH Current (Note 5)	V_{IN} = 2.7 V; V_{DD} = 3.6 V	-50	10	µA
PCI Bus Interface - 5 V Signaling					
V_{IH}	Input HIGH Voltage		2.0	Vcc + 0.5	V
V_{IL}	Input LOW Voltage		-0.5	0.8	V
I_{OZ}	Output Leakage Current (Note 3)	0 V <V_{IN} < V_{DD}	-10	10	µA
I_{IL}	Input LOW Current	V_{IN} = 0.5 V	--	-70	µA
I_{IH}	Input HIGH Current	V_{IN} = 2.7 V	--	70	µA
I_{IX_PME}	Input Leakage Current (Note 6)	0 V = < V_{IN} < 5.5 V	-1	1	µA
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH} = -2 mA	2.4		V
V_{OL}	Output LOW Voltage	I_{OL}3 = 3 mA I_{OL}4 = 6 mA (Note 1)		0.55	V
PCI Bus Interface - 3.3 V Signaling					
V_{IH}	Input HIGH Voltage		0.5 V_{DD}	V_{DD} + 0.5	V
V_{IL}	Input LOW Voltage		-0.5	0.3 V_{DD}	V
I_{OZ}	Output Leakage Current (Note 3)	0 V < V_{OUT} < V_{DD}	-10	10	µA
I_{IL}	Input HIGH Current	0 V < V_{IN} < V_{DD}	-10	10	µA
I_{IX_PME}	Input Leakage Current (Note 6)	0 V = < V_{IN} < 5.5 V	-1	1	µA
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH} = -500 µA	0.9 V_{DD}		V
V_{OL}	Output LOW Voltage	I_{OL} = 1500 µA		0.1 V_{DD}	V
Pin Capacitance					
C_{IN}	Pin Capacitance	F_C = 1 MHz (Note 7)		10	pF
C_{CLK}	CLK Pin Capacitance	F_C = 1 MHz (Notes 7,8)	5	12	pF
C_{IDSEL}	IDSEL Pin Capacitance	Fc = 1 MHz (Notes 7,9		8	pF
LPIN	Pin Inductance	Fc = 1 MHz (Note 7)		20	nH

Downloaded from Arrow.com.

Power Supply Current				
I_{DD}	Dynamic Current	PCI CLK at 33 MHz, MII Interface at 25 MHz, Full-Duplex operation, SRAM at 90 MHz	340	mA
I_{DD_WU1}	Wake-up current when the device is in the D1, D2, or D3 state and the PCI bus is in the B0 or B1 state.	PCI CLK at 33 MHz, MII Interface at 25 MHz, SRAM at 90 MHz. Device at Magic Packet or OnNow mode, receiving non-matching packets	280	mA
I_{DD_WU2}	Wake-up current when the device is in the D2 or D3 state and the PCI bus is in the B2 or B3 state.	PCI CLK LOW, MII Interface at 25 MHz, SRAM at 90 MHz, PG LOW, Device at Magic Packet or OnNow mode, receiving non-matching packets	265	mA
I_{DD_S}	Static I_{DD}	PCI CLK, SRAM pins and MII pins LOW.	1	mA

Notes:

1. *I_{OL1} applies to all non LED pins.*
   I_{OL2} applies to LED0, LED1, LED2, LED3, and WUMI.
   I_{OL3} applies to AD[31:0], C/BE[3:0], PAR, and REQ pins in a 5 V signaling environment.
   I_{OL4} applies to FRAME, TRDY, IRDY, DEVSEL, STOP, SERR, PERR, and INTA.
2. *V_{OH} does not apply to open-drain output pins.*
3. *I_{OZ} applies to all 3-state output and bidirectional pins, except the PME pin.*
4. *I_{IX} applies to all input pins except PME, TDI, TCK, and TMS pins. Tests are performed at V_{IN} = 0 V and at V_{DD} only.*
5. *I_{IL} and I_{IH} apply to the TDI, TCK, and TMS pins.*
6. *I_{IX_PME} applies to the PME pin only. Tests are performed at V_{IN} = 0 V and 5.5 V only.*
7. *Parameter not tested. Value determined by characterization.*
8. *C_{CLK} applies only to the CLK pin.*
9. *C_{IDSEL} applies only to the IDSEL pin.*

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: BUS INTERFACE

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

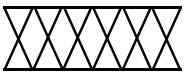
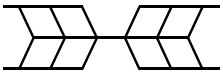
Note:

1. Not tested; parameter guaranteed by design characterization.

Downloaded from Arrow.com.

SWITCHING WAVEFORMS

Key to Switching Waveforms

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

KS000010-PAL

Downloaded from Arrow.com.

SWITCHING TEST CIRCUITS

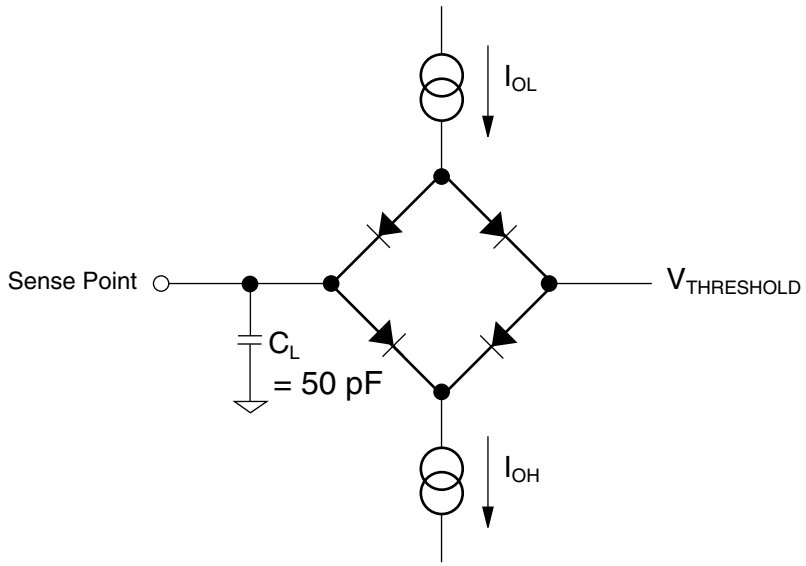


Figure 48. Normal and Tri-State Outputs

Downloaded from Arrow.com.

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE

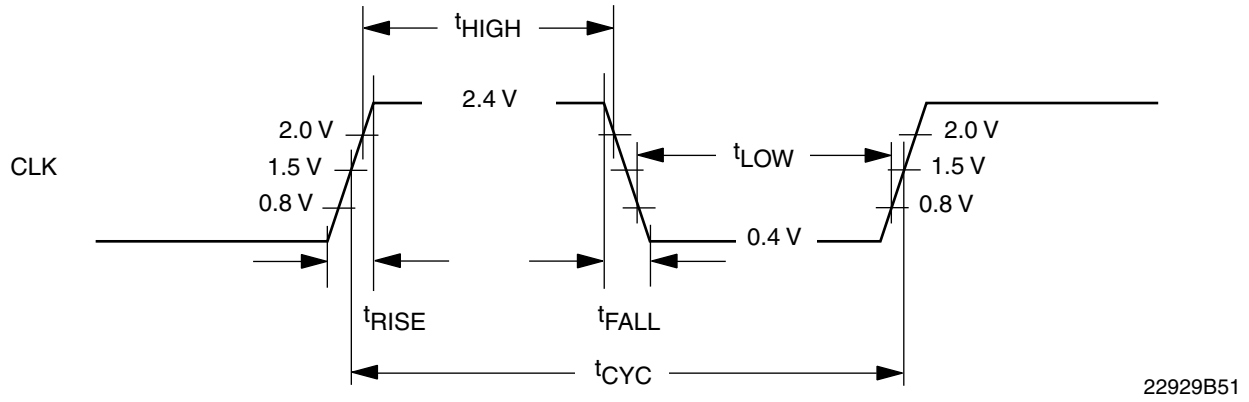


Figure 49. CLK Waveform for 5 V Signaling

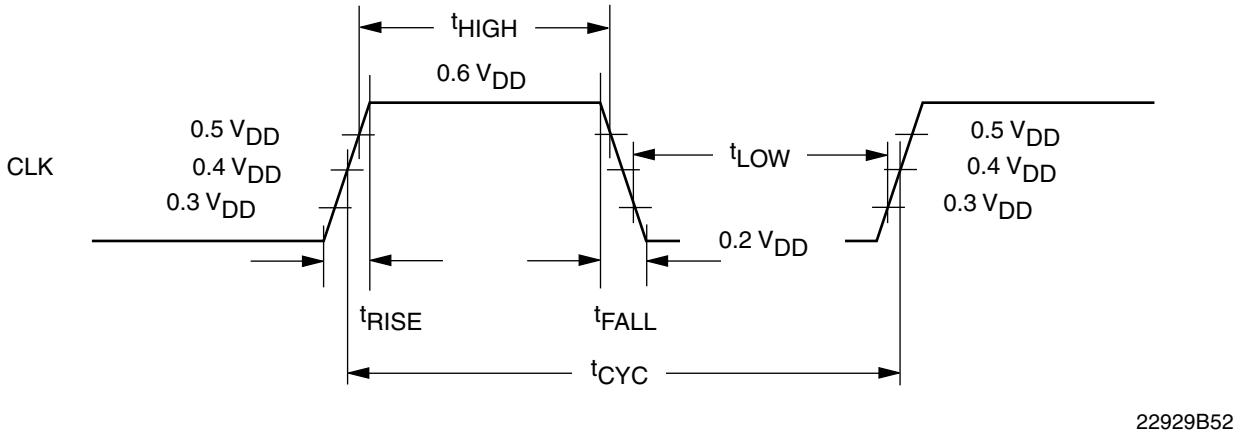


Figure 50. CLK Waveform for 3.3 V Signaling

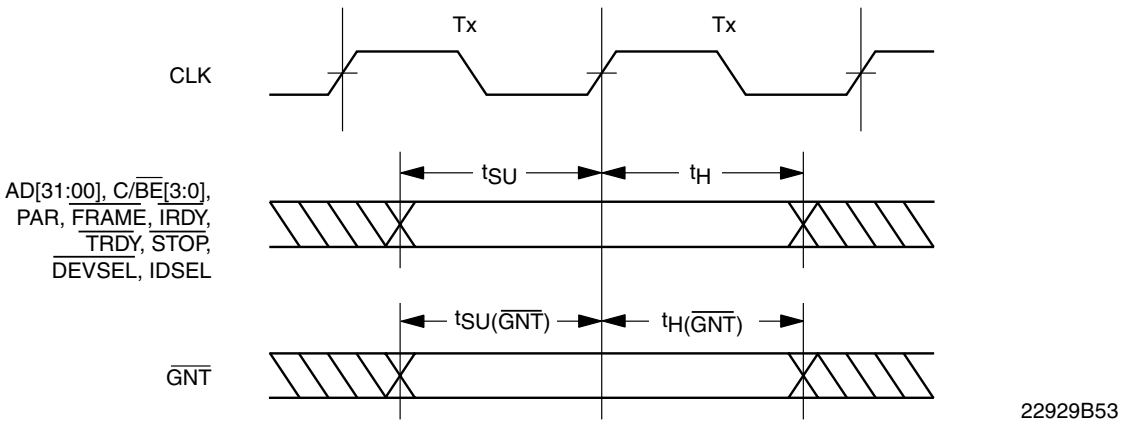


Figure 51. Input Setup and Hold Timing

Downloaded from Arrow.com.

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE (Continued)

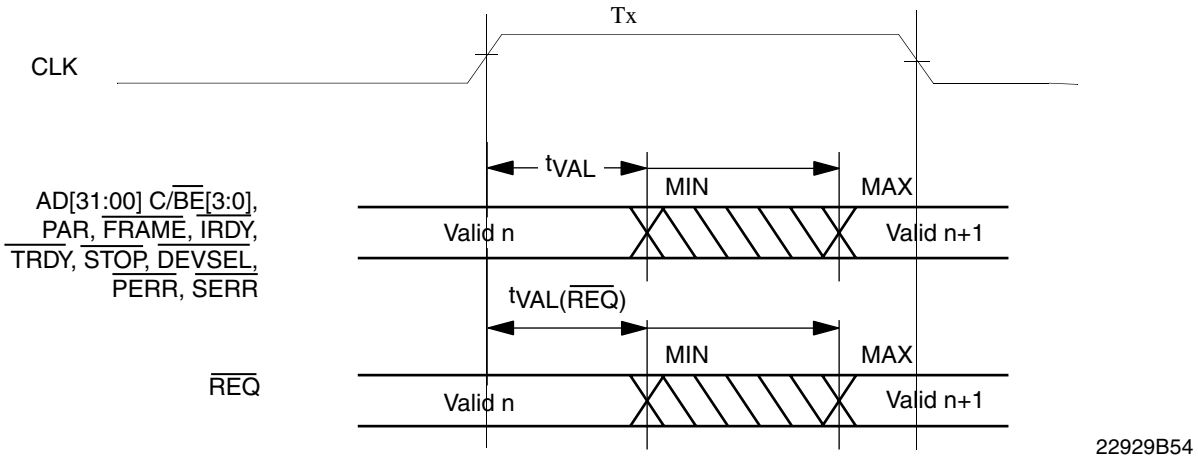


Figure 52. Output Valid Delay Timing

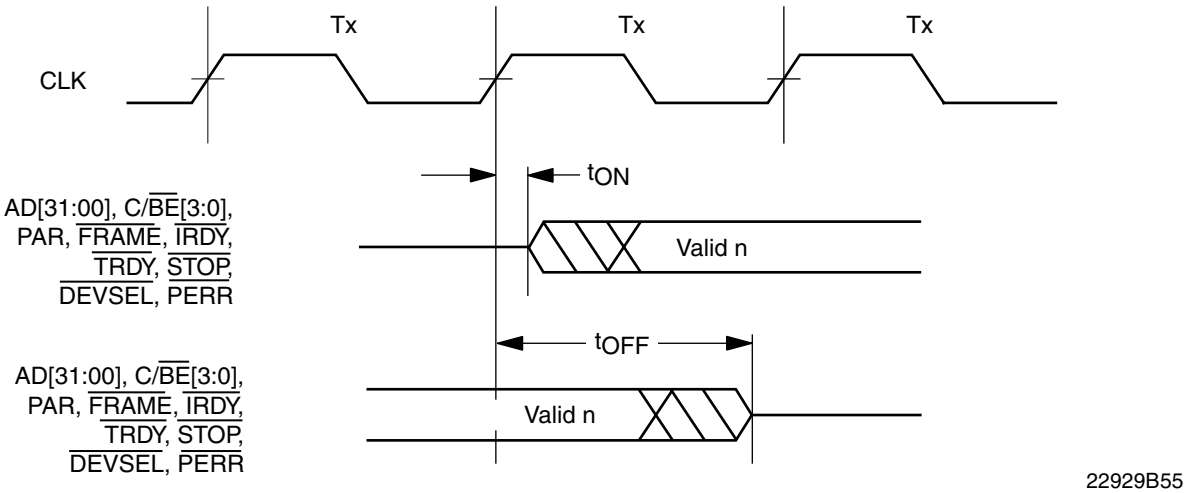


Figure 53. Output Tri-State Delay Timing

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: EEPROM INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
EEPROM Timing					
f_{EESK}	EESK Frequency	(Note 1)		650	kHz
t_{HIGH} (EESK)	EESK High Time		780		ns
t_{LOW} (EESK)	EESK Low Time		780		ns
t_{VAL} (EEDI)	EEDI Valid Output Delay from EESK	(Note 1)	-15	15	ns
t_{VAL} (EECS)	EECS Valid Output Delay from EESK	(Note 1)	-15	15	ns
t_{LOW} (EECS)	EECS Low Time		1550		ns
t_{SU} (EEDO)	EEDO Setup Time to EESK	(Note 1)	50		ns
t_H (EEDO)	EEDO Hold Time from EESK	(Note 1)	0		ns

Note:

1. *Parameter value is given for automatic EEPROM read operation. When EEPROM port (BCR19) is used to access the EEPROM, software is responsible for meeting EEPROM timing requirements.*

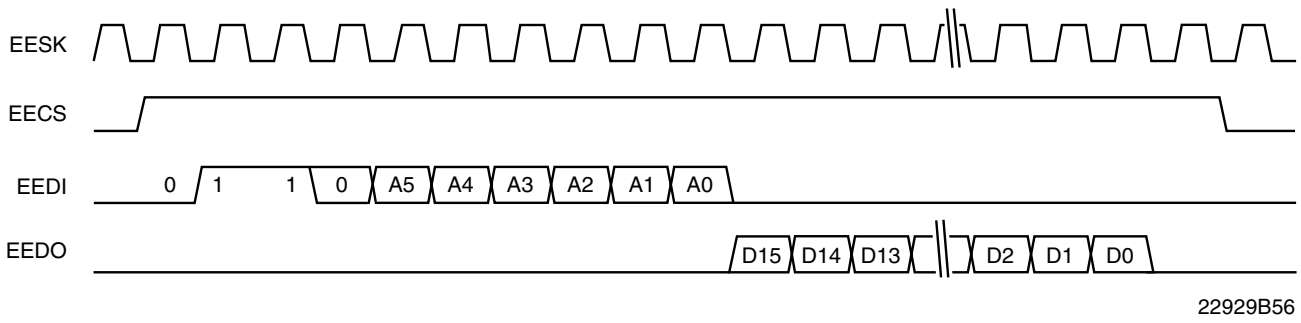


Figure 54. EEPROM Read Functional Timing

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: EEPROM INTERFACE (Continued)

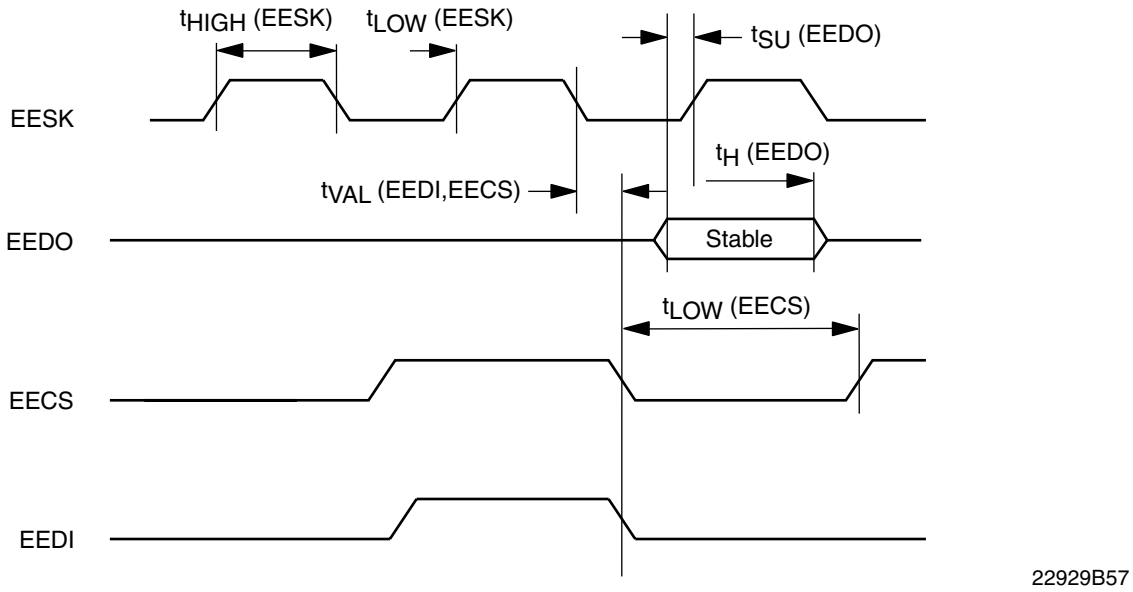


Figure 55. Automatic PREAD EEPROM Timing

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: JTAG TIMING

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
JTAG (IEEE 1149.1) Test Signal Timing					
t_{J1}	TCK Frequency			10	MHz
t_{J2}	TCK Period		100		ns
t_{J3}	TCK High Time	@ 2.0 V	45		ns
t_{J4}	TCK Low Time	@ 0.8 V	45		ns
t_{J5}	TCK Rise Time			4	ns
t_{J6}	TCK Fall Time			4	ns
t_{J7}	TDI, TMS Setup Time		8		ns
t_{J8}	TDI, TMS Hold Time		10		ns
t_{J9}	TDO Valid Delay		3	30	ns
t_{J10}	TDO Float Delay			50	ns
t_{J11}	All Outputs (Non-Test) Valid Delay		3	25	ns
t_{J12}	All Outputs (Non-Test) Float Delay			36	ns
t_{J13}	All Inputs (Non-Test)) Setup Time		8		ns
t_{J14}	All Inputs (Non-Test) Hold Time		7		ns

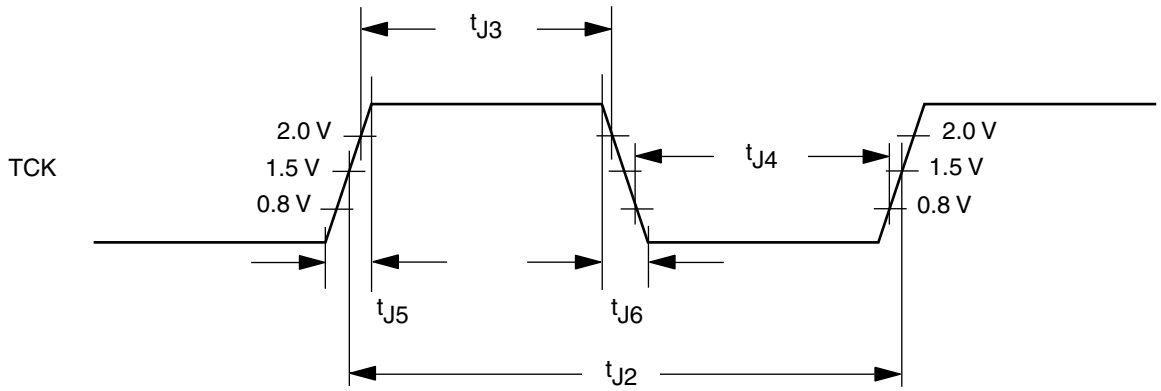


Figure 56. JTAG (IEEE 1149.1) TCK Waveform for 5 V Signaling

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: JTAG TIMING (Continued)

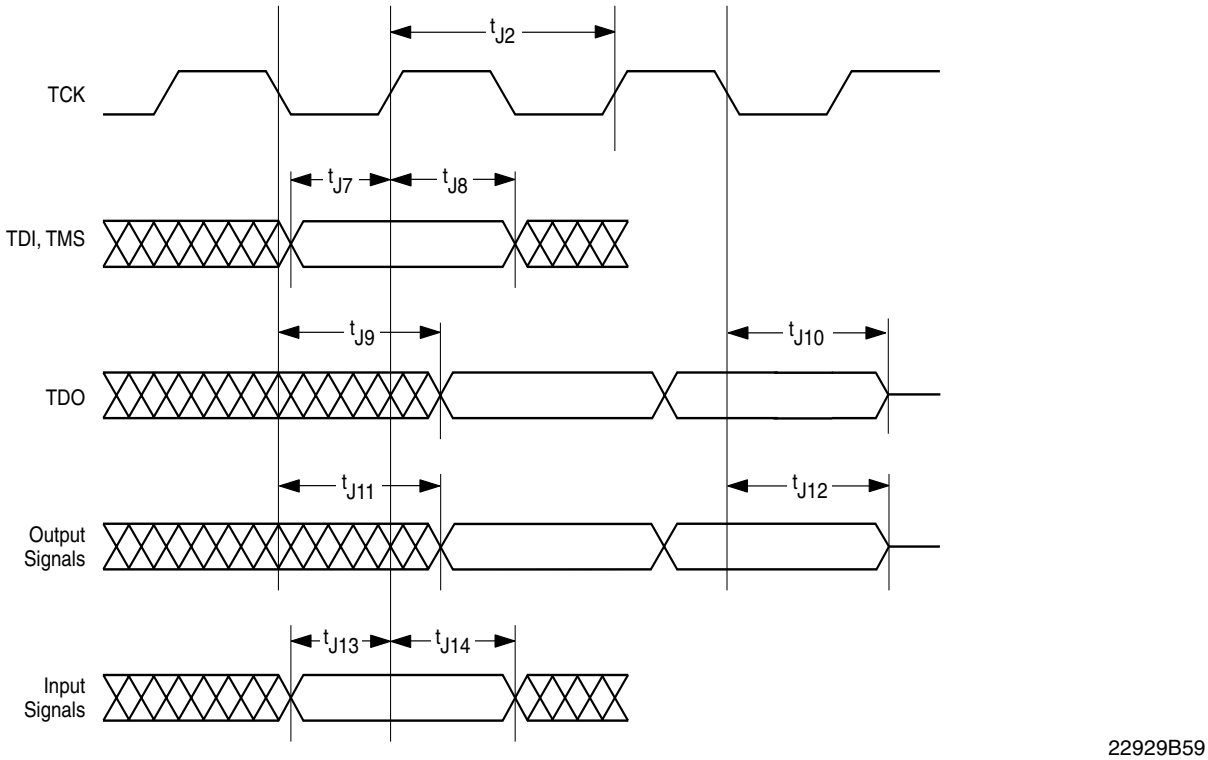


Figure 57. JTAG (IEEE 1149.1) Test Signal Timing

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: MEDIA INDEPENDENT INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
Transmit Timing					
t_{TVAL}	TX_EN, TX_ER, TXD valid from ↑ TX_CLK	measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V	0	25	ns
Receive Timing					
t_{RSU}	RX_DV, RX_ER, RXD setup to ↑ RX_CLK	measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V	10		ns
t_{RH}	RX_DV, RX_ER, RXD hold from ↑ RX_CLK	measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V	10		ns
Management Cycle Timing					
t_{MHIGH}	MDC Pulse Width HIGH Time	C_{LOAD} = 390 pf	160		ns
t_{MLOW}	MDC Pulse Width LOW Time	C_{LOAD} = 390 pf	160		ns
t_{MCYC}	MDC Cycle Period	C_{LOAD} = 390 pf	400		ns
t_{MSU}	MDIO setup to ↑ MDC	C_{LOAD} = 470 pf, measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V (Note 1, 3)	25		ns
t_{MH}	MDIO hold from ↑ MDC	C_{LOAD} = 470 pf, measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V (Note 1, 3)	10		ns
t_{MVAL}	MDIO valid from ↑ MDC	C_{LOAD} = 470 pf, measured from V_{ilmax} = 0.8 V or measured from V_{ihmin} = 2.0V, (Note 1, 4)	t_{MCYC} - t_{MSU}		ns

Notes:

1. *MDIO valid measured at the exposed mechanical Media Independent Interface.*
2. *TXCLK and RXCLK frequency and timing parameters are defined for the external physical layer transceiver as defined in the IEEE 802.3u standard. They are not replicated here.*
3. *t_{MSU} and t_{MH} are input requirements when MDIO is driven by an external PHY device.*
4. *t_{MVAL} is the output delay when MDIO is driven by the Am79C976 device.*

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: MEDIA INDEPENDENT INTERFACE (Continued)

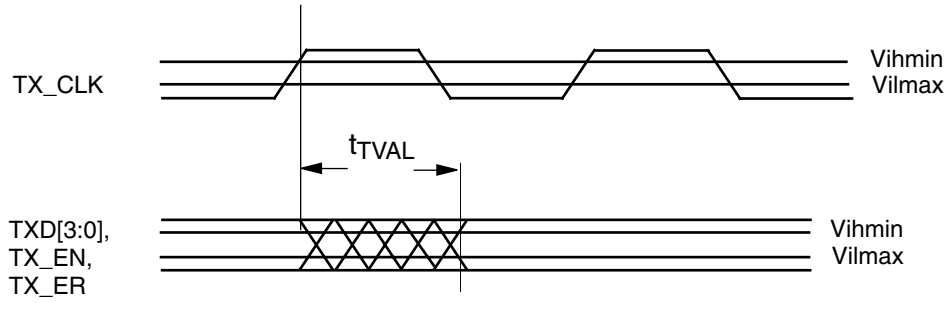


Figure 58. Transmit Timing

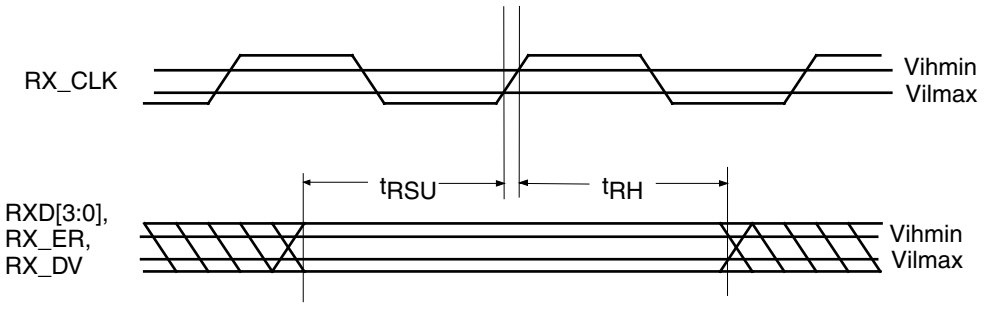


Figure 59. Receive Timing

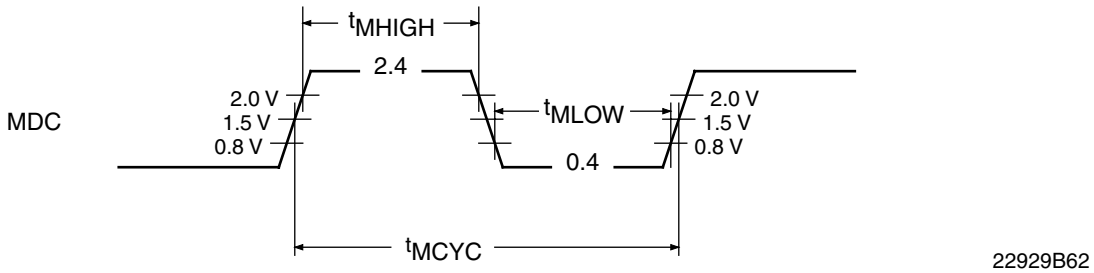


Figure 60. MDC Waveform

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: MEDIA INDEPENDENT INTERFACE (Concluded)

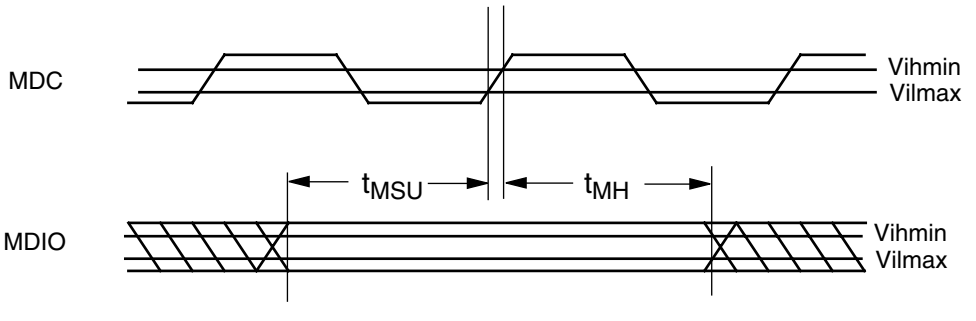


Figure 61. Management Data Setup and Hold Timing

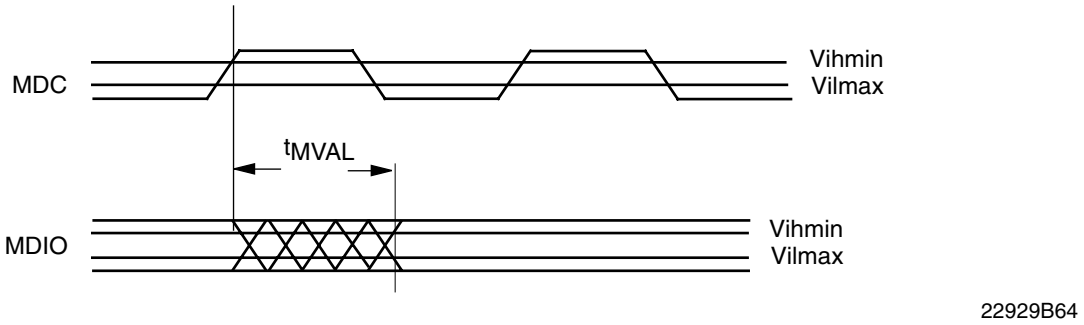


Figure 62. Management Data Output Valid Delay Timing

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: EXTERNAL ADDRESS DETECTION INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
External Address Detection Interface: External PHY - MII @ 25 MHz					
t_{EAD7}	SFBD change from ↓ RX_CLK		0	20 (Note 1)	ns
t_{EAD8}	$\overline{EAR}$ deassertion to ↑ RX_DV (first rising edge)		40		ns
t_{EAD9}	$\overline{EAR}$ assertion after SFD event (frame rejection)		0	5,080	ns
t_{EAD10}	$\overline{EAR}$ assertion width		50		ns
External Address Detection Interface: External PHY - MII @ 2.5 MHz					
t_{EAD11}	$\overline{EAR}$ deassertion to ↑ RX_DV (first rising edge)		400		ns
t_{EAD12}	$\overline{EAR}$ assertion after SFD event (frame rejection)		0	50,800	ns
t_{EAD13}	$\overline{EAR}$ assertion width		500		ns
Receive Frame Tag Timing with Media Independent Interface					
t_{EAD14}	RXFRTGE assertion from ↑SFBD (first rising edge)		0		ns
t_{EAD15}	RXFRTGE, RXFRTGD setup to ↑ RX_CLK		10		ns
t_{EAD16}	RXFRTGE, RXFRTGD hold from ↑ RX_CLK		10		ns
t_{EAD17}	RXFRTGE deassertion to ↓ RX_DV	RX_CLK @25 MHz RX_CLK @2.5 MHz	40 400		ns ns

Note:

1. May need to delay RX_CLK to capture Start Frame Byte Delimiter (SFBD) at 100 Mbps operation.

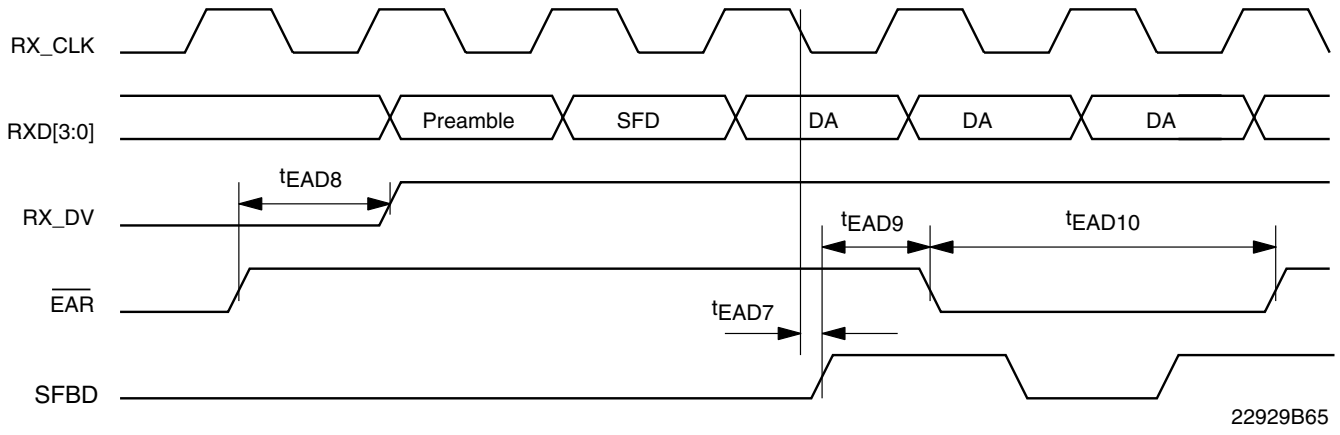


Figure 63. Reject Timing - External PHY MII @ 25 MHz

Downloaded from Arrow.com.

SWITCHING CHARACTERISTICS: EXTERNAL ADDRESS DETECTION INTERFACE (Concluded)

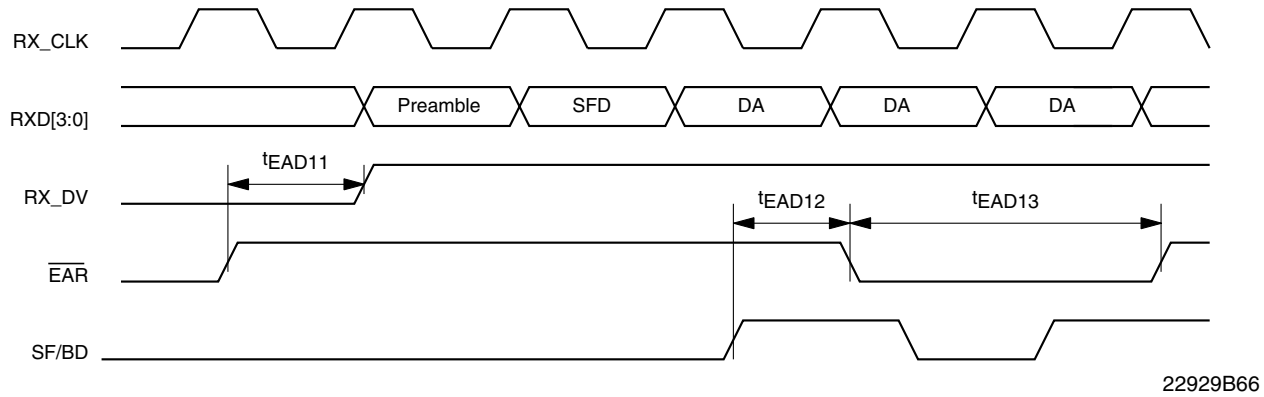


Figure 64. Reject Timing - External PHY MII @ 2.5 MHz

Downloaded from Arrow.com.

SWITCHING WAVEFORMS: RECEIVE FRAME TAG

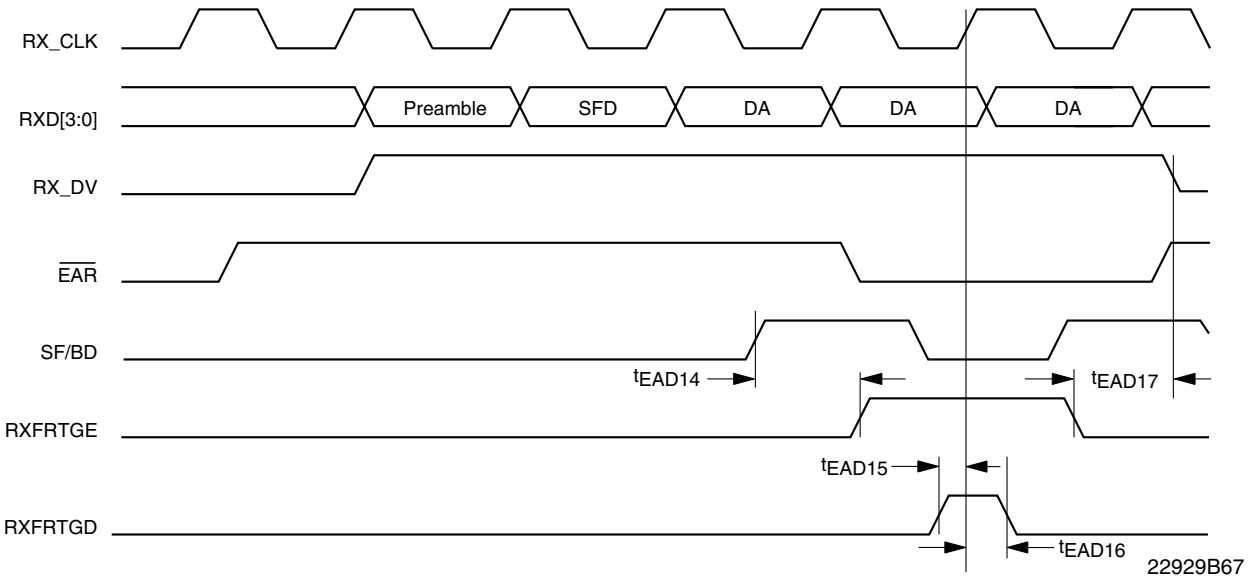


Figure 65. Receive Frame Tag Timing with Media Independent Interface

Downloaded from Arrow.com.

SWITCHING WAVEFORMS: EXTERNAL MEMORY INTERFACE

Parameter Symbol	Parameter Name	Test Condition	Min	Max	Unit
SSRAM Timing					
t_{ERCLK}	ERCLK Period	@ 1.5 V	11.111		ns
t_{HIGH}	ERCLK High Time	@ 1.5 V	4.5		ns
t_{LOW}	ERCLK Low Time	@ 1.5 V	4.5		ns
t_{O_SU}	SSRAM Output Signals Set up time with respect to rising edge of ERCLK	@ 1.5 V	2.5		ns
t_{ERD_SU}	SSRAM ERD Input Signals Set Up Time	@ 1.5 V	5		ns
t_{ERD_H}	SSRAM ERD Input Signals Hold Time	@ 1.5 V	1		ns

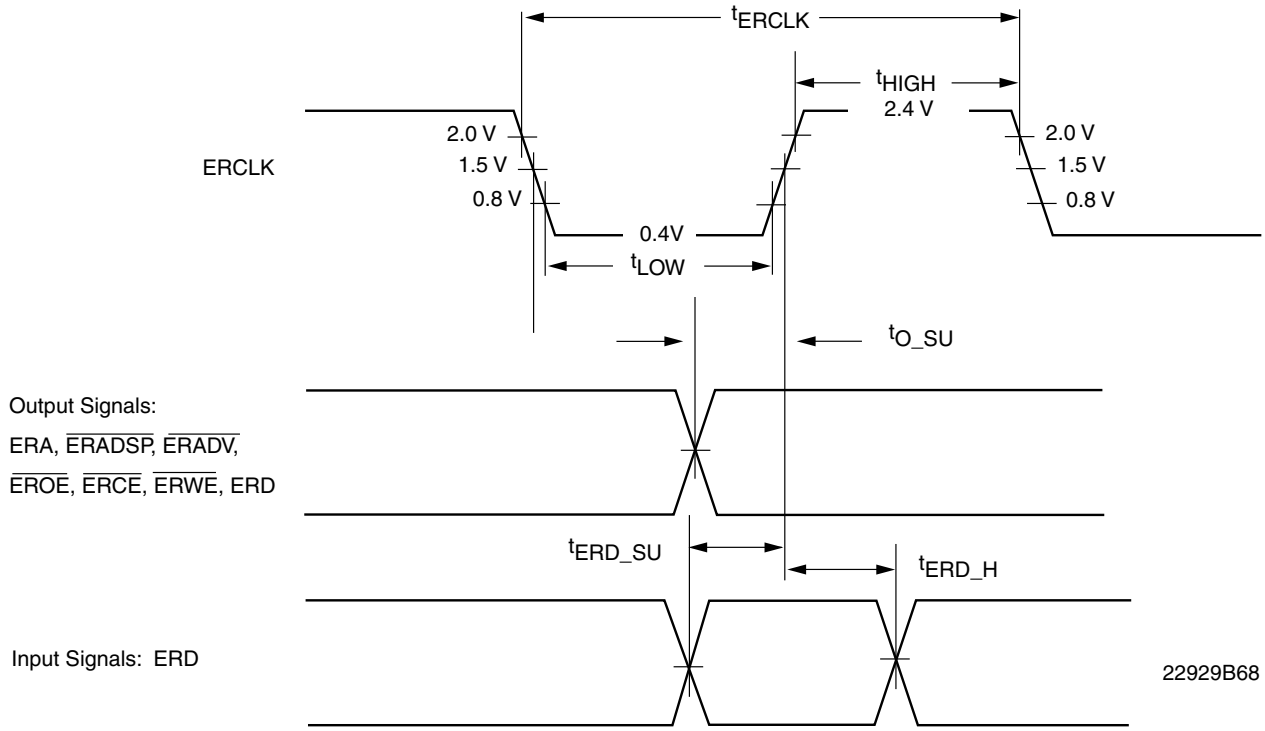


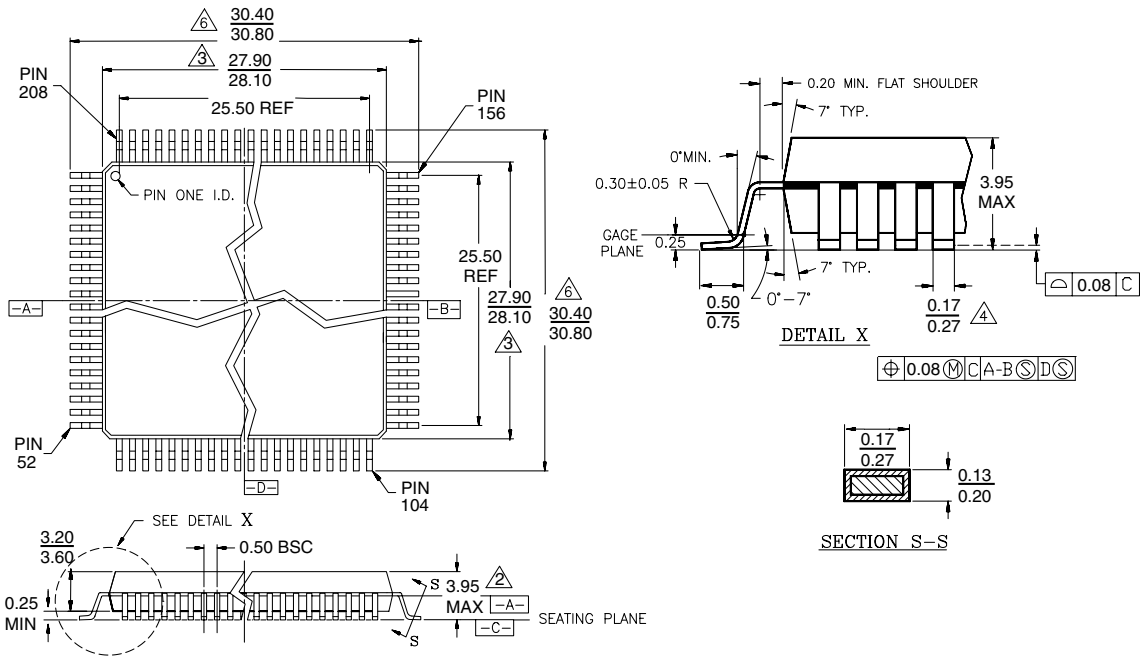
Figure 66. External Memory Timing

Downloaded from Arrow.com.

PHYSICAL DIMENSIONS*

PQFP208

Plastic Quad Flat Pack Trimmed and Formed



PQR 208

K.Koller
Rev. AI; 5/18/99

**For reference only. BSC is an ANSI standard for Basic Space Centering.*

Notes:

1. *All dimensions and tolerances conform to ASME Y14.5M-1992*
2. *Controlling dimensions: Millimeters*
3. *These dimensions do not include mold protrusion. Allowable protrusion is 0.25 mm per side. These dimensions include mold mismatch and are determined at datum plane -A-.*
4. *This dimension does not include dambar protrusion.*
5. *Datum plane -A- is located at the mold parting line and is coincident with the bottom of the lead where the lead exits the plastic body.*
6. *These dimensions are measured from both innermost and outermost points.*
7. *Deviation from lead-tip true position shall be within ±0.04 mm.*
8. *Lead co-planarity shall be within 0.076 mm. Co-planarity is measured per specification 06.500.*
9. *Half span (center of package to lead tip) shall be within ±0.0085 inches.*

Downloaded from Arrow.com.

Downloaded from Arrow.com.

AMD

Look-Ahead Packet Processing (LAPP) Concept

APPENDIX A: LOOK-AHEAD PACKET PROCESSING

Introduction

A driver for the Am79C976 controller would normally require that the CPU copy receive frame data from the controllers buffer space to the applications buffer space after the entire frame has been received by the controller. For applications that use a ping-pong windowing style, the traffic on the network will be halted until the current frame has been completely processed by the entire application stack. This means that the time between last byte of a receive frame arriving at the client's Ethernet controller and the client's transmission of the first byte of the next outgoing frame will be separated by:

1. The time that it takes the client's CPU interrupt procedure to pass software control from the current task to the driver,
2. Plus the time that it takes the client driver to pass the header data to the application and request an application buffer,
3. Plus the time that it takes the application to generate the buffer pointer and then return the buffer pointer to the driver,
4. Plus the time that it takes the client driver to transfer all of the frame data from the controller's buffer space into the application's buffer space and then call the application again to process the complete frame,
5. Plus the time that it takes the application to process the frame and generate the next outgoing frame,
6. Plus the time that it takes the client driver to set up the descriptor for the controller and then write a TDMD bit to CSR0.

The sum of these times can often be about the same as the time taken to actually transmit the frames on the wire, thereby, yielding a network utilization rate of less than 50 percent.

An important thing to note is that the Am79C976 controller's data transfers to its buffer space are such that the system bus is needed by the Am79C976 controller for approximately 4 percent of the time. This leaves 96 percent of the system bus bandwidth for the CPU to perform some of the interframe operations in advance of the completion of network receive activity, if possible. The question then becomes: how much of the tasks that need to be performed between reception of a frame and transmission of the next frame can be performed before the reception of the frame actually ends at the network, and how can the CPU be instructed to perform these tasks during the network reception time?

The answer depends upon exactly what is happening in the driver and application code, but the steps that can be performed at the same time as the receive data are arriving include as much as the first three steps and part of the fourth step shown in the sequence above. By performing these steps before the entire frame has arrived, the frame throughput can be substantially increased.

A good increase in performance can be expected when the first three steps are performed before the end of the network receive operation. A much more significant performance increase could be realized if the Am79C976 controller could place the frame data directly into the application's buffer space; (i.e., eliminate the need for step 4.) In order to make this work, it is necessary that the application buffer pointer be determined before the frame has completely arrived, then the buffer pointer in the next descriptor for the receive frame would need to be modified in order to direct the Am79C976 controller to write directly to the application buffer. More details on this operation will be given later.

An alternative modification to the existing system can gain a smaller but still significant improvement in performance. This alternative leaves step 4 unchanged in that the CPU is still required to perform the copy operation, but it allows a large portion of the copy operation to be done before the frame has been completely received by the controller, i.e., the CPU can perform the copy operation of the receive data from the Am79C976 controller's buffer space into the application buffer space before the frame data has completely arrived from the network. This allows the copy operation of step 4 to be performed concurrently with the arrival of network data, rather than sequentially, following the end of network receive activity.

Downloaded from Arrow.com.

Outline of LAPP Flow

This section gives a suggested outline for a driver that utilizes the LAPP feature of the Am79C976 controller.

Note: *The labels in the following text are used as references in the timeline diagram that follows (Figure A-1).*

Setup

The driver should set up descriptors in groups of three, with the OWN and STP bits of each set of three descriptors to read as follows: 11b, 10b, 00b.

An option bit (LAPPEN) exists in CSR3, bit position 5; the software should set this bit. When set, the LAPPEN bit directs the Am79C976 controller to generate an INTERRUPT when STP has been written to a receive descriptor by the Am79C976 controller.

Flow

The Am79C976 controller polls the current receive descriptor at some point in time before a message arrives. The Am79C976 controller determines that this receive buffer is OWNed by the Am79C976 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 Am79C976 controller to begin frame data DMA operations to the first buffer.

C0 When the 64th byte of the message arrives, the Am79C976 controller performs a lookahead operation to the next receive descriptor. This descriptor should be owned by the Am79C976 controller.

C1 The Am79C976 controller intermittently requests the bus to transfer frame data to the first buffer as it arrives on the wire.

S1 The driver remains idle.

C2 When the Am79C976 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 Am79C976 controller to the CPU, the Am79C976 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 Am79C976 controller's driver to run.

C4 During the CPU interrupt-generated task switching, the Am79C976 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 Am79C976 controller, existing AMD Ethernet controllers will continue to perform data DMA into the buffer space that the controller already owns (i.e., buffer number 2). The controller does not know if buffer space in buffer number 2 will be sufficient or not for this frame, but it has no way to tell except by trying to move the entire message into that space. Only when the message does not fit will it signal a buffer error condition--there is no need to panic at this point that it discovers that it does not yet own descriptor number 3.*

S2 The first task of the drivers interrupt service route is to collect the header information from the Am79C976 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 Am79C976 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 Am79C976 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 Am79C976 controller.

C5 Interleaved with S2, S3, and S4 driver activity, the Am79C976 controller will write frame data to buffer number 2.

S4 The driver will next proceed to copy the contents of the Am79C976 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 Am79C976 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 Am79C976 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

Downloaded from Arrow.com.

application pointer into this descriptor and then changed the ownership to give the descriptor to the Am79C976 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 Am79C976 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 Am79C976 controller, the next driver poll of the second descriptor will show ownership granted to the CPU. The driver now copies the data from buffer number 2 into the middle section of the application buffer space. This operation is interleaved with the C7 and C8 operations.

C8 The Am79C976 controller will perform data DMA to the last buffer, whose pointer is pointing to application space. Data entering the least buffer will not need the infamous double copy that is required by existing drivers, since it is being placed directly into the application buffer space.

N2 The message on the wire ends.

S7 When the driver completes the copy of buffer number 2 data to the application buffer space, it begins polling descriptor number 3.

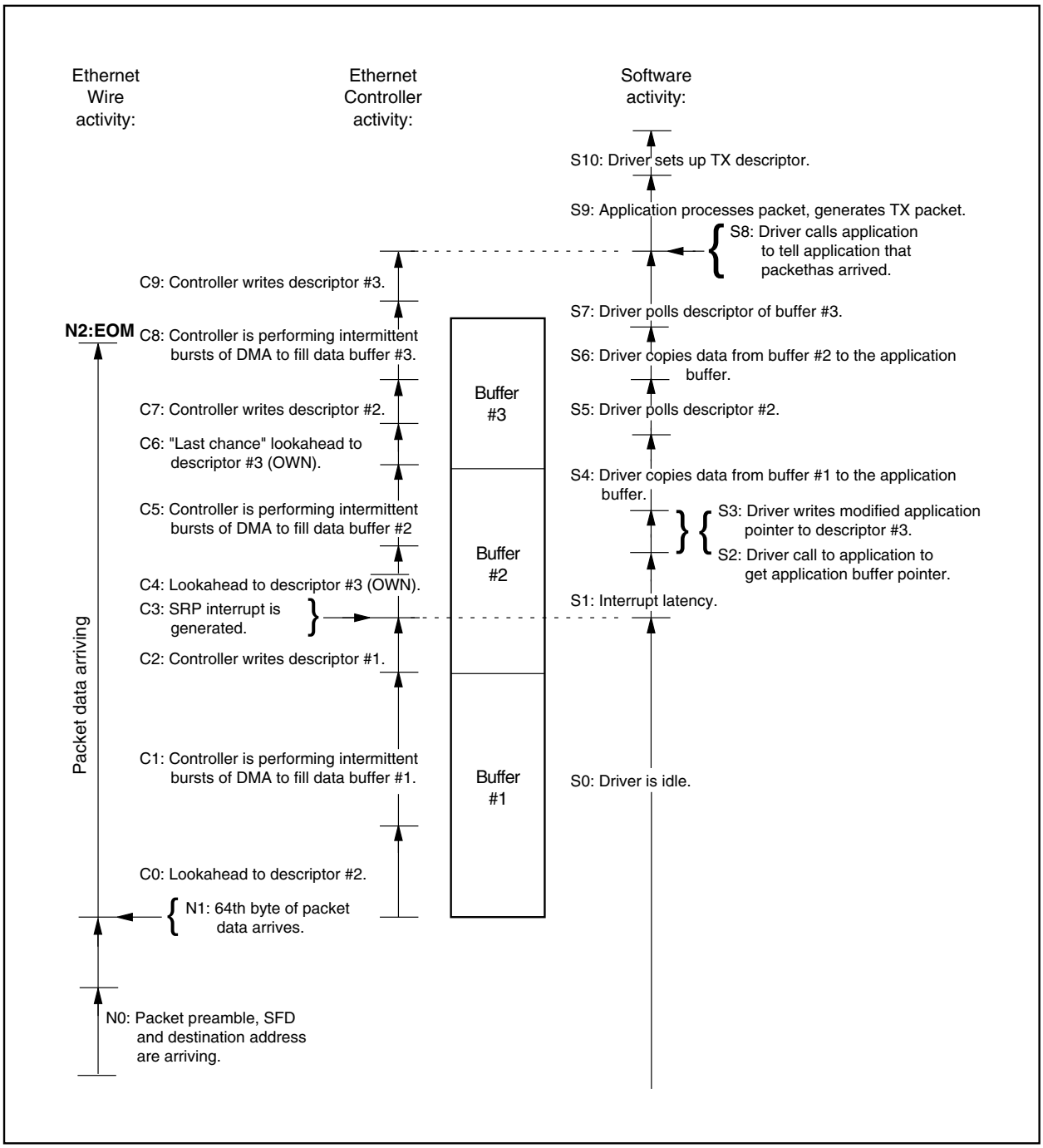
C9 When the Am79C976 controller has finished all data DMA operations, it writes status and changes ownership of descriptor number 3.

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 Am79C976 controller.

Downloaded from Arrow.com.



D-1

Figure A-1. LAPP Timeline

Downloaded from Arrow.com.

LAPP Software Requirements

Software needs to set up a receive ring with descriptors formed into groups of three. The first descriptor of each group should have OWN = 1 and STP = 1, the second descriptor of each group should have OWN = 1 and STP = 0. The third descriptor of each group should have OWN = 0 and STP = 0. The size of the first buffer (as indicated in the first descriptor) should be at least equal to the largest expected header size; however, for maximum efficiency of CPU utilization, the first buffer size should be larger than the header size. It should be equal to the expected number of message bytes, minus the time needed for interrupt latency and minus the application call latency, minus the time needed for the driver to write to the third descriptor, minus the time needed for the drive to copy data from buffer number 2 to the application buffer space. Note that the time needed for the copies performed by the driver depends upon the sizes of the second and third buffers, and that the sizes of the second and third buffers need to be set according to the time needed for the data copy operations. This means that an iterative self-adjusting mechanism needs to be placed into the software to determine the correct buffer sizing for optimal operation. Fixed values for buffer sizes may be used; in such a case, the LAPP method will still provide a significant performance increase, but the performance increase will not be maximized.

Figure A-2 illustrates this setup for a receive ring size of 9.

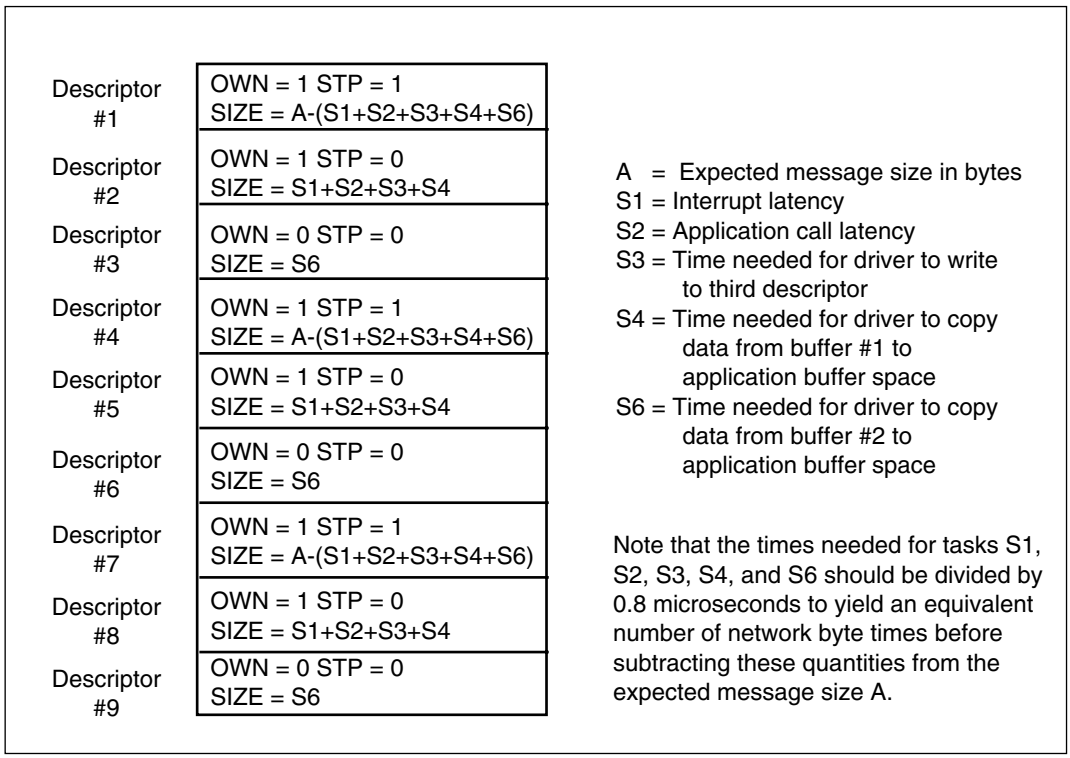


Figure A-2. LAPP 3 Buffer Grouping

LAPP Rules for Parsing Descriptors

When using the LAPP method, software must use a modified form of descriptor *parsing* as follows:

- Software will examine OWN and STP to determine where an RCV frame begins. RCV frames will only begin in buffers that have OWN = 0 and STP = 1.
- Software shall assume that a frame continues until it finds either ENP = 1 or ERR = 1.
- Software must discard all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for the beginning of a new frame; ENP and ERR should be ignored by software during this search.
- Software cannot change an STP value in the receive descriptor ring after the initial setup of the ring is complete, even if software has ownership of the STP

Downloaded from Arrow.com.

descriptor, unless the previous STP descriptor in the ring is also OWNED by the software.

When LAPPEN = 1, then hardware will use a modified form of descriptor *parsing* as follows:

- The controller will examine OWN and STP to determine where to begin placing an RCV frame. A new RCV frame will only begin in a buffer that has OWN = 1 and STP =1.
- The controller will always obey the OWN bit for determining whether or not it may use the next buffer for a chain.
- The controller will always mark the end of a frame with either ENP = 1 or ERR = 1.

The controller will discard all descriptors with OWN = 1 and STP = 0 and move to the next descriptor when searching for a place to begin a new frame. It discards these descriptors by simply changing the ownership bit from OWN = 1 to OWN = 0. Such a descriptor is unused for receive purposes by the controller, and the driver must recognize this. (The driver will recognize this if it follows the software rules.)

The controller will ignore all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for a place to begin a new frame. In other words, the controller is allowed to skip entries in the ring that it does not own, but only when it is looking for a place to begin a new frame.

Some Examples of LAPP Descriptor Interaction

Choose an expected frame size of 1060 bytes. Choose buffer sizes of 800, 200, and 200 bytes.

- **Example 1**: Assume that a 1060 byte frame arrives correctly, and that the timing of the early interrupt and the software is smooth. The descriptors will have changed from:

Descriptor Number	Before the Frame Arrives			After the Frame Arrives			Comments (After Frame Arrival)
	OWN	STP	ENP[a]	OWN	STP	ENP[b]	
1	1	1	X	0	1	0	Bytes 1-800
2	1	0	X	0	0	0	Bytes 801-1000
3	0	0	X	0	0	1	Bytes 1001-1060
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Net yet used

a. *& b. ENP or ERR.*

- **Example 2**: Assume that instead of the expected 1060 byte frame, a 900 byte frame arrives, either because there was an error in the network, or because this is the last frame in a file transmission sequence

Descriptor Number	Before the Frame Arrives			After the Frame Arrives			Comments (After Frame Arrival)
	OWN	STP	ENP[a]	OWN	STP	ENP[b]	
1	1	1	X	0	1	0	Bytes 1-800
2	1	0	X	0	0	0	Bytes 801-1000
3	0	0	X	0	0	?*	Discarded buffer
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Net yet used

a. *& b. ENP or ERR.*

Note: *The Am79C976 controller might write a ZERO to ENP location in the third descriptor. Here are the two possibilities:*

1. If the controller finishes the data transfers into buffer number 2 after the driver writes the application modified buffer pointer into the third descriptor, then

Downloaded from Arrow.com.

the controller will write a ZERO to ENP for this buffer and will write a ZERO to OWN and STP.

2. If the controller finishes the data transfers into buffer number 2 before the driver writes the applications modified buffer point into the third descriptor, then the controller will complete the frame in buffer number 2 and then skip the then unowned third buffer. In this case, the Am79C976 controller will not have had the opportunity to RESET the ENP bit in this descriptor, and it is possible that the software left this bit as ENP = 1 from the last time through the ring. Therefore, the software must treat the location as a *don't care.* The rule is, after finding ENP = 1 (or ERR = 1) in descriptor number 2, the software must ignore ENP bits until it finds the next STP = 1.

■ **Example 3**: Assume that instead of the expected 1060 byte frame, a 100 byte frame arrives, because there was an error in the network, or because this is the last frame in a file transmission sequence, or perhaps because it is an acknowledge frame.

**Same as note in example 2 above, except that in this case, it is very unlikely that the driver can respond to the interrupt and get the pointer from the application before the Am79C976 controller has completed its poll of the next descriptors. This means that for almost all occurrences of this case, the Am79C976 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 Am79C976 controller will not have had an opportunity to modify it.*

***Note that even though the Am79C976 controller will write a ZERO to this ENP location, the software should treat the location as a don't care, since after finding the ENP = 1 in descriptor number 2, the software should ignore ENP bits until it finds the next STP = 1.*

Descriptor Number	Before the Frame Arrives			After the Frame Arrives			Comments (After Frame Arrival)
	OWN	STP	ENP[a]	OWN	STP	ENP[b]	
1	1	1	X	0	1	0	Bytes 1-800
2	1	0	X	0	0	0**	Discarded buffer
3	0	0	X	0	0	?	Discarded buffer
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Net yet used

a. *& b.ENP or ERR*.

Buffer Size Tuning

For maximum performance, buffer sizes should be adjusted depending upon the expected frame size and the values of the interrupt latency and application call latency. The best driver code will minimize the CPU utilization while also minimizing the latency from frame end on the network to the frame sent to application from driver (frame latency). These objectives are aimed at increasing throughput on the network while decreasing CPU utilization.

Note: *The buffer sizes in the ring may be altered at any time that the CPU has ownership of the corresponding descriptor. The best choice for buffer sizes will maximize the time that the driver is swapped out, while minimizing the time from the last byte written by the Am79C976 controller to the time that the data is passed from the driver to the application. In the diagram, this corresponds to maximizing S0, while minimizing the time between C9 and S8. (the timeline happens to show a minimal time from C9 to S8.)*

Note: *By increasing the size of buffer number 1, we increase the value of S0. However, when we increase the size of buffer number 1, we also increase the value of S4. If the size of buffer number 1 is too large, then the driver will not have enough time to perform tasks S2, S3, S4, S5, and S6. The result is that there will be delay from the execution of task C9 until the execution of task S8. A perfectly timed system will have the values for S5 and S7 at a minimum.*

An average increase in performance can be achieved, if the general guidelines of buffer sizes in Figure A-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

Downloaded from Arrow.com.

upon the interrupt latency, which may vary from system to system, depending upon both the hardware and the software installed in each system.

In order to deal with the unpredictable nature of the message size, the driver can implement a self-tuning mechanism that examines the amount of time spent in tasks S5 and S7. As such, while the driver is polling for each descriptor, it could count the number of poll operations performed and then adjust the number 1 buffer size to a larger value, by adding “t” bytes to the buffer count, if the number of poll operations was greater than ”x.” If fewer than “x” poll operations were needed for each of S5 and S7, then software should adjust the buffer size to a smaller value by subtracting “y” bytes from the buffer count. Experiments with such a tuning mechanism must be performed to determine the best values for “x” and “y.”

Note: *Whenever the size of buffer number 1 is adjusted, buffer sizes for buffer number 2 and buffer number 3 should also be adjusted.*

In some systems, the typical mix of receive frames on a network for a client application consists mostly of large data frames, with very few small frames. In this case, for maximum efficiency of buffer sizing, when a frame arrives under a certain size limit, the driver should not adjust the buffer sizes in response to the short frame.

An Alternative LAPP Flow: Two-Interrupt Method

An alternative to the above suggested flow is to use two interrupts, one at the start of the receive frame and the other at the end of the receive frame, instead of just looking for the SRP interrupt as described above. This alternative attempts to reduce the amount of time that the software wastes while polling for descriptor own bits. This time would then be available for other CPU tasks. It also minimizes the amount of time the CPU needs for data copying. This savings can be applied to other CPU tasks.

The time from the end of frame arrival on the wire to delivery of the frame to the application is labeled as frame latency. For the one-interrupt method, frame latency is minimized, while CPU utilization increases. For the two-interrupt method, frame latency becomes greater, while CPU utilization decreases. See Figure A-3.

Note: *Some of the CPU time that can be applied to non-Ethernet tasks is used for task switching in the CPU. One task switch is required to swap a non-Ethernet task into the CPU (after S7A) and a second task switch is needed to swap the Ethernet driver back in again (at S8A). If the time needed to perform these task switches exceeds the time saved by not polling descriptors, then there is a net loss in performance with this method. Therefore, the LAPP method implemented should be carefully chosen.*

Figure A-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.

Downloaded from Arrow.com.

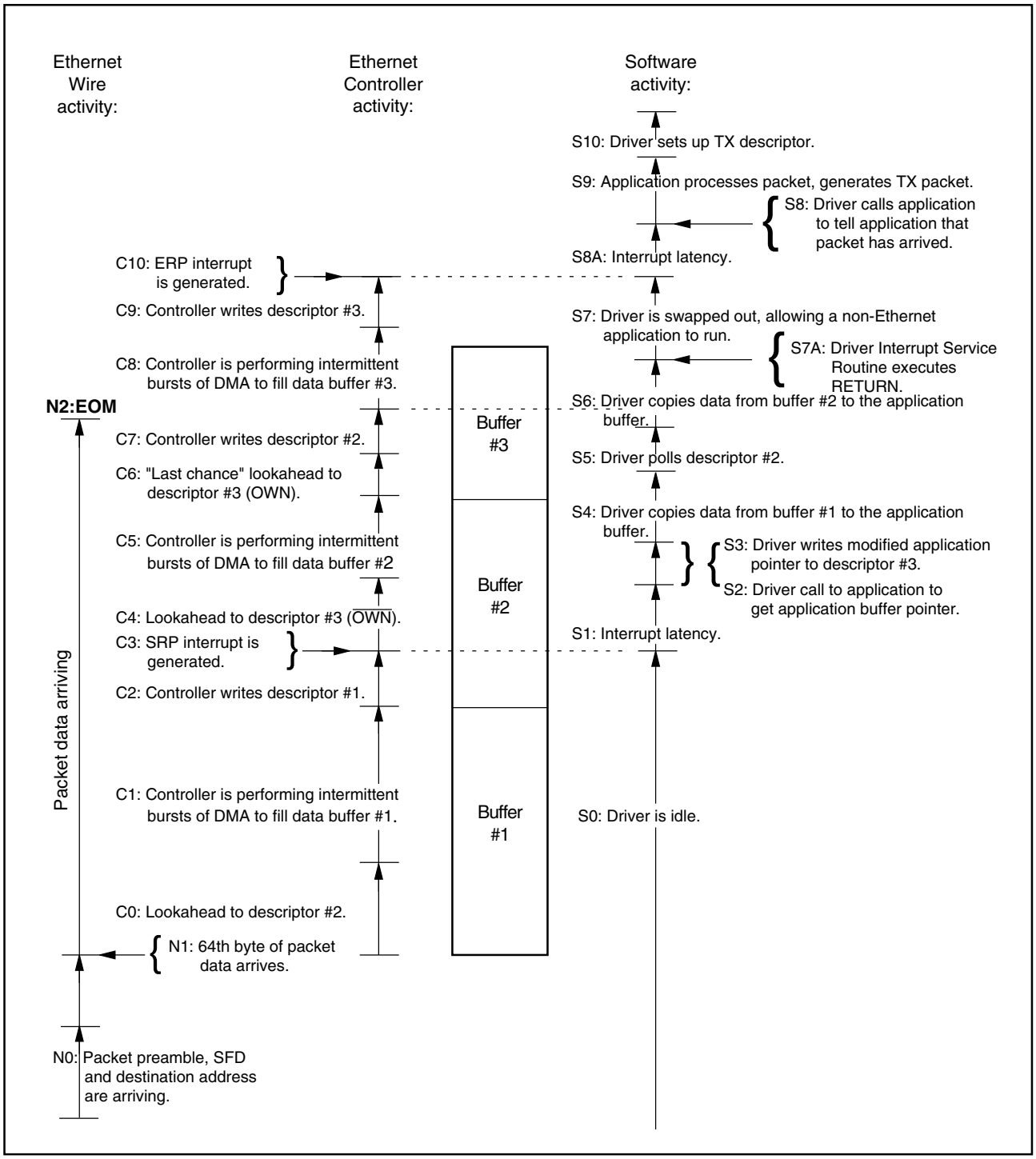


Figure A-3. LAPP Timeline for Two-Interrupt Method

Downloaded from Arrow.com.

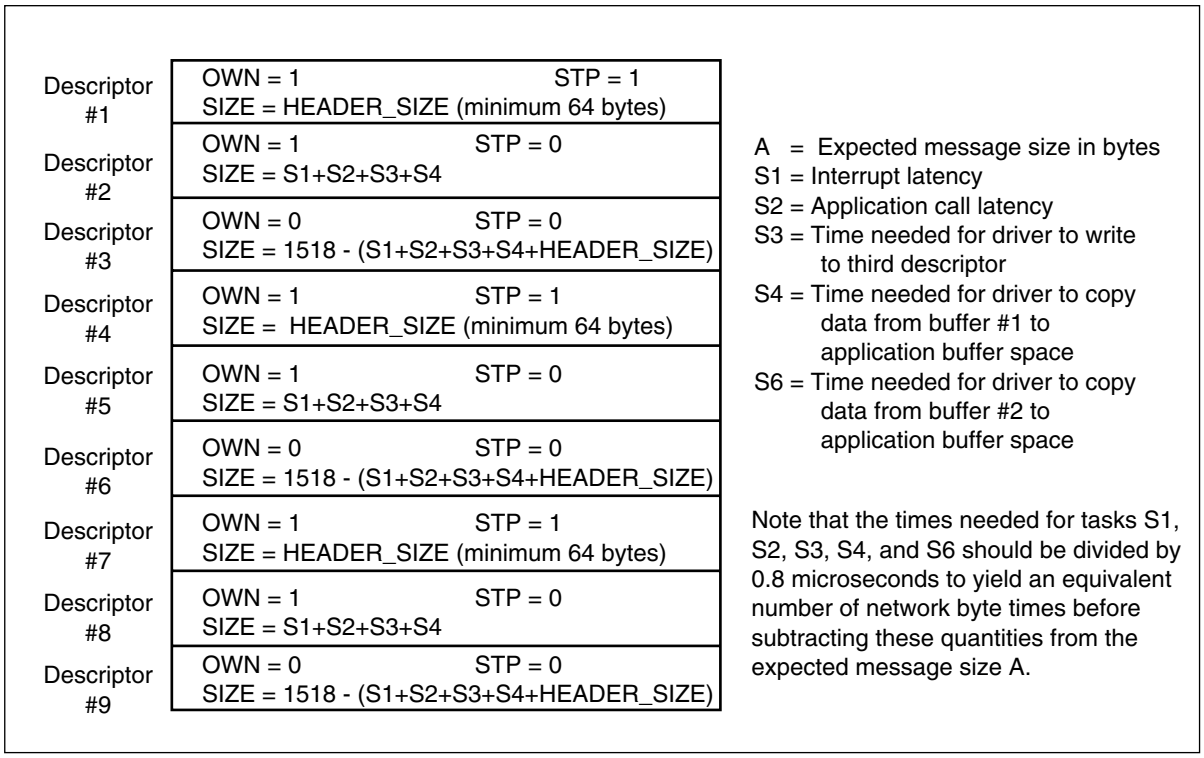


Figure A-4. LAPP 3 Buffer Grouping for Two-Interrupt Method

Downloaded from Arrow.com.

MII Management Registers

APPENDIX B: MII MANAGEMENT REGISTERS

As specified in the IEEE standard, the basic register set consists of the Control Register (Register 0) and the Status Register (Register 1). The extended register set consists of Registers 2 to 31 (decimal). All PHYs that provide an MII shall incorporate the basic register set. Both sets of registers are accessible through the MII Management Interface.

Table 1 lists the most interesting registers.

Table B-1. MII Management Register Set

Register Address	Register Name	Basic/ Extended
0	MII Control	B
1	MII Status	B
2-3	PHY Identifier	E
4	Auto-Negotiation Advertisement	E
5	Auto-Negotiation Link Partner Ability	E

Control Register (Register 0)

Table 2 shows the MII Management Control Register (Register 0).

Table B-2. MII Management Control Register (Register 0)

Bits	Name	Description	Read/Write (Note 1)
15	Soft Reset	When write: 1 = PHY software reset 0 = normal operation When read: 1 = reset in process 0 = reset done	R/W, SC
14	Loopback	1 = enables Loopback mode 0 = disables Loopback mode	R/W
13	Speed Selection	1 = 100 Mbps 0 = 10 Mbps	R/W
12	Auto-Negotiation Enable	1 = enable Auto-Negotiation 0 = disable Auto-Negotiation	R/W
11	Power Down	1 = power down, 0 = normal operation	R/W
10	Isolate	1 = electrically isolate PHY from MII 0 = normal operation	R/W
9	Restart Auto-Negotiation	1 = restart Auto-Negotiation 0 = normal operation	R/W, SC
8	Duplex Mode	1 = full duplex 0 = half duplex	R/W
7	Collision Test	1 = enable COL signal test 0 = disable COL signal test	R/W
6-0	Reserved	Write as 0, ignore on read	RO

Note:

1. R/W = Read/Write, SC = Self Clearing, RO = Read Only.

Downloaded from Arrow.com.

Status Register (Register 1)

The MII Management Status Register identifies the physical and auto-negotiation capabilities of the PHY. This register is read only; a write will have no effect. See Table 3.

Table B-3. MII Management Status Register (Register 1)

Bits	Name	Description	Read/Write (Note 1)
15	100BASE-T4	1 = PHY able to perform 100BASE-T4 0 = PHY not able to perform 100BASE-T4	RO
14	100BASE-X Full Duplex	1 = PHY able to perform full-duplex 100BASE-X 0 = PHY not able to perform full-duplex 100BASE-X	RO
13	100BASE-X Half Duplex	1 = PHY able to perform half-duplex 100BASE-X 0 = PHY not able to perform half-duplex 100BASE-X	RO
12	10 Mbps Full Duplex	1 = PHY able to operate at 10 Mbps full-duplex mode 0 = PHY not able to operate at 10 Mbps full-duplex mode	RO
11	10 Mbps Half Duplex	1 = PHY able to operate at 10 Mbps half-duplex mode 0 = PHY not able to operate at 10 Mbps half-duplex mode	RO
10-7	Reserved	Ignore when read	RO
6	MF Preamble Suppression	1 = PHY will accept management frames with preamble suppressed 0 = PHY not able to accept management frames with preamble suppressed	RO
5	Auto-Negotiation Complete	1 = Auto-Negotiation process completed 0 = Auto-Negotiation process not completed	RO
4	Remote Fault	1 = remote fault condition detected 0 = no remote fault condition detected	RO, LH
3	Auto-Negotiation Ability	1 = PHY is able to perform Auto-Negotiation 0 = PHY is not able to perform Auto-Negotiation	RO
2	Link Status	1 = link is up 0 = link is down	RO, LL
1	Jabber Detect	1 = jabber condition detected 0 = no jabber condition detected	RO
0	Extended Capability	1 = extended register capabilities 0 = basic register set capabilities only	RO

Note:

1. RO = Read Only, LH = Latching High, LL = Latching Low.

Downloaded from Arrow.com.

Auto-Negotiation Advertisement Register (Register 4)

The purpose of this register is to advertise the technology ability to the link partner device. See Table 4.

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

Table B-4. Auto-Negotiation Advertisement Register (Register 4)

Bit(s)	Name	Description	Read/ Write
15	Next Page	When set, the device wishes to engage in next page exchange. If clear, the device does not wish to engage in next page exchange	R/W
14	Reserved	Ignore when read	RO
13	Remote Fault	When set, a remote fault bit is inserted into the base link code word during the Auto Negotiation process. When cleared, the base link code word will have the bit position for remote fault as cleared	R/W
12:5	Technology Ability	Technology ability field	R/W
4:0	Selector Field	Selector field	RO

Technology Ability Field Bit Assignments

The technology bit field consists of bits A0-A8 in the IEEE 802.3 Selector Base Page. Table 5 summarizes the bit assignments.

Table B-5. Technology Ability Field Bit Assignments

Bit	Technology
A0	10BASE-T
A1	10BASE-T Full Duplex
A2	100BASE-TX
A3	100BASE-TX Full Duplex
A4	100BASE-T4
A5	PAUSE operation for full duplex links
A6	Reserved for future technology
A7	Reserved for future technology

Downloaded from Arrow.com.

Auto-Negotiation Link Partner Ability Register (Register 5)

The Auto-Negotiation Link Partner Ability Register is Read Only. The register contains the advertised ability of the link partner. The bit definitions represent the received link code word. This register contains either the base page or the link partner's next pages. See Table 6.

Table B-6. Auto-Negotiation Link Partner Ability Register (Register 5) - Base Page Format

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

Downloaded from Arrow.com.

INDEX

Numerics

16-Bit Software Model 64
3.3 Vaux Presence Sense 28
32-bit multiplexed bus interface unit 3
32-Bit Software Model 65

A

Absolute Maximum Ratings 268
AD 25
Address and Data 25
Address Match Logic 229
Address Matching 73
Address Parity Error Response 40
Address PROM Space 109
Analog Power (1 Pin) 32
AP_VALUE0
Auto-Poll Value0 Register 124
AP_VALUE1
Auto-Poll Value1 Register 124
AP_VALUE2
Auto-Poll Value2 Register 125
AP_VALUE3
Auto-Poll Value3 Register 125
AP_VALUE4
Auto-Poll Value4 Register 125
AP_VALUE5
Auto-Poll Value5 Register 125
APDW Values 220
Automatic EEPROM Read Operation 98
Automatic Pad Generation 71
Automatic Pad Stripping 74
Automatic PREAD EEPROM Timing 277
Auto-Negotiation 87
Auto-Negotiation Advertisement Register
(Register 4) B-3
Auto-Negotiation With Multiple PHY Devices
88
Auto-Poll State Machine 86
AUTOPOLL0
Auto-Poll0 Register 125
AUTOPOLL1
Auto-Poll1 Register 126
AUTOPOLL2
Auto-Poll2 Register 127
AUTOPOLL3
Auto-Poll3 Register 128
AUTOPOLL4
Auto-Poll4 Register 129
AUTOPOLL5
Auto-Poll5 Register 130
AVDD 32

B

Basic Burst Read Transfer 42
Basic Burst Write Transfer 45
Basic Functions 33
Basic Non-Burst Read Transfer 41
Basic Non-Burst Write Transfer 43
BCR0
Master Mode Read Active 200
BCR1
Master Mode Write Active 200
BCR16
I/O Base Address Lower 209
BCR19
EEPROM Control and Status 211
BCR2
Miscellaneous Configuration 201
BCR20
Software Style 214
BCR28
Expansion Bus Port Address Lower
(Used for Flash/EPROM and
SRAM Accesses) 218
BCR29
Expansion Port Address Upper
(Used for Flash/EPROM Accesses) 218
BCR30
Expansion Bus Data Port Register 219
BCR31
Software Timer Register 219
BCR32
MII Control and Status Register 219
BCR33
MII Address Register 221
BCR34
MII Management Data Register 222
BCR35
PCI Vendor ID Register 222
BCR36
PCI Power Management Capabilities (PMC)
Alias Register 222

Downloaded from Arrow.com.

BCR37
PCI DATA Register Zero (DATA0)
Alias Register 223
BCR38
PCI DATA Register One (DATA1)
Alias Register 223
BCR39
PCI DATA Register Two (DATA2)
Alias Register 223
BCR4
LED 0 Status 201
BCR40
PCI Data Register Three (DATA3)
Alias Register 224
BCR41
PCI DATA Register Four (DATA4)
Alias Register 224
BCR42
PCI DATA Register Five (DATA5)
Alias Register 224
BCR43
PCI DATA Register Six (DATA6)
Alias Register 225
BCR44
PCI DATA Register Seven (DATA7)
Alias Register 225
BCR45
OnNow Pattern Matching Register #1 . . .225
BCR46
OnNow Pattern Matching Register #2 . . .226
BCR47
OnNow Pattern Matching Register #3 . . .226
BCR5
LED1 Status 203
BCR6
LED2 Status 205
BCR7
LED3 Status 207
BCR9
Full-Duplex Control 209
Block Diagram 4
Board Interface 27
Boot ROM Chip Select 30
Boundary Scan Circuit 105
Boundary Scan Register 106
BSR Mode Of Operation 106
Buffer Size Tuning A-7
Burst Write Transfer 46
Bus Acquisition 41, 42
Bus Command and Byte Enables 25
Bus Configuration Registers 199, 254, 266
Bus Grant 25
Bus Master DMA Transfers 41
Bus Request 26

C

C/BE 25
Carrier Sense 30
CHIPID
Chip ID Register 131
CHPOLLTIME
Chain Poll Timer Register 131
CLK 25
CLK Waveform for 3.3 V Signaling 274
CLK Waveform for 5 V Signaling 274
CLKSEL1 28
CLKSEL2 28
Clock 25
Clock Select 1 28
Clock Select 2 28
Clock Timing 271
CMD0
Command0 132
CMD2
Command2 133
CMD3
Command3 137
CMD7
Command7 140
COL 30
Collision 30
Collision Handling 70
command style access 124
Connection Diagram (PQR208) 19
Control and Status Registers 175, 250, 264
Control Register (Register 0) B-1
CRS 30
CSR0
Controller Status and Control Register . .175
CSR1
Initialization Block Address 0 177
CSR10
Logical Address Filter 2 186
CSR100
Bus Timeout 195
CSR101-111
Reserved 195
CSR11
Logical Address Filter 3 186

Downloaded from Arrow.com.

CSR112
Missed Frame Count 195
CSR113
Reserved 195
CSR114
Receive Collision Count 195
CSR116
OnNow Power Mode Register 196
CSR117-123
Reserved 197
CSR12
Physical Address Register 0 186
CSR124
Test Register 1 197
CSR125
MAC Enhanced Configuration Control . .197
CSR13
Physical Address Register 1 187
CSR14
Physical Address Register 2 187
CSR15
Mode 187
CSR16-23
Reserved Locations 188
CSR2
Initialization Block Address 1 177
CSR24
Base Address of Receive Ring Lower . . .189
CSR25
Base Address of Receive Ring Upper . . .189
CSR26-29
Reserved 189
CSR3
Interrupt Masks and Deferral Control . . .177
CSR30
Base Address of Transmit Ring Lower . .189
CSR31
Base Address of Transmit Ring Upper . .189
CSR32-46
Reserved 190
CSR4
Test and Features Control 179
CSR47
Transmit Polling Interval 190
CSR48
Receive Poll Time Counter 190
CSR49
Receive Polling Interval 190
CSR5
Extended Control and Interrupt 1 181
CSR50-57
Reserved 191
CSR58
Software Style 191
CSR59-75
Reserved 193
CSR6
RX/TX Descriptor Table Length 183
CSR7
Extended Control and Interrupt 2 183
CSR76
Receive Ring Length 193
CSR77
Reserved 193
CSR78
Transmit Ring Length 193
CSR79
Reserved 193
CSR8
Logical Address Filter 0 186
CSR80
DMA Transfer Counter and FIFO
Threshold Control 193,195
CSR81-87
Reserved 195
CSR88
Chip ID Register Lower 195
CSR89
Chip ID Register Upper 195
CSR9
Logical Address Filter 1 186
CSR90-99
Reserved 195
CTRL0
Control0 Register 141
CTRL1
Control1 Register 142
CTRL2
Control2 Register 145
CTRL3
Control3 Register 146
Cycle Frame 25

D

DATAMBIST
Memory BIST Access Register 147
DC Characteristics 269
Descriptor DMA Transfers 53
Descriptor Management 63

Downloaded from Arrow.com.

Descriptor Management Unit 61
Descriptor Ring Read In Burst Mode 55
Descriptor Ring Read In Non-Burst Mode ..54
Descriptor Ring Write In Burst Mode 58
Descriptor Ring Write In Non-Burst Mode ..57
Descriptor Rings 63
Destination Address Handling 69
Detailed Functions 34
Device Select 25
DEVSEL 25
Digital and I/O Buffer Power (24 Pins) 32
Digital Ground (8 Pins) 32
Digital I/O (Non-PCI Pins) 269
Direct Access to the Interface 100
Direct Flash Access 96
Disconnect Of Burst Transfer 39
Disconnect Of Slave Burst Transfer - Host Inserts Wait States 40
Disconnect Of Slave Burst Transfer - No Host Wait States 39
Disconnect Of Slave Cycle When Busy 39
Disconnect When Busy 38
Disconnect With Data Transfer 46, 47
Disconnect Without Data Transfer 47, 48
DMA Burst Alignment 46
Double Word I/O Mode 110
Dual-speed CSMA/CD 2

E

EAR 31
EECS 29
EEDET Setting 213
EEDI 29
EEDO 29
EEPROM Auto-Detection 100
EEPROM Chip Select 29
EEPROM CRC Calculation 100
EEPROM Data In 29
EEPROM Data Out 29
EEPROM Interface 29
EEPROM programmable pin (PHY_RST) ...3
EEPROM Read Functional Timing 276
EEPROM Serial Clock 29
EEPROM Timing 276
EEPROM_ACC EEPROM Access Register 150
EESK 29
ERA 29
ERADSP/CEN 30
ERADV 30
ERCE 30
ERCLK 30
ERD 30
EROE 30
Error Detection 69
ERWE/FLWE 30
Expansion Bus Interface 93
Expansion ROM - Boot Device Access 94
Expansion ROM Bus Read Sequence 95
Expansion ROM Output Enable 30
Expansion ROM Transfers 38
External Address Detection Interface 2, 31
External PHY - MII @ 2.5 MHz 283
External PHY - MII @ 25 MHz 283
Receive Frame Tagging 93
External Address Detection Interface (EADI) 23
External Address Reject 31
External Clock Input 28
External Memory Address 29
External Memory Address Advance 30
External Memory Address Strobe 30
External Memory Clock 30
External Memory Data 30
External Memory Interface 29
External Memory Write Enable 30
External SSRAM Chip Enable 30
External SSRAM Output Enable 30

F

FIFO Burst Write At End Of Unaligned Buffer 61
FIFO Burst Write At Start Of Unaligned Buffer 59
FIFO DMA Transfers 58
FLA 29
Flash Read from Expansion Bus Data Port 96
Flash/EPROM Read 96
FLASH_ADDR Flash Address Register 152
FLASH_DATA Flash Data Register 152
FLCS 30
FLOE 30

Downloaded from Arrow.com.

FLOW
Flow Control Register 153
Flow, LAPP A-2
FMDC Values 220
FRAME 25
Frame Format at the MII Interface Connection 85
Framing 68
Full-Duplex Link Status LED Support 83
Full-Duplex Operation 82

G

General Description 3
GNT 25

H

H_RESET 106

I

I/O Buffer Ground (25 Pins) 32
I/O Map In DWord I/O Mode (DWIO = 1) 111
I/O Registers 109
I/O Resources 108
IDSEL 25
IEEE 1149.1 (1990) Test Access Port Interface 32, 105
IEEE 1149.1 Test Access Port Interface (JTAG) 24
IEEE 802.3 Frame And Length Field Transmission Order 75
IFS1
Inter-Frame Spacing Part 1 Register 154
Initialization 61
Initialization Block 227
Initialization Block (SSIZE32 = 0) 227
Initialization Block (SSIZE32 = 1) 227
Initialization Block DMA Transfers 53
Initialization Device Select 25
Initiator Ready 26
Input Setup and Hold Timing 274
Instruction Register and Decoding Logic . .106
INT0
Interrupt0 155
INTA 25
INTEN0
Interrupt0 Enable 157
Interface Pin Assignment 151, 213
Interrupt Request 25
IPG
Inter-Packet Gap Register 159
IRDY 26

J

JTAG (IEEE 1149.1) TCK Waveform for 5 V Signaling 278
JTAG (IEEE 1149.1) Test Signal Timing 278, 279

K

Key to Switching Waveforms 272

L

LAPP 3 Buffer Grouping A-5
LAPP 3 Buffer Grouping for Two-Interrupt Method A-10
LAPP Flow
Two-Interrupt Method A-8
LAPP Timeline A-4
LAPP Timeline for Two-Interrupt Method A-9
Late Collision 73
LED Control Logic 101
LED Default Configuration 101
LED Support 100
LED0 27
LED0 Control Register 160
LED1 27
LED1 Control Register 161
LED2 27
LED2 Control Register 161
LED3 27
LED3 Control Register 161
Legal I/O Accesses in Double Word I/O Mode (DWIO =1) 111
Legal I/O Accesses in Word I/O Mode (DWIO = 0) 111
Logical Address Filter Register 159
Look-Ahead Packet Processing 2
Look-Ahead Packet Processing (LAPP) Concept A-1
Loopback Operation 82
Loss of Carrier 73

M

MAC Control Pause Frames 89
Magic Packet application 3
Management Cycle Timing 280
Management Data Clock 31
Management Data I/O 31

Downloaded from Arrow.com.

Management Data Output Valid Delay Timing 282
Management Data Setup and Hold Timing 282
Master Abort 50, 52
Master Bus Interface Unit 41
Master Cycle Data Parity Error Response . . .53
Master Initiated Termination 49
MAX_LAT_A PCI Maximum Latency Alias Register . .162
MDC 31
MDC Waveform 281
MDIO 31
Media Access Control 68
Media Access Management 69
Media Independent Interface 30, 83
Media Independent Interface (MII) 1, 23
Medium Allocation 69
Memory-Mapped Registers 124
MIB Offset 124
MII Management Control Register (Register 0) B-1
MII Management Frames 84
MII Management Interface 84
MII Management Registers B-1
MII Network Status Interface 84
MII Receive Frame Tagging 93
MII Receive Interface 83
MII Transmit Interface 83
MIN_GNT_A PCI Minimum Grant Alias Register 162
Miscellaneous Loopback Features 82
Mode 228

N

Network Interface 33
Network Port Manager 87
Non-Burst Read Transfer 43
Non-Burst Write Transfer 45
Normal and Tri-State Outputs 273

O

OnNow Functional Diagram 102
OnNow Wake-Up Sequence 102
Operating Ranges 268
Operation Without MMI Management Interface 88
Other Data Registers 106
Outline of LAPP Flow A-2
Output and Float Delay Timing 271
Output Tri-state Delay Timing 275
Output Valid Delay Timing 275

P

PADR 228
Physical Address Register 162
PAL function 3
PAR 26
Parity 26
Parity Error 26
Parity Error Response 40, 52
Pattern Match RAM 105
Pause Count Register 163
PCI Base-Class Register Offset 0Bh 116
PCI Bus Interface Pins - 3.3 V Signaling . .269
PCI Bus Interface Pins - 5 V Signaling 269
PCI Cache Line Size Register Offset 0Ch 117
PCI Capabilities Pointer Register Offset 34h 120
PCI Capability Identifier Register Offset 44h 121
PCI Command Register 113
PCI Command Register Offset 04h 113
PCI Configuration Alias registers 113
PCI Configuration Registers 108, 113, 247
PCI Configuration Space Layout 108
PCI Data Register Offset 4Bh 123
PCI Device ID Register Offset 02h 113
PCI Expansion ROM Base Address Register Offset 30h 119
PCI Header Type Register Offset 0Eh 117
PCI Interface 25
PCI Interrupt Line Register Offset 3Ch . . .120
PCI Interrupt Pin Register Offset 3Dh 120
PCI Latency Timer Register Offset 0Dh . . .117
PCI MAX_LAT Register Offset 3Fh 121
PCI Memory Mapped I/O Base Address Register Offset 14h 118
PCI MIN_GNT Register Offset 3Eh 120
PCI Next Item Pointer Register Offset 45h 121
PCI PMCSR Bridge Support Extensions Register Offset 4Ah 123
PCI Power Management Capabilities Register (PMC) Offset 46h 121
PCI Power Management Control/Status Register (PMCSR) Offset 48h 122
PCI Programming Interface Register Offset 09h 116

Downloaded from Arrow.com.

PCI Revision ID Register Offset 08h 116
PCI Status Register Offset 06h 115
PCI Sub-Class Register Offset 0Ah 116
PCI Subsystem ID Register Offset 2Eh 119
PCI Subsystem Vendor ID
Register Offset 2Ch 118
PCI Vendor ID Register 113
PCI Vendor ID Register Offset 00h 113
PCIDATA0
PCI DATA Register Zero
Alias Register 163
PCIDATA1
PCI DATA Register One
Alias Register 163
PCIDATA2
PCI DATA Register Two
Alias Register 164
PCIDATA3
PCI DATA Register Three
Alias Register 164
PCIDATA4
PCI DATA Register Four
Alias Register 164
PCIDATA5
PCI DATA Register Five
Alias Register 164
PCIDATA6
PCI DATA Register Six
Alias Register 164
PCIDATA7
PCI DATA Register Seven
Alias Register 164
Peripheral Component Interconnect
(PCI) bus 1
PERR 26
PG 27
PHY Access Register 165
PHY Reset 29
PHY_RST 29
Physical Dimensions 287
Pin Capacitance 269
Pin Descriptions 25
Pin Designations by Group 22
Pin Designations by Pin Number
(PQR208) 20
PMAT0
OnNow Pattern Register 0 166
PMAT1
OnNow Pattern Register1 166
PMC_A
PCI Power Management Capabilities
Alias Register 167
PME 27
Polling 65
Power Good 27
Power Management Event 27
Power Management Interface 23
Power Management Support 101
Power on Reset 107
Power Savings Mode 101
Power Supplies 24
Power Supply Current 270
Power Supply Pins 32
PQR208 Plastic Quad Flat Pack
(measured in millimeters) 287
Preemption During Burst Transaction ...49, 51
Preemption During Non-Burst
Transaction 49, 50
Programmable Inter Packet Gap (IPG) 2
Programmable Registers 264

R

RAP
Register Address Port 174
RAP Register 174
RCV_RING_LEN
Receive Ring Length Register 167
RDRA and TDRA 228
Receive Address Match 74
Receive Clock 31
Receive Data 31
Receive Data Valid 31
Receive Descriptor (SWSTYLE = 0) 229
Receive Descriptor (SWSTYLE = 2) 229
Receive Descriptor (SWSTYLE = 3) 230
Receive Descriptor (SWSTYLE = 4) 230
Receive Descriptor (SWSTYLE = 5) 230
Receive Descriptors 229
Receive Error 31
Receive Exception Conditions 75
Receive FCS Checking 75
Receive Frame Tag Data 32
Receive Frame Tag Enable 32
Receive Frame Tag Timing with Media
Independent Interface 283, 285
Receive Function Programming 73
Receive Operation 73
Receive Polling 67
Receive Protect Register 167

Downloaded from Arrow.com.

Receive Ring Base Address Register 130
Receive Timing 280, 281
Receive Watermark Programming 194
Register Bit Cross Reference 255
Register initialization 113
Register Programming Summary 264
Register Summary 247
Regulating Network Traffic 89
Re-Initialization 62
Reject Timing - External PHY MII @ 2.5 MHz 284
Reject Timing - External PHY MII @ 25 MHz 283
Remote Wake Up 27
REQ 26
Reset 26
Reset Register 109
RLEN and TLEN 228
ROM_CFG
ROM Base Address Configuration Register 168
RST 26
Run and Suspend 62
Running registers 113
RWU 27
RX_CLK 31
RX_DV 31
RX_ER 31
RXD 31
RXFRTGD 32
RXFRTGE 32

S

S_RESET 107
SERR 26
Setup A-2
Setup and Hold Timing 271
Setup registers 113
SFBD 31
SID_A
PCI Subsystem ID Alias Register 168
Slave Bus Interface Unit 34
Slave Configuration Read 35
Slave Configuration Transfers 34
Slave Configuration Write 35
Slave Cycle Data Parity Error Response 41
Slave I/O Transfers 34
Slave Read Using I/O Command 36
Slave Write Using Memory Command 37
Software 216
Software Access 108
Software Interface 33
Software Interrupt Timer 68
Software Timer Value Register 171
Some Examples of LAPP Descriptor Interaction A-6
SR2
Initialization Block Address 1 177
SRAM Boundary Register 169
SRAM Configuration 97
SRAM Size Register 169
Start Frame-Byte Delimiter 31
STAT0
Status0 170
Status Register (Register 1) B-2
STOP 26, 107
Stop 26
Supported Instructions 105
SVID_A
PCI Subsystem Vendor ID Alias Register 172
Switching Characteristics
Bus Interface 271
EEPROM Interface 276
External Address Detection Interface 283, 284
JTAG Timing 278
Media Independent Interface 280
Switching Test Circuits 273
Switching Waveforms 272
Expansion Bus Interface 286
Receive Frame Tag 285
System Bus Interface 33
System Error 26

T

TAP Finite State Machine 105
Target Abort 48, 49
Target Initiated Termination 46
Target Ready 26
TCK 32
TDI 32
TDO 32
Technology Ability Field Bit Assignments B-3
TEST 28
Test Clock 32
Test Data In 32
Test Data Out 32
Test Mode Select 32

Downloaded from Arrow.com.

Test Reset 28
TMS 32
Transmit and Receive Message Data Encapsulation 68
Transmit Clock 30
Transmit Data 30
Transmit Descriptors 240
Transmit Enable 30
Transmit Error Conditions 72
Transmit FCS Generation 72
Transmit Function Programming 71
Transmit Operation 71
Transmit Polling 66
Transmit Ring Base Address Register 131
Transmit Start Point Programming 194
Transmit Timing 280, 281
Transmit Watermark Programming 194
TRDY 26
TX_CLK 30
TX_EN 30
TXD 30

U

User Accessible Registers 113

V

VAUX_SENSE 28
VDD 32
VID_A PCI Vendor ID Alias Register 173
VLAN Support 80
VSS 32
VSSB 32

W

Wake-Up Mode Indicator 28
Word I/O Mode 110
WUMI 28

X

XCLK 28
XMT_RING_LEN Transmit Ring Length Register 173
XMTPOLLTIME Transmit Poll Timer Register 173
XTAL1 28
XTAL2 29

Downloaded from Arrow.com.

The contents of this document are provided in connection with Advanced Micro Devices, Inc. (“AMD”) products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD’s Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD’s products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD’s product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Trademarks

Copyright © 1999, 2000 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD logo, and combinations thereof are registered trademarks of Advanced Micro Devices, Inc.

AlertIT, any1Home, eIMR, eIMR+, GigaPHY, HIMIB, HomePHY, IMR2, MACE, Magic Packet, NetPHY, PCnet, PCnet-Home, QuEST, and QuIET are trademarks of Advanced Micro Devices, Inc.

RLL25 is a trademark of Tut Systems, Inc.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

22929D

Downloaded from Arrow.com.