Digital Semiconductor 21140A
PCI Fast Ethernet LAN Controller

Hardware Reference Manual

Order Number: EC–QN7NC–TE

January 1996

While Digital believes the information included in this publication is correct as of the date of publication, it is subject to change without notice.

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

© Digital Equipment Corporation 1995, 1996. All rights reserved.
Printed in U.S.A.

DEC, DEChip, DNA, Digital, Digital Semiconductor, VAX DOCUMENT, and the DIGITAL logo are trademarks of Digital Equipment Corporation.

Digital Semiconductor is a Digital Equipment Corporation business.

IEEE is a registered trademark of The Institute of Electrical and Electronics Engineers, Inc. MicroWire is a registered trademark of BankAmerican Corporation.

All other trademarks and registered trademarks are the property of their respective owners.

This document was prepared using VAX DOCUMENT Version 2.1.
Contents

Preface .......................................................... xiii

1 Introduction
   1.1 General Description .................................... 1–1
       1.1.1 Features ........................................ 1–2
   1.2 Hardware Overview ................................... 1–3

2 Signal Descriptions
   2.1 21140A Pinout ....................................... 2–1
   2.2 Signal Descriptions ................................... 2–3

3 Registers
   3.1 Configuration Operation ............................... 3–1
       3.1.1 Configuration Register Mapping ................. 3–1
       3.1.2 Configuration Registers ......................... 3–2
           3.1.2.1 Configuration ID Register (CFID–Offset 00H) . 3–2
           3.1.2.2 Command and Status Configuration Register (CFCS–Offset 04H) . 3–3
           3.1.2.3 Configuration Revision Register (CFRV–Offset 08H) . . . . . 3–6
           3.1.2.4 Configuration Latency Timer Register (CFLT–Offset 0CH) . . . . . 3–7
           3.1.2.5 Subsystem ID Register (SSID–2CH) ......... 3–10
           3.1.2.6 Expansion ROM Base Address Register (CBER–Offset 30H) . . . . . 3–11
           3.1.2.7 Configuration Interrupt Register (CFIT–Offset 3CH) . . . . . 3–12
3.1.2.10 Configuration Device and Driver Area Register
(CFDD-Offset 40H) .................................... 3–14
3.2 CSR Operation .................................... 3–15
3.2.1 Control and Status Register Mapping ................. 3–16
3.2.2 Host CSRs .................................... 3–17
3.2.2.1 Bus Mode Register (CSR0-Offset 00H) .......... 3–17
3.2.2.2 Transmit Poll Demand Register (CSR1-Offset 08H) ... 3–20
3.2.2.3 Receive Poll Demand Register (CSR2-Offset 10H) .... 3–21
3.2.2.4 Descriptor List Address Registers
(CSR3-Offset 18H and CSR4-Offset 20H) .................. 3–21
3.2.2.5 Status Register (CSR5-Offset 28H) ................ 3–23
3.2.2.6 Operation Mode Register (CSR6-Offset 30H) ....... 3–28
3.2.2.7 Interrupt Enable Register (CSR7-Offset 38H) ....... 3–37
3.2.2.8 Missed Frames and Overflow Counter (CSR8-Offset 40H)
.................................................. 3–41
3.2.2.9 Boot ROM, Serial ROM, and MII Management Register
(CSR9-Offset 48H) .................................... 3–42
3.2.2.10 Boot ROM Programming Address Register
(CSR10-Offset 50H) ................................... 3–45
3.2.2.11 General-Purpose Timer Register (CSR11-Offset 58H) .. 3–46
3.2.2.12 General-Purpose Port Register (CSR12-Offset 60H) ... 3–47
3.2.2.13 Watchdog Timer Register (CSR15-Offset 78H) ....... 3–49

4 Host Communication
4.1 Data Communication .................................. 4–1
4.2 Descriptor Lists and Data Buffers ....................... 4–1
4.2.1 Receive Descriptors ................................ 4–2
4.2.1.1 Receive Descriptor 0 (RDES0) ................... 4–3
4.2.1.2 Receive Descriptor 1 (RDES1) ................... 4–8
4.2.1.3 Receive Descriptor 2 (RDES2) ................... 4–9
4.2.1.4 Receive Descriptor 3 (RDES3) ................... 4–10
4.2.1.5 Receive Descriptor Status Validity ............... 4–10
4.2.2 Transmit Descriptors ............................ 4–11
4.2.2.1 Transmit Descriptor 0 (TDES0) ................ 4–12
4.2.2.2 Transmit Descriptor 1 (TDES1) ................ 4–14
4.2.2.3 Transmit Descriptor 2 (TDES2) ................ 4–17
4.2.2.4 Transmit Descriptor 3 (TDES3) ................ 4–18
4.2.2.5 Transmit Descriptor Status Validity ............ 4–18
4.2.3 Setup Frame ........................................... 4–19
4.2.3.1 First Setup Frame ................................ 4–20
4.2.3.2 Subsequent Setup Frames ....................... 4–20
4.2.3.3 Perfect Filtering Setup Frame Buffer .......... 4–20
4.2.3.4 Imperfect Filtering Setup Frame Buffer ........ 4–22
4.3 Functional Description ................................ 4–28
4.3.1 Reset Commands .................................. 4–29
4.3.2 Arbitration Scheme ............................... 4–29
4.3.3 Interrupts ........................................ 4–31
4.3.4 Startup Procedure ............................... 4–32
4.3.5 Receive Process .................................. 4–33
4.3.5.1 Descriptor Acquisition ......................... 4–33
4.3.5.2 Frame Processing ............................... 4–33
4.3.5.3 Receive Process Suspended ..................... 4–34
4.3.5.4 Receive Process State Transitions ............... 4–34
4.3.6 Transmit Process .................................. 4–35
4.3.6.1 Frame Processing ............................... 4–36
4.3.6.2 Transmit Polling Suspended .................... 4–36
4.3.6.3 Transmit Process State Transitions ............. 4–37

5 Host Bus Operation

5.1 Overview ........................................... 5–1
5.2 Bus Commands ...................................... 5–2
5.3 Bus Slave Operation ............................... 5–2
5.3.1 Slave Read Cycle (I/O or Memory Target) ....... 5–3
5.3.2 Slave Write Cycle (I/O or Memory Target) ....... 5–4
5.3.3 Configuration Read and Write Cycles ............... 5–5
5.4 Bus Master Operation ............................... 5–6
5.4.1 Bus Arbitration ................................. 5–7
5.4.2 Memory Read Cycle ............................... 5–8
5.4.3 Memory Write Cycle ............................... 5–9
5.5 Termination Cycles ................................. 5–10
5.5.1 Slave-Initiated Termination ....................... 5–10
5.5.2 Master-Initiated Termination ..................... 5–11
5.5.2.1 21140A-Initiated Termination ................ 5–12
5.5.2.1.1 Normal Completion .......................... 5–12
5.5.2.1.2 Timeout .................................... 5–12
5.5.2.1.3 Master Abort ................................ 5–13
5.5.2.2 Memory-Controller-Initiated Termination .... 5–13
5.5.2.2.1 Target Abort ................................. 5–13
5.5.2.2.2 Target Disconnect Termination ............... 5–14
5.5.2.2.3 Target Retry ................................. 5–15
6 Network Interface Operation

6.1 MII/SYM Port .............................................. 6–1
6.1.1 100BASE-T Terminology ............................ 6–1
6.1.2 Interface Description ............................... 6–2
6.1.2.1 Signal Standards ................................ 6–2
6.1.2.2 Operating Modes .............................. 6–4
6.2 Serial Port ............................................. 6–5
6.3 Media Access Control Operation .................. 6–5
6.3.1 MAC Frame Format .................................. 6–7
6.3.1.1 Ethernet and IEEE 802.3 Frames .............. 6–8
6.3.1.2 Ethernet Frame Format Description .......... 6–8
6.3.2 Ethernet Reception Addressing .................. 6–10
6.3.3 Detailed Transmit Operation ..................... 6–11
6.3.3.1 Transmit Initiation ............................. 6–12
6.3.3.2 Frame Encapsulation ........................... 6–12
6.3.3.3 Initial Deferral ................................ 6–13
6.3.3.4 Collision ...................................... 6–13
6.3.3.5 Terminating Transmission .................... 6–14
6.3.3.6 Transmit Parameter Values ................... 6–15
6.3.4 Detailed Receive Operation ....................... 6–15
6.3.4.1 Receive Initiation .............................. 6–15
6.3.4.2 Preamble Processing ........................... 6–16
6.3.4.2.1 MII or Serial Mode Preambles ............. 6–16
6.3.4.2.2 100BASE-TX or 100BASE-FX Mode Preambles .... 6–17
6.3.4.3 Address Matching .............................. 6–17
6.3.4.4 Frame Decapsulation ........................... 6–17
6.3.4.5 Terminating Reception ......................... 6–18
6.3.4.6 Frame Reception Status ....................... 6–18
6.4 Loopback Operations .................................... 6–19
6.4.1 Internal Loopback Mode ......................... 6–19
6.4.2 External Loopback Mode ......................... 6–19
6.5 Full-Duplex Operation ................................ 6–20
6.6 Capture Effect–A Value-Added Feature .......... 6–20
6.6.1 What Is Capture Effect? ......................... 6–21
6.6.2 Resolving Capture Effect ......................... 6–22
6.6.3 Enhanced Resolution for Capture Effect ........ 6–23
6.7 Power-Saving Modes ................................... 6–23
6.7.1 Sleep Power-Saving Mode ....................... 6–23
6.7.2 Snooze Power-Saving Mode ..................... 6–24
7 External Ports

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Overview</td>
<td>7–1</td>
</tr>
<tr>
<td>7.2</td>
<td>Boot ROM and Serial ROM Connection</td>
<td>7–1</td>
</tr>
<tr>
<td>7.3</td>
<td>Boot ROM Operations</td>
<td>7–3</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Byte Read</td>
<td>7–3</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Byte Write</td>
<td>7–4</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Dword Read</td>
<td>7–5</td>
</tr>
<tr>
<td>7.4</td>
<td>Serial ROM Operations</td>
<td>7–7</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Read Operation</td>
<td>7–7</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Write Operation</td>
<td>7–10</td>
</tr>
<tr>
<td>7.5</td>
<td>External Register Operation</td>
<td>7–13</td>
</tr>
<tr>
<td>7.6</td>
<td>General-Purpose Port Register (CSR12)</td>
<td>7–14</td>
</tr>
<tr>
<td>7.7</td>
<td>LED Support</td>
<td>7–14</td>
</tr>
</tbody>
</table>

A Joint Test Action Group—Test Logic

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>General Description</td>
<td>A–1</td>
</tr>
<tr>
<td>A.2</td>
<td>Registers</td>
<td>A–2</td>
</tr>
<tr>
<td>A.2.1</td>
<td>Instruction Register</td>
<td>A–2</td>
</tr>
<tr>
<td>A.2.2</td>
<td>Bypass Register</td>
<td>A–3</td>
</tr>
<tr>
<td>A.2.3</td>
<td>Boundary-Scan Register</td>
<td>A–3</td>
</tr>
<tr>
<td>A.2.4</td>
<td>Test Access Port Controller</td>
<td>A–4</td>
</tr>
</tbody>
</table>

B DNA CSMA/CD Counters and Events Support

C Hash C Routine

D Technical Support and Ordering Information

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.1</td>
<td>Technical Support</td>
<td>D–1</td>
</tr>
<tr>
<td>D.2</td>
<td>Ordering Digital Semiconductor Products</td>
<td>D–1</td>
</tr>
<tr>
<td>D.3</td>
<td>Ordering Associated Literature</td>
<td>D–2</td>
</tr>
<tr>
<td>D.4</td>
<td>Ordering Third-Party Literature</td>
<td>D–2</td>
</tr>
</tbody>
</table>
Index

Examples

4–1  Perfect Filtering Buffer ........................................... 4–22
4–2  Imperfect Filtering Buffer ........................................... 4–25

Figures

1–1  21140A Block Diagram ........................................... 1–5
2–1  21140A Pinout Diagram (Top View) ............................. 2–2
3–1  CFID Configuration ID Register ................................... 3–2
3–2  CFCS Command and Status Configuration Register ........... 3–4
3–3  CFRT Configuration Revision Register .......................... 3–6
3–4  CFLT Configuration Latency Timer Register ................... 3–7
3–5  CBIO Configuration Base I/O Address Register ............... 3–9
3–6  CBMA Configuration Base Memory Address Register .......... 3–10
3–7  SSID Subsystem ID Register ........................................ 3–11
3–8  CBER Configuration Base Address Register .................... 3–12
3–9  CFIT Configuration Interrupt Register ........................... 3–13
3–10 CFDD Configuration Driver Area Register ....................... 3–14
3–11 CSR0 Bus Mode Register ........................................... 3–17
3–12 CSR1 Transmit Poll Demand Register ............................ 3–20
3–13 CSR2 Receive Poll Demand Register .............................. 3–21
3–14 CSR3 Transmit List Base Address Register ...................... 3–22
3–15 CSR4 Transmit List Base Address Register ...................... 3–23
3–16 CSR5 Status Register .............................................. 3–24
3–17 CSR6 Operating Mode Register .................................... 3–29
3–18 CSR7 Interrupt Enable Register ................................... 3–38
3–19 CSR8 Missed Frames and Overflow Counter .................... 3–41
3–20 CSR9 Boot ROM, Serial ROM, and MII Management Register ........................................... 3–43
3–21 CSR10 Boot ROM Programming Address Register .............. 3–45
3–22 CSR11 General-Purpose Timer Register .......................... 3–46
3–23 CSR12 General-Purpose Port Register ............................ 3–47
3–24 CSR15 Watchdog Timer Register ................................... 3–49
4–1  Descriptor Ring and Chain Structure Examples ................. 4–2
### Tables

<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Functional Description of 21140A Signals</td>
</tr>
<tr>
<td>3-1</td>
<td>Configuration Registers Mapping</td>
</tr>
<tr>
<td>3-2</td>
<td>CFID Configuration ID Register Description</td>
</tr>
<tr>
<td>3-3</td>
<td>CFID Access Rules</td>
</tr>
<tr>
<td>3-4</td>
<td>CFCS Command and Status Configuration Register Description</td>
</tr>
<tr>
<td>3-5</td>
<td>CFCS Access Rules</td>
</tr>
<tr>
<td>3-6</td>
<td>CFRV Configuration Revision Register Description</td>
</tr>
<tr>
<td>3-7</td>
<td>CFRV Access Rules</td>
</tr>
<tr>
<td>3-8</td>
<td>CFLT Configuration Latency Timer Register Description</td>
</tr>
<tr>
<td>3-9</td>
<td>CFLT Access Rules</td>
</tr>
<tr>
<td>3-10</td>
<td>CBIO Configuration Base I/O Address Register Description</td>
</tr>
<tr>
<td>3-11</td>
<td>CBIO Access Rules</td>
</tr>
<tr>
<td>3-12</td>
<td>CBMA Configuration Base Memory Address Register Description</td>
</tr>
<tr>
<td>3-13</td>
<td>CBMA Access Rules</td>
</tr>
<tr>
<td>3-14</td>
<td>SSID Subsystem ID Register Description</td>
</tr>
<tr>
<td>3-15</td>
<td>SSID Access Rules</td>
</tr>
<tr>
<td>3-16</td>
<td>CBER Expansion ROM Base Address Register Description</td>
</tr>
<tr>
<td>3-17</td>
<td>CBER Access Rules</td>
</tr>
<tr>
<td>3-18</td>
<td>CFIT Configuration Interrupt Register Description</td>
</tr>
<tr>
<td>3-19</td>
<td>CFIT Access Rules</td>
</tr>
<tr>
<td>3-20</td>
<td>CFDD Configuration Driver Area Register Description</td>
</tr>
<tr>
<td>3-21</td>
<td>CFDD Access Rules</td>
</tr>
<tr>
<td>3-22</td>
<td>CSR Mapping</td>
</tr>
<tr>
<td>3-23</td>
<td>CSR0 Bus Mode Register Description</td>
</tr>
<tr>
<td>3-24</td>
<td>Transmit Automatic Polling Bits</td>
</tr>
<tr>
<td>3-25</td>
<td>CSR0 Access Rules</td>
</tr>
<tr>
<td>3-26</td>
<td>Cache Alignment Bits</td>
</tr>
<tr>
<td>3-27</td>
<td>CSR1 Transmit Poll Demand Register Description</td>
</tr>
<tr>
<td>3-28</td>
<td>CSR1 Access Rules</td>
</tr>
</tbody>
</table>
3–29 CSR2 Receive Poll Demand Register Description .......... 3–21
3–30 CSR2 Access Rules ........................................ 3–21
3–31 CSR3 Receive List Base Address Register Description .... 3–22
3–32 CSR3 Access Rules ........................................ 3–22
3–33 CSR4 Transmit List Base Address Register Description ... 3–23
3–34 CSR4 Access Rules ........................................ 3–23
3–35 CSR5 Status Register Description ......................... 3–24
3–36 Fatal Bus Error Bits ........................................ 3–27
3–37 Transmit Process State .................................... 3–27
3–38 Receive Process State ...................................... 3–28
3–39 CSR5 Access Rules ........................................ 3–28
3–40 CSR6 Operating Mode Register Description ............... 3–30
3–41 Transmit Threshold ........................................ 3–35
3–42 Port and Data Rate Selection ................................ 3–35
3–43 Loopback Operation Mode ................................ 3–36
3–44 Filtering Mode .............................................. 3–36
3–45 CSR6 Access Rules ........................................ 3–37
3–46 CSR7 Interrupt Enable Register Description .............. 3–38
3–47 CSR7 Access Rules ........................................ 3–41
3–48 CSR8 Missed Frames and Overflow Counter Description . 3–42
3–49 CSR8 Access Rules ........................................ 3–42
3–50 CSR9 Boot ROM, Serial ROM, and MII Management ... 3–43
3–51 CSR9 Access Rules ........................................ 3–45
3–52 CSR10 Boot ROM Programming Address Description ...... 3–45
3–53 CSR10 Access Rules ........................................ 3–45
3–54 CSR11 General-Purpose Timer Register Description ...... 3–46
3–55 CSR11 Access Rules ........................................ 3–47
3–56 CSR12 General-Purpose Port Register Description ....... 3–48
3–57 CSR12 Access Rules ........................................ 3–48
3–58 CSR15 Watchdog Timer Register Description .............. 3–50
3–59 CSR15 Access Rules ........................................ 3–51
4–1 RDES0 Receive Descriptor 0 Description .................. 4–5
4–2 RDES1 Receive Descriptor 1 Description .................. 4–9
4–3 RDES2 Receive Descriptor 2 Description .................. 4–9
4–4 RDES3 Receive Descriptor 3 Description .................. 4–10
4–5 Receive Descriptor Status Validity .......................... 4–11
Preface

Purpose and Audience

The Digital Semiconductor 21140A PCI Fast Ethernet LAN Controller Hardware Reference Manual describes the operation of the Digital Semiconductor 21140A PCI Fast Ethernet LAN Controller (also referred to as the 21140A). This manual is for designers who use the 21140A.

Manual Organization

This manual contains seven chapters, four appendixes, and an index.

• Chapter 1, Introduction, includes a general description of the 21140A. It also provides an overview of the 21140A hardware components.
• Chapter 2, Signal Descriptions, provides the physical layout of the 21140A and describes each of the input and output signals.
• Chapter 3, Registers, provides a complete bit description of the 21140A control and status registers (CSRs) and the configuration registers.
• Chapter 4, Host Communication, describes how the 21140A communicates with the host by using descriptor lists and data buffers. It also describes the transmit and receive processes.
• Chapter 5, Host Bus Operation, provides a description of the read, write, and termination cycles.
• Chapter 6, Network Interface Operation, describes the MII/SYM port and serial port interfaces. It includes a complete description of media access control (MAC) operations. It also provides detailed transmitting and receiving operation information.
• Chapter 7, External Ports, describes the interface and operation of the MicroWire serial ROM, the boot ROM, the general-purpose port, and the network activity LEDs.
• Appendix A, Joint Test Action Group—Test Logic, provides descriptions of the testing, the observing, and the modifying of circuit activity during normal operation.

• Appendix B, DNA CSMA/CD Counters and Events Support, describes features that support the driver in implementing and reporting the specified counters and events.

• Appendix C, Hash C Routine, provides an example of a C routine that generates a hash index for a given Ethernet address.

• Appendix D, Technical Support and Ordering Information, contains technical support and ordering information.

**Document Conventions**

The values 1, 0, and X are used in some tables. X signifies a don't care (1 or 0) state, which can be determined by the system designer.
This chapter provides a general description of the 21140A and its features. The chapter also includes an overview of the 21140A hardware components.

1.1 General Description

The 21140A is a fast Ethernet LAN controller for both 100Mb/s and 10Mb/s data rates, providing a direct interface to the peripheral component interconnect (PCI) local bus. The 21140A interfaces to the host processor by using onchip control and status registers (CSRs) and a shared host memory area, set up mainly during initialization. This minimizes processor involvement in the 21140A operation during normal reception and transmission. Large FIFOs allow the 21140A to efficiently operate in systems with longer latency periods. Bus traffic is also minimized by filtering out received runt frames and by automatically retransmitting collided frames without a repeated fetch from the host memory.

The 21140A provides two network ports: a 10Mb/s port and a 10/100Mb/s port. The 10Mb/s port provides a conventional 7-wire interface for the existing 10Mb/s front-end decoder (ENDEC).

The 10/100Mb/s port can be programmed to support various levels of interconnect. It can be programmed to support either full media-independent interface (MII) functionality or 100BASE-X physical coding sublayer (PCS), which includes 4B/5B encoder/decoder, framer, and scrambler/descrambler.

The 21140A can sustain transmission or reception of minimal-sized back-to-back packets at full line speed with an interpacket gap (IPG) of 9.6 µs for 10Mb/s and 0.96 µs for 100Mb/s. The 21140A has several additional features that are not available on the 21140, yet it remains pin and software compatible with the 21140.
1.1 General Description

1.1.1 Features

The 21140A has the following features:

- Offers a single-chip Fast Ethernet controller for PCI local bus:
  - Provides a glue-less connection to the PCI bus
  - Supports two network ports: 10Mb/s and 10/100Mb/s
- Provides a standard 10/100Mb/s MII supporting CAT3 unshielded twisted-pair (UTP), CAT5 UTP, shielded twisted-pair (STP) and fiber cables
- Contains onchip scrambler and PCS for CAT5 to significantly reduce cost of 100BASE-T solutions
- Supports full-duplex operation on both 10Mb/s and 10/100Mb/s ports
- Provides external and internal loopback capability on both ports
- Contains a variety of flexible address filtering modes (including perfect, hash tables, inverse perfect, and promiscuous):
  - 16 perfect addresses (normal or inverse filtering)
  - 512 hash-filtered addresses
  - 512 hash-filtered multicast addresses and one perfect address
  - Pass all multicast
  - Promiscuous
  - Pass all incoming packets with a status report
- Offers a unique, patented solution to Ethernet capture-effect problem
- Contains large independent receive and transmit FIFOs; no additional onboard memory required
- Includes a powerful onchip direct memory access (DMA) with programmable burst size providing for low CPU utilization
- Implements unique, patent-pending intelligent arbitration between DMA channels preventing underflow or overflow
- Supports PCI clock frequency from dc to 33 MHz; network operational with PCI clock from 20 MHz to 33 MHz
- Supports an unlimited PCI burst
- Supports PCI read multiple command

1 Indicates 21140A features that are not available on the 21140.
1.1 General Description

- Supports early interrupts on transmit
- Implements low-power management with two power-saving modes (sleep or snooze)
- Supports both PCI 5.0-V and 3.3-V signaling environments
- Supports either big or little endian byte ordering for buffers and descriptors
- Contains 8-bit, general-purpose, programmable register and corresponding I/O pins
- Provides LED support for various network activity indications
- Provides MicroWire interface for serial ROM (1K and 4K EEPROM)
- Provides an upgradable boot ROM (flash or EEPROM) interface of up to 256KB
- Supports automatic loading of subvendor ID configuration register
- Implements JTAG-compatible test-access port with boundary-scan pins
- Supports IEEE 802.3, ANSI 8802-3, and Ethernet standards
- Implements low-power, 3.3-V complementary metal-oxide semiconductor (CMOS) process technology

1.2 Hardware Overview

The following list describes the 21140A hardware components, and Figure 1–1 shows a block diagram of the 21140A:

- PCI interface—Includes all interface functions to the PCI bus; handles all interconnect control signals; and executes PCI DMA and I/O transactions.
- DMA—Contains dual receive and transmit controller; supports bursts of up to 32 longwords; handles data transfers between CPU memory and onchip memory.
- FIFOs—Contains two FIFOs for receive and transmit; supports automatic packet deletion on receive (runt packets or after a collision) and packet retransmission after a collision on transmit.
- TxM—Handles all CSMA/CD media access control transmit operations, and transfers data from transmit FIFO to the ENDEC for transmission.

1 Indicates 21140A features that are not available on the 21140.
2 Carrier-sense multiple access with collision detection
3 Media access control
1.2 Hardware Overview

- RxM—Handles all CSMA/CD receive operations, and transfers the data from the ENDEC to the receive FIFO.
- Physical coding sublayer—Implements the encoding and decoding sublayer of the 100BASE-TX (CAT5) specification, including the squelch.
- Scrambler/descrambler—Implements the twisted-pair physical layer medium dependent (TP-PMD) scrambler/descrambler scheme.
- General-purpose register—Enables software to use for input or output functions.
- Serial interface—Provides a 7-wire conventional interface to the Ethernet ENDEC components.
- MII/SYM interface—Provides a full MII signal interface and a direct interface to the 10/100Mb/s ENDEC for CAT5.
- Serial ROM—Provides a direct interface to the MicroWire Ethernet address ROM and system parameters.
- Boot ROM—Includes the required interface to the boot ROM for read and write operations; supports accesses to bytes or longwords (32-bit). Also provides the ability to connect an external 8-bit register to the boot ROM port.
1.2 Hardware Overview

Figure 1–1 21140A Block Diagram
This chapter provides the 21140A pinout, and a functional description of each of the signals.

2.1 21140A Pinout

The 21140A is housed in the 144-pin PQFP. Figure 2–1 shows the 21140A pinout.
### 2.1 21140A Pinout

**Figure 2–1** 21140A Pinout Diagram (Top View)

![21140A Pinout Diagram](image)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>int_l</td>
</tr>
<tr>
<td>2</td>
<td>rst_l</td>
</tr>
<tr>
<td>3</td>
<td>vdd</td>
</tr>
<tr>
<td>4</td>
<td>vss</td>
</tr>
<tr>
<td>5</td>
<td>pci_clk</td>
</tr>
<tr>
<td>6</td>
<td>vdd</td>
</tr>
<tr>
<td>7</td>
<td>gnt_l</td>
</tr>
<tr>
<td>8</td>
<td>req_l</td>
</tr>
<tr>
<td>9</td>
<td>vss</td>
</tr>
<tr>
<td>10</td>
<td>ad&lt;31&gt;</td>
</tr>
<tr>
<td>11</td>
<td>ad&lt;30&gt;</td>
</tr>
<tr>
<td>12</td>
<td>ad&lt;29&gt;</td>
</tr>
<tr>
<td>13</td>
<td>ad&lt;28&gt;</td>
</tr>
<tr>
<td>14</td>
<td>vss</td>
</tr>
<tr>
<td>15</td>
<td>ad&lt;27&gt;</td>
</tr>
<tr>
<td>16</td>
<td>ad&lt;26&gt;</td>
</tr>
<tr>
<td>17</td>
<td>vdd</td>
</tr>
<tr>
<td>18</td>
<td>ad&lt;25&gt;</td>
</tr>
<tr>
<td>19</td>
<td>ad&lt;24&gt;</td>
</tr>
<tr>
<td>20</td>
<td>c_be_l&lt;3&gt;</td>
</tr>
<tr>
<td>21</td>
<td>idsel</td>
</tr>
<tr>
<td>22</td>
<td>vss</td>
</tr>
<tr>
<td>23</td>
<td>ad&lt;23&gt;</td>
</tr>
<tr>
<td>24</td>
<td>ad&lt;22&gt;</td>
</tr>
<tr>
<td>25</td>
<td>ad&lt;21&gt;</td>
</tr>
<tr>
<td>26</td>
<td>ad&lt;20&gt;</td>
</tr>
<tr>
<td>27</td>
<td>vdd</td>
</tr>
<tr>
<td>28</td>
<td>ad&lt;19&gt;</td>
</tr>
<tr>
<td>29</td>
<td>ad&lt;18&gt;</td>
</tr>
<tr>
<td>30</td>
<td>vdd</td>
</tr>
<tr>
<td>31</td>
<td>vss</td>
</tr>
<tr>
<td>32</td>
<td>vss</td>
</tr>
<tr>
<td>33</td>
<td>vss</td>
</tr>
<tr>
<td>34</td>
<td>ad&lt;17&gt;</td>
</tr>
<tr>
<td>35</td>
<td>ad&lt;16&gt;</td>
</tr>
<tr>
<td>36</td>
<td>vss</td>
</tr>
</tbody>
</table>

*Note: Pinout details include various voltage levels (vss, vdd), control signals (int_l, rst_l), and address lines (ad<31>). The diagram also indicates various symmetrical and non-symmetrical pins (mii/sym_tclk, mii/sym_rclk, etc.).*
2.2 Signal Descriptions

Table 2–1 provides a functional description of each of the 21140A signals.

The following terms describe the 21140A pinout:

- **Address phase**
  Address and appropriate bus commands are driven during this cycle.

- **Data phase**
  Data and the appropriate byte enable codes are driven during this cycle.

- **_l**
  All pin names with the _l suffix are asserted low.

The following abbreviations are used in Table 2–1.

- I = Input
- O = Output
- I/O = Input/output
- O/D = Open drain
- P = Power

Table 2–1 Functional Description of 21140A Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ad&lt;31:0&gt;</td>
<td>I/O</td>
<td>See Figure 2–1.</td>
<td>32-bit PCI address and data lines. Address and data bits are multiplexed on the same pins. During the first clock cycle of a transaction, the address bits contain a physical address (32 bits). During subsequent clock cycles, these same lines contain data (32 bits). A 21140A bus transaction consists of an address phase followed by one or more data phases. The 21140A supports both read and write bursts (in master operation only). Little and big endian byte ordering can be used.</td>
</tr>
<tr>
<td>br_a&lt;0&gt;</td>
<td>O</td>
<td>102</td>
<td>Boot ROM address line bit 0. In a 256KB configuration, this pin also carries in two consecutive address cycles, boot ROM address bits 16 and 17.</td>
</tr>
</tbody>
</table>

(continued on next page)
## 2.2 Signal Descriptions

### Table 2–1 (Cont.) Functional Description of 21140A Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>br_a&lt;1&gt;</td>
<td>O</td>
<td>103</td>
<td>Boot ROM address line bit 1. This pin also latches the boot ROM address and control lines by the two external latches.</td>
</tr>
<tr>
<td>br_ad&lt;7:0&gt;</td>
<td>I/O</td>
<td>See Figure 2–1.</td>
<td>Boot ROM address and data multiplexed lines bits 7 through 0. In the first of two consecutive address cycles, these lines contain the boot ROM address bits 7 through 2, oe_l and we_l, followed by boot ROM address bits 15 through 8 in the second cycle. During the data cycle, bits 7 through 0 contain data. During operation with the external register, these lines are used to carry data bits 7 through 0 to and from the external register.</td>
</tr>
<tr>
<td>br_ce_l</td>
<td>O</td>
<td>101</td>
<td>Boot ROM or external register chip enable.</td>
</tr>
<tr>
<td>c_be_l&lt;3:0&gt;</td>
<td>I/O</td>
<td>See Figure 2–1.</td>
<td>Bits 0 through 3 of the bus command and byte enable lines. Bus command and byte enable are multiplexed on the same PCI pins. During the address phase of the transaction, these 4 bits provide the bus command. During the data phase, these 4 bits provide the byte enable. The byte enable determines which byte lines carry valid data. For example, bit 0 applies to byte 0, and bit 3 applies to byte 3.</td>
</tr>
<tr>
<td>devsel_l</td>
<td>I/O</td>
<td>42</td>
<td>Device select is asserted by the target of the current bus access. When the 21140A is the initiator of the current bus access, it expects the target to assert devsel_l within 5 bus cycles, confirming the access. If the target does not assert devsel_l within the required bus cycles, the 21140A aborts the cycle. To meet the timing requirements, the 21140A asserts this signal in a medium speed (within 2 bus cycles).</td>
</tr>
</tbody>
</table>

(continued on next page)
Table 2–1 (Cont.) Functional Description of 21140A Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frame_l</td>
<td>I/O</td>
<td>39</td>
<td>The signal frame_l is driven by the 21140A (bus master) to indicate the beginning and duration of an access. Signal frame_l asserts to indicate the beginning of a bus transaction. While frame_l is asserted, data transfers continue. Signal frame_l deasserts to indicate that the next data phase is the final data phase transaction.</td>
</tr>
<tr>
<td>gep&lt;7:0&gt;</td>
<td>I/O</td>
<td>See Figure 2–1.</td>
<td>General-purpose pins can be used by software as either status pins or control pins. These pins can be configured by software to perform either input or output functions.</td>
</tr>
<tr>
<td>gnt_l</td>
<td>I</td>
<td>7</td>
<td>Bus grant asserts to indicate to the 21140A that access to the bus is granted.</td>
</tr>
<tr>
<td>idsel</td>
<td>I</td>
<td>22</td>
<td>Initialization device select asserts to indicate that the host is issuing a configuration cycle to the 21140A.</td>
</tr>
<tr>
<td>int_l</td>
<td>O/D</td>
<td>1</td>
<td>Interrupt request asserts when one of the appropriate bits of CSR5 sets and causes an interrupt, provided that the corresponding mask bit in CSR7 is not asserted. Interrupt request deasserts by writing a 1 into the appropriate CSR5 bit. If more than one interrupt bit is asserted in CSR5 and the host does not clear all input bits, the 21140A deasserts int_l for one cycle to support edge-triggered systems. This pin must be pulled up by an external resistor.</td>
</tr>
</tbody>
</table>

(continued on next page)
2.2 Signal Descriptions

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>irdy_l</td>
<td>I/O</td>
<td>40</td>
<td>Initiator ready indicates the bus master’s ability to complete the current data phase of the transaction. A data phase is completed on any rising edge of the clock when both irdy_l and target ready trdy_l are asserted. Wait cycles are inserted until both irdy_l and trdy_l are asserted together. When the 21140A is the bus master, irdy_l is asserted during write operations to indicate that valid data is present on the 32-bit ad lines. During read operations, the 21140A asserts irdy_l to indicate that it is ready to accept data.</td>
</tr>
<tr>
<td>mii_clsn</td>
<td>I</td>
<td>112</td>
<td>Collision detected is asserted when detected by an external physical layer protocol (PHY) device.</td>
</tr>
<tr>
<td>mii_crs</td>
<td>I</td>
<td>113</td>
<td>Carrier sense is asserted by the PHY when the media is active.</td>
</tr>
<tr>
<td>mii_dv</td>
<td>I</td>
<td>111</td>
<td>Data valid is asserted by an external PHY when receive data is present on the mii/sym_rxd lines and is deasserted at the end of the packet. This signal should be synchronized with the mii/sym_rclk signal.</td>
</tr>
<tr>
<td>mii_err</td>
<td>I</td>
<td>110</td>
<td>Receive error asserts when a data decoding error is detected by an external PHY device. This signal is synchronized to mii/sym_rclk and can be asserted for a minimum of one receive clock. When asserted during a packet reception, it sets the cyclic redundancy check (CRC) error bit in the receive descriptor (RDES0).</td>
</tr>
<tr>
<td>mii_mdc</td>
<td>O</td>
<td>106</td>
<td>MII management data clock is sourced by the 21140A to the PHY devices as a timing reference for the transfer of information on the mii_mdio signal.</td>
</tr>
</tbody>
</table>

(continued on next page)
### Table 2–1 (Cont.) Functional Description of 21140A Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mii_mdio</td>
<td>I/O</td>
<td>105</td>
<td>MII management data input/output transfers control information and status between the PHY and the 21140A.</td>
</tr>
<tr>
<td>mii/srl</td>
<td>O</td>
<td>133</td>
<td>Indicates the selected port: SRL or MII/SYM. When asserted, the MII/SYM port is active. When deasserted, the SRL port is active.</td>
</tr>
<tr>
<td>mii/sym_rclk</td>
<td>I</td>
<td>114</td>
<td>Supports either the 25-MHz or 2.5-MHz receive clock. This clock is recovered by the PHY.</td>
</tr>
<tr>
<td>mii/sym_rxd&lt;3:0&gt;</td>
<td>I</td>
<td>See Figure 2–1.</td>
<td>Four parallel receive data lines when MII mode is selected. This data is driven by an external PHY that attached the media and should be synchronized with the mii/sym_rclk signal.</td>
</tr>
<tr>
<td>mii/sym_tclk</td>
<td>I</td>
<td>123</td>
<td>Supports the 25-MHz or 2.5-MHz transmit clock supplied by the external physical layer medium dependent (PMD) device. This clock should always be active.</td>
</tr>
<tr>
<td>mii/sym_txd&lt;3:0&gt;</td>
<td>O</td>
<td>See Figure 2–1.</td>
<td>Four parallel transmit data lines. This data is synchronized to the assertion of the mii/sym_tclk signal and is latched by the external PHY on the rising edge of the mii/sym_tclk signal.</td>
</tr>
<tr>
<td>mii_txen</td>
<td>O</td>
<td>125</td>
<td>Transmit enable signals that the transmit is active to an external PHY device. In PCS mode (CSR6&lt;23&gt;), this signal reflects the transmit activity of the MAC sublayer.</td>
</tr>
<tr>
<td>nc</td>
<td>O</td>
<td>104</td>
<td>No connection.</td>
</tr>
<tr>
<td>par</td>
<td>I/O</td>
<td>47</td>
<td>Parity is calculated by the 21140A as an even parity bit for the 32-bit ad and 4-bit c_be_l lines.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>During address and data phases, parity is calculated on all the ad and c_be_l lines whether or not any of these lines carry meaningful information.</td>
</tr>
</tbody>
</table>

(continued on next page)
2.2 Signal Descriptions

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pci_clk</td>
<td>I</td>
<td>5</td>
<td>The clock provides the timing for the 21140A related PCI bus transactions. All the bus signals are sampled on the rising edge of pci_clk. The clock frequency range is between 25 MHz and 33 MHz.</td>
</tr>
<tr>
<td>perr_l</td>
<td>I/O</td>
<td>45</td>
<td>Parity error asserts when a data parity error is detected. When the 21140A is the bus master and a parity error is detected, the 21140A asserts both CSR5 bit 13 (system error) and CFCS bit 8 (serr_l enable). Next, it completes the current data burst transaction, and then stops operation. After the host clears the system error, the 21140A continues its operation. When the 21140A is the bus target and a parity error is detected, the 21140A asserts perr_l. This pin must be pulled up by an external resistor.</td>
</tr>
<tr>
<td>rcv_match</td>
<td>O</td>
<td>122</td>
<td>Receive match indication is asserted when a received packet has passed address recognition.</td>
</tr>
<tr>
<td>req_l</td>
<td>O</td>
<td>8</td>
<td>Bus request is asserted by the 21140A to indicate to the bus arbiter that it wants to use the bus.</td>
</tr>
<tr>
<td>rst_l</td>
<td>I</td>
<td>2</td>
<td>Resets the 21140A to its initial state. This signal must be asserted for at least 10 active PCI clock cycles. When in the reset state, all PCI output pins are put into tristate and all PCI open drain (O/D) signals are floated.</td>
</tr>
<tr>
<td>sd</td>
<td>I</td>
<td>109</td>
<td>Signal detect indication supplied by an external physical layer medium dependent (PMD) device.</td>
</tr>
</tbody>
</table>

(continued on next page)
2.2 Signal Descriptions

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serr_l</td>
<td>I/O</td>
<td>46</td>
<td>If an address parity error is detected while CFCS bit 31 (detected parity error) is enabled, the 21140A asserts both serr_l (system error) and CFCS bit 30 (signal system error). When an address parity error is detected, system error asserts two clock cycles after the failing address. This pin must be pulled up by an external resistor.</td>
</tr>
<tr>
<td>sr_ck</td>
<td>O</td>
<td>78</td>
<td>Serial ROM clock signal.</td>
</tr>
<tr>
<td>sr_cs</td>
<td>O</td>
<td>79</td>
<td>Serial ROM chip-select signal.</td>
</tr>
<tr>
<td>sr_di</td>
<td>O</td>
<td>77</td>
<td>Serial ROM data-in signal.</td>
</tr>
<tr>
<td>sr_do</td>
<td>I</td>
<td>76</td>
<td>Serial ROM data-out signal.</td>
</tr>
<tr>
<td>srl_clsn</td>
<td>I</td>
<td>134</td>
<td>Collision detect signals a collision occurrence on the Ethernet cable to the 21140A. It may be asserted and deasserted asynchronously by the external ENDEC to the receive clock.</td>
</tr>
<tr>
<td>srl_rclk</td>
<td>I</td>
<td>137</td>
<td>Receive clock carries the recovered receive clock supplied by an external ENDEC. During idle periods, srl_rclk may be inactive.</td>
</tr>
<tr>
<td>srl_rxd</td>
<td>I</td>
<td>135</td>
<td>Receive data carries the input receive data from the external ENDEC. The incoming data should be synchronous with the srl_rclk signal.</td>
</tr>
<tr>
<td>srl_rxen</td>
<td>I</td>
<td>136</td>
<td>Receive enable signals activity on the Ethernet cable to the 21140A. It is asserted when receive data is present on the Ethernet cable and deasserted at the end of a frame. It may be asserted and deasserted asynchronously to the receive clock (srl_rclk) by the external ENDEC.</td>
</tr>
<tr>
<td>srl_tclk</td>
<td>I</td>
<td>139</td>
<td>Transmit clock carries the transmit clock supplied by an external ENDEC. This clock must always be active (even during reset).</td>
</tr>
</tbody>
</table>

(continued on next page)
2.2 Signal Descriptions

Table 2–1 (Cont.) Functional Description of 21140A Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>srl_txd</td>
<td>O</td>
<td>138</td>
<td>Transmit data carries the serial output data from the 21140A. This data is synchronized to the srl_tclk signal.</td>
</tr>
<tr>
<td>srl_txen</td>
<td>O</td>
<td>140</td>
<td>Transmit enable signals an external ENDEC that the 21140A transmit is in progress.</td>
</tr>
<tr>
<td>stop_l</td>
<td>I/O</td>
<td>43</td>
<td>Stop indicator indicates that the current target is requesting the bus master to stop the current transaction. The 21140A responds to the assertion of stop_l when it is the bus master, either to disconnect, retry, or abort.</td>
</tr>
<tr>
<td>sym_link</td>
<td>O</td>
<td>124</td>
<td>Indicates that the descrambler is locked to the input data signal.</td>
</tr>
<tr>
<td>sym_rxd&lt;4&gt;</td>
<td>I</td>
<td>119</td>
<td>Receive data together with the four receive lines mii/sym_rxd&lt;3:0&gt; provide five parallel lines of data in symbol form for use in PCS mode (100BASE-T, CSR6&lt;23&gt;). This data is driven by an external PMD device and should be synchronized to the mii/sym_rclk signal.</td>
</tr>
<tr>
<td>sym_txd&lt;4&gt;</td>
<td>O</td>
<td>132</td>
<td>Transmit data together with the four transmit lines mii/sym_txd&lt;3:0&gt; provide five parallel lines of data in symbol form for use in PCS mode (100BASE-T, CSR6&lt;23&gt;). This data is synchronized on the rising of the sym_tclk signal.</td>
</tr>
<tr>
<td>tck</td>
<td>I</td>
<td>141</td>
<td>J TAG clock shifts state information and test data into and out of the 21140A during J TAG test operations. This pin should not be left unconnected.</td>
</tr>
<tr>
<td>tdi</td>
<td>I</td>
<td>143</td>
<td>J TAG data-in pin is used to serially shift test data and instructions into the 21140A during J TAG test operations. This pin should not be left unconnected.</td>
</tr>
<tr>
<td>tdo</td>
<td>O</td>
<td>144</td>
<td>J TAG data-out pin is used to serially shift test data and instructions out of the 21140A during J TAG test operations.</td>
</tr>
</tbody>
</table>

(continued on next page)
2.2 Signal Descriptions

Table 2–1 (Cont.) Functional Description of 21140A Signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tms</td>
<td>I</td>
<td>142</td>
<td>JTAG test mode select controls the state operation of JTAG testing in the 21140A. This pin should not be left unconnected.</td>
</tr>
<tr>
<td>trdy_l</td>
<td>I/O</td>
<td>41</td>
<td>Target ready indicates the target agent's ability to complete the current data phase of the transaction. A data phase is completed on any clock when both trdy_l and irdy_l are asserted. Wait cycles are inserted until both irdy_l and trdy_l are asserted together. When the 21140A is the bus master, target ready is asserted by the bus slave on the read operation, indicating that valid data is present on the ad lines. During a write cycle, it indicates that the target is prepared to accept data.</td>
</tr>
<tr>
<td>vdd</td>
<td>P</td>
<td>See Figure 2–1.</td>
<td>A 3.3-V supply input voltage.</td>
</tr>
<tr>
<td>vdd_clamp</td>
<td>P</td>
<td>73</td>
<td>A 5-V reference for a 5-V signaling environment and a 3.3-V reference for a 3.3-V signaling environment.</td>
</tr>
<tr>
<td>vss</td>
<td>P</td>
<td>See Figure 2–1.</td>
<td>Ground pins.</td>
</tr>
</tbody>
</table>
This chapter describes the configuration registers, and the command and status registers (CSRs) of the 21140A. The 21140A uses 10 configuration registers for initialization and configuration, and 16 CSRs (CSR0 through CSR15) for host communication. Configuration registers are used to identify and query the 21140A. The CSRs, which are mapped in the host I/O or memory address space, are used for initialization, pointers, commands, and status reporting.

3.1 Configuration Operation

The 21140A enables a full software-driven initialization and configuration. This permits the software to identify and query the 21140A.

The 21140A treats configuration space write operations to registers that are reserved as no-ops. That is, the access completes normally on the bus and the data is discarded. Read accesses, to reserved or unimplemented registers, complete normally and a data value of 0 is returned.

Software reset (CSR0<0>) has no effect on the configuration registers. Hardware reset sets the configuration registers to their default values.

The 21140A supports byte, word, and longword accesses to configuration registers.

3.1.1 Configuration Register Mapping

Table 3-1 lists the definitions and addresses for the configuration registers.
3.1 Configuration Operation

Table 3–1 Configuration Registers Mapping

<table>
<thead>
<tr>
<th>Configuration Register</th>
<th>Identifier</th>
<th>I/O Address Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>CFID</td>
<td>00H</td>
</tr>
<tr>
<td>Command and status</td>
<td>CFCS</td>
<td>04H</td>
</tr>
<tr>
<td>Revision</td>
<td>CFRV</td>
<td>08H</td>
</tr>
<tr>
<td>Latency timer</td>
<td>CFLT</td>
<td>0CH</td>
</tr>
<tr>
<td>Base I/O address</td>
<td>CBIO</td>
<td>10H</td>
</tr>
<tr>
<td>Base memory address</td>
<td>CBMA</td>
<td>14H</td>
</tr>
<tr>
<td>Reserved</td>
<td>—</td>
<td>18H–28H</td>
</tr>
<tr>
<td>Subsystem ID</td>
<td>SSID</td>
<td>2CH</td>
</tr>
<tr>
<td>Expansion ROM base address</td>
<td>CBER</td>
<td>30H</td>
</tr>
<tr>
<td>Reserved</td>
<td>—</td>
<td>34H–38H</td>
</tr>
<tr>
<td>Interrupt</td>
<td>CFIT</td>
<td>3CH</td>
</tr>
<tr>
<td>Device and Driver area</td>
<td>CFDD</td>
<td>40H</td>
</tr>
</tbody>
</table>

3.1.2 Configuration Registers

The 21140A implements 10 configuration registers. These registers are described in the following subsections.

3.1.2.1 Configuration ID Register (CFID–Offset 00H)

The CFID register identifies the 21140A. Figure 3–1 shows the CFID register bit fields and Table 3–2 describes the bit fields.

Figure 3–1 CFID Configuration ID Register
3.1 Configuration Operation

Table 3–2  CFID Configuration ID Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:16</td>
<td>Device ID</td>
</tr>
<tr>
<td></td>
<td>Provides the unique 21140A ID number (0009H).</td>
</tr>
<tr>
<td>15:0</td>
<td>Vendor ID</td>
</tr>
<tr>
<td></td>
<td>Specifies the manufacturer of the 21140A (1011H).</td>
</tr>
</tbody>
</table>

Table 3–3 lists the access rules for the CFID register.

Table 3–3  CFID Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after hardware reset</td>
<td>00091011H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>Writing has no effect.</td>
</tr>
</tbody>
</table>

3.1.2.2 Command and Status Configuration Register (CFCS–Offset 04H)

The CFCS register is divided into two sections: a command register (CFCS<15:0>) and a status register (CFCS<31:16>).

The command register provides control of the 21140A's ability to generate and respond to PCI cycles. Writing 0 to this register, the 21140A logically disconnects from the PCI bus for all accesses except configuration accesses.

The status register records status information for the PCI bus-related events. The CFCS status bits do not clear when read. Writing 1 to these bits clears them; writing 0 has no effect.

Figure 3–2 shows the CFCS bit fields and Table 3–4 describes the bit fields.
3.1 Configuration Operation

Figure 3–2 CFCS Command and Status Configuration Register

Table 3–4 CFCS Command and Status Configuration Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Register Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| 31    | Status        | Detected Parity Error  
When set, indicates that the 21140A detected a parity error, even if parity error handling is disabled in parity error response (CFCS<6>). |
| 30    | Status        | Signal System Error  
When set, indicates that the 21140A asserted the system error (serr_i) pin. |
| 29    | Status        | Received Master Abort  
When set, indicates that the 21140A terminated a master transaction with master abort. |
| 28    | Status        | Received Target Abort  
When set, indicates that the 21140A master transaction was terminated due to a target abort. |

(continued on next page)
### Table 3–4 (Cont.) CFCS Command and Status Configuration Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Register Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26:25</td>
<td>Status</td>
<td><strong>Device Select Timing</strong>&lt;br&gt;Indicates the timing of the assertion of device select (devsel_l). These bits are fixed at 01, which indicates a medium assertion of devsel_l.</td>
</tr>
<tr>
<td>24</td>
<td>Status</td>
<td><strong>Data Parity Report</strong>&lt;br&gt;This bit sets when the following conditions are met:&lt;br&gt;- The 21140A asserts parity error (perr_l) or it senses the assertion of perr_l by another device.&lt;br&gt;- The 21140A operates as a bus master for the operation that caused the error.&lt;br&gt;- Parity error response (CFCS&lt;6&gt;) is set.</td>
</tr>
<tr>
<td>23</td>
<td>Status</td>
<td><strong>Fast Back-to-Back</strong>&lt;br&gt;Always set by the 21140A. This indicates that the 21140A is capable of accepting fast back-to-back transactions that are not sent to the same bus device.</td>
</tr>
<tr>
<td>8</td>
<td>Command</td>
<td><strong>System Error Enable</strong>&lt;br&gt;When set, the 21140A asserts system error (serr_l) when it detects a parity error on the address phase (ad&lt;31:0&gt; and c_be_l&lt;3:0&gt;).</td>
</tr>
<tr>
<td>6</td>
<td>Command</td>
<td><strong>Parity Error Response</strong>&lt;br&gt;When set, the 21140A asserts system error (CSR5&lt;13&gt;) after a parity error detection.&lt;br&gt;When reset, any detected parity error is ignored and the 21140A continues normal operation.&lt;br&gt;Parity checking is disabled after reset.</td>
</tr>
<tr>
<td>2</td>
<td>Command</td>
<td><strong>Master Operation</strong>&lt;br&gt;When set, the 21140A is capable of acting as a bus master.&lt;br&gt;When reset, the 21140A capability to generate PCI accesses is disabled.&lt;br&gt;For normal 21140A operation, this bit must be set.</td>
</tr>
</tbody>
</table>

(continued on next page)
3.1 Configuration Operation

Table 3–4 (Cont.) CFCS Command and Status Configuration Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Register Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1     | Command       | Memory Space Access  
When set, the 21140A responds to memory space accesses. 
When reset, the 21140A does not respond to memory space accesses. |
| 0     | Command       | I/O Space Access  
When set, the 21140A responds to I/O space accesses. 
When reset, the 21140A does not respond to I/O space accesses. |

Table 3–5 lists the access rules for the CFCS register.

Table 3–5 CFCS Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after hardware reset</td>
<td>02800000H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.1.2.3 Configuration Revision Register (CFRV–Offset 08H)

The CFRV register contains the 21140A revision number. Figure 3–3 shows the CFRV bit fields and Table 3–6 describes the bit fields.

Figure 3–3 CFRV Configuration Revision Register
3.1 Configuration Operation

Table 3–6 CFRV Configuration Revision Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td><strong>Base Class</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates the network controller and is equal to 2H.</td>
</tr>
<tr>
<td>23:16</td>
<td><strong>Subclass</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates the fast Ethernet controller and is equal to 0H.</td>
</tr>
<tr>
<td>7:4</td>
<td><strong>Revision Number</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates the 21140A revision number and is equal to 2H. This number is incremented for subsequent 21140A revisions.</td>
</tr>
<tr>
<td>3:0</td>
<td><strong>Step Number</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates the 21140A step number and is equal to 0H. This number is incremented for subsequent 21140A steps within the current revision.</td>
</tr>
</tbody>
</table>

Table 3–7 lists the access rules for the CFRV register.

Table 3–7 CFRV Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after hardware reset</td>
<td>02000020H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>Writing has no effect.</td>
</tr>
</tbody>
</table>

3.1.2.4 Configuration Latency Timer Register (CFLT–Offset 0CH)

This register configures the 21140A bus latency timer. Figure 3–4 shows the CFLT bit field and Table 3–8 describes the bit field.

Figure 3–4 CFLT Configuration Latency Timer Register

![Configuration Latency Timer](LJ-04697.A15)
### 3.1 Configuration Operation

**Table 3–8 CFLT Configuration Latency Timer Register Description**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:8</td>
<td><strong>Configuration Latency Timer</strong></td>
</tr>
<tr>
<td></td>
<td>Specifies, in units of PCI bus clocks, the value of the latency timer of the 21140A. When the 21140A asserts \texttt{frame_l}, it enables its latency timer to count. If the 21140A deasserts \texttt{frame_l} prior to count expiration, the content of the latency timer is ignored. Otherwise, after the count expires, the 21140A initiates transaction termination as soon as its \texttt{gnt_l} is deasserted.</td>
</tr>
</tbody>
</table>

Table 3–9 lists the access rules for the CFLT register.

**Table 3–9 CFLT Access Rules**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after hardware reset</td>
<td>0H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

#### 3.1.2.5 Configuration Base I/O Address Register (CBIO–Offset 10H)

The CBIO register specifies the base I/O address for accessing the 21140A CSRs (CSR0–15). For example, if the CBIO register is programmed to 1000H, the I/O address of CSR15 is equal to CBIO + CSR15-offset for a value of 1078H (Table 3–22).

This register must be initialized prior to accessing any CSR with I/O access.

Figure 3–5 shows the CBIO bit fields and Table 3–10 describes the bit fields.
3.1 Configuration Operation

Figure 3–5  CBIO Configuration Base I/O Address Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:7</td>
<td>Configuration Base I/O Address</td>
</tr>
<tr>
<td></td>
<td>Defines the address assignment mapping of 21140A CSRs.</td>
</tr>
<tr>
<td>6:1</td>
<td>This field value is 0 when read.</td>
</tr>
<tr>
<td>0</td>
<td>Memory I/O Space Indicator</td>
</tr>
<tr>
<td></td>
<td>Determines that the register maps into the I/O space. The value in this field is 1. This is a read-only field.</td>
</tr>
</tbody>
</table>

Table 3–11 lists the access rules for the CBIO register.

Table 3–11  CBIO Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after hardware reset</td>
<td>Undefined</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.1.2.6 Configuration Base Memory Address Register (CBMA–14H)

The CBMA register specifies the base memory address for memory accesses to the 21140A CSRs (CSR0–15).

This register must be initialized prior to accessing any CSR with memory access.
3.1 Configuration Operation

Figure 3–6 shows the CBMA bit fields and Table 3–12 describes the bit fields.

**Figure 3–6 CBMA Configuration Base Memory Address Register**

![Diagram of CBMA Configuration Base Memory Address Register]

**Table 3–12 CBMA Configuration Base Memory Address Register Description**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:7</td>
<td><strong>Configuration Base Memory Address</strong>&lt;br&gt;Defines the address assignment mapping of the 21140A CSRs.</td>
</tr>
<tr>
<td>6:1</td>
<td><strong>This field value is 0 when read.</strong></td>
</tr>
<tr>
<td>0</td>
<td><strong>Memory Space Indicator</strong>&lt;br&gt;Determines that the register maps into the memory space. The value in this field is 0. This is a read-only field.</td>
</tr>
</tbody>
</table>

Table 3–13 lists the access rules for the CBMA register.

**Table 3–13 CBMA Access Rules**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after hardware reset</td>
<td>Undefined</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.1.2.7 **Subsystem ID Register (SSID–2CH)**

The SSID register is a read-only 32-bit register. The content of the SSID is loaded from the serial ROM after a hardware reset. The value is 0 if the serial ROM data integrity check fails.
3.1 Configuration Operation

Figure 3–7 shows the SSID bit fields and Table 3–14 describes the bit fields.

Figure 3–7  SSID Subsystem ID Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:16</td>
<td>Subsystem ID</td>
</tr>
<tr>
<td>15:0</td>
<td>Subsystem Vendor ID</td>
</tr>
</tbody>
</table>

Indicates a 16-bit field containing the subsystem ID.

Indicates a 16-bit field containing the subsystem vendor ID.

Table 3–15 lists the access rules for the SSID register.

Table 3–15  SSID Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after hardware reset</td>
<td>Read from serial ROM.</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.1.2.8 Expansion ROM Base Address Register (CBER–Offset 30H)

The CBER register specifies the base address and provides information about the expansion ROM size. This register must be initialized prior to accessing the expansion ROM.

Figure 3–8 shows the CBER bit fields and Table 3–16 describes the bit fields.
3.1 Configuration Operation

Figure 3–8  CBER Expansion ROM Base Address Register

![Diagram of CBER Expansion ROM Base Address Register]

Table 3–16  CBER Expansion ROM Base Address Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:10</td>
<td><strong>Expansion ROM Base Address</strong></td>
</tr>
<tr>
<td></td>
<td>Defines the address assignment mapping of the expansion ROM. It also</td>
</tr>
<tr>
<td></td>
<td>provides information about the expansion ROM size. CBER&lt;17:10&gt; are</td>
</tr>
<tr>
<td></td>
<td>hardwired to 0, indicating that the expansion ROM size is up to 256KB.</td>
</tr>
<tr>
<td>9:1</td>
<td><strong>This field value is 0 when read.</strong></td>
</tr>
<tr>
<td>0</td>
<td><strong>Expansion ROM Enable Bit</strong></td>
</tr>
<tr>
<td></td>
<td>The 21140A responds to its expansion ROM accesses only if the memory</td>
</tr>
<tr>
<td></td>
<td>space access bit (CFCS&lt;1&gt;) and the expansion ROM enable bit are both</td>
</tr>
<tr>
<td></td>
<td>set to 1.</td>
</tr>
</tbody>
</table>

Table 3–17 lists the access rules for the CBER register.

Table 3–17  CBER Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>Software reset has no effect.</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.1.2.9 Configuration Interrupt Register (CFIT–Offset 3CH)

The CFIT register is divided into two sections: the interrupt line and the interrupt pin. CFIT configures both the system’s interrupt line and the 21140A interrupt pin connection.
3.1 Configuration Operation

Figure 3-9 shows the CFIT bit fields and Table 3-18 describes the bit fields.

**Figure 3–9  CFIT Configuration Interrupt Register**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td>MAX_LAT</td>
</tr>
<tr>
<td></td>
<td>This field indicates how often the device needs to gain access to the PCI bus. Time unit is equal to 0.25 μs.</td>
</tr>
<tr>
<td>23:16</td>
<td>MIN_GNT</td>
</tr>
<tr>
<td></td>
<td>This field indicates the burst period length the device needs, assuming a clock rate of 33 MHz. Time unit is equal to 0.25 μs.</td>
</tr>
<tr>
<td>15:8</td>
<td>Interrupt Pin</td>
</tr>
<tr>
<td></td>
<td>Indicates which interrupt pin the 21140A uses. The 21140A uses INTA# and the read value is 01H.</td>
</tr>
<tr>
<td>7:0</td>
<td>Interrupt Line</td>
</tr>
<tr>
<td></td>
<td>Provides interrupt line routing information. The basic input/output system (BIOS) writes the routing information into this field when it initializes and configures the system. The value in this field indicates which input of the system interrupt controller is connected to the 21140A's interrupt pin. The driver can use this information to determine priority and vector information. Values in this field are system architecture specific.</td>
</tr>
</tbody>
</table>

Table 3-19 lists the access rules for the CFIT register.
3.1 Configuration Operation

Table 3–19 CFIT Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after hardware reset</td>
<td>281401XXH</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.1.2.10 Configuration Device and Driver Area Register (CFDD–Offset 40H)

The CFDD register can be used to store driver-specific information during initialization.

Figure 3–10 shows the CFDD bit field and Table 3–20 describes the bit field.

Figure 3–10 CFDD Configuration Driver Area Register
3.1 Configuration Operation

Table 3–20   CFDD Configuration Driver Area Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>Sleep Mode</td>
</tr>
<tr>
<td></td>
<td>When set, the 21140A enters sleep mode and most of its clocks are disconnected. While in sleep mode, the 21140A can only be accessed through its configuration space. The 21140A exits from this mode upon hardware reset. When this bit is reset, a permanent exit from this mode is accomplished. Note that this bit should not be asserted together with bit 30 (snooze mode) in this register.</td>
</tr>
<tr>
<td>30</td>
<td>Snooze Mode</td>
</tr>
<tr>
<td></td>
<td>When this bit is set, the following conditions exist. The 21140A enters snooze mode and most of its clocks are disconnected. The 21140A temporarily exits from snooze mode to normal operation mode upon sensing network activity, transmission start, or when it is being accessed by the host. When the activity is completed, the 21140A reenters snooze mode. When this bit is reset, a permanent exit from this mode is accomplished. Note that this bit should not be asserted together with bit 31 (sleep mode) in this register.</td>
</tr>
<tr>
<td>15:8</td>
<td>Driver Special Use</td>
</tr>
<tr>
<td></td>
<td>Specifies read and write fields for the driver’s special use.</td>
</tr>
</tbody>
</table>

Table 3–21 lists the access rules for the CFDD register.

Table 3–21   CFDD Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>Software reset has no effect.</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.2 CSR Operation

The 21140A CSRs are located in the host I/O or memory address space. The CSRs are quadword aligned, 32 bits long, and must be accessed using longword instructions with quadword-aligned addresses only.
3.2 CSR Operation

Note

Please note the following:

- Reserved bits should be written with 0; failing to do this could cause incompatibility problems with a future release. Reserved bits are UNPREDICTABLE on read access.
- Retries on second data transactions occur in response to burst accesses.

CSR0 are physically located in the chip. The host uses a single instruction to access to a CSR.

3.2.1 Control and Status Register Mapping

Table 3–22 lists the definitions and addresses for the CSR registers.

Table 3–22 CSR Mapping

<table>
<thead>
<tr>
<th>Register</th>
<th>Meaning</th>
<th>Offset from CSR Base Address (CBIO and CBMA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSR0</td>
<td>Bus mode</td>
<td>00H</td>
</tr>
<tr>
<td>CSR1</td>
<td>Transmit poll demand</td>
<td>08H</td>
</tr>
<tr>
<td>CSR2</td>
<td>Receive poll demand</td>
<td>10H</td>
</tr>
<tr>
<td>CSR3</td>
<td>Receive list base address</td>
<td>18H</td>
</tr>
<tr>
<td>CSR4</td>
<td>Transmit list base address</td>
<td>20H</td>
</tr>
<tr>
<td>CSR5</td>
<td>Status</td>
<td>28H</td>
</tr>
<tr>
<td>CSR6</td>
<td>Operation mode</td>
<td>30H</td>
</tr>
<tr>
<td>CSR7</td>
<td>Interrupt enable</td>
<td>38H</td>
</tr>
<tr>
<td>CSR8</td>
<td>Missed frames and overflow counter</td>
<td>40H</td>
</tr>
<tr>
<td>CSR9</td>
<td>Boot ROM, serial ROM, and MII management</td>
<td>48H</td>
</tr>
<tr>
<td>CSR10</td>
<td>Boot ROM programming address</td>
<td>50H</td>
</tr>
<tr>
<td>CSR11</td>
<td>General-purpose timer</td>
<td>58H</td>
</tr>
<tr>
<td>CSR12</td>
<td>General-purpose port</td>
<td>60H</td>
</tr>
<tr>
<td>CSR13</td>
<td>Reserved</td>
<td>68H</td>
</tr>
<tr>
<td>CSR14</td>
<td>Reserved</td>
<td>70H</td>
</tr>
<tr>
<td>CSR15</td>
<td>Watchdog timer</td>
<td>78H</td>
</tr>
</tbody>
</table>
3.2 CSR Operation

3.2.2 Host CSRs

The 21140A implements 16 CSRs (CSR0 through CSR15), which can be accessed by the host. Two of these registers (CSR13 and CSR14) are reserved.

3.2.2.1 Bus Mode Register (CSR0–Offset 00H)

Figure 3–11 shows the CSR0 bit fields and Table 3–23 describes the bit fields. CSR0 establishes the bus operating modes.

Figure 3–11 CSR0 Bus Mode Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>RML - Read Multiple</td>
</tr>
<tr>
<td>30-22</td>
<td>DBO - Descriptor Byte Ordering Mode</td>
</tr>
<tr>
<td>21-19</td>
<td>TAP - Transmit Automatic Polling</td>
</tr>
<tr>
<td>18</td>
<td>MBZ - Must Be Zero</td>
</tr>
<tr>
<td>17</td>
<td>CAL - Cache Alignment</td>
</tr>
<tr>
<td>16</td>
<td>PBL - Programmable Burst Length</td>
</tr>
<tr>
<td>15</td>
<td>BLE - Big/Little Endian</td>
</tr>
<tr>
<td>14-11</td>
<td>DSL - Descriptor Skip Length</td>
</tr>
<tr>
<td>10</td>
<td>BAR - Bus Arbitration</td>
</tr>
<tr>
<td>9</td>
<td>SWR - Software Reset</td>
</tr>
</tbody>
</table>

Preliminary—Subject to Change—January 1996
3.2 CSR Operation

Table 3–23  CSR0 Bus Mode Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 21    | **RML — Read Multiple**  
When set, the 21140A supports the memory-read-multiple command on the PCI bus. This bus command is used in memory read bursts with more than one longword. When reset, the 21140A uses memory-read command in all its memory read accesses on the PCI bus. |
| 20    | **DBO — Descriptor Byte Ordering Mode**  
When set, the 21140A operates in big endian ordering mode for descriptors only. When reset, the 21140A operates in little endian mode. |
| 19:17 | **TAP — Transmit Automatic Polling**  
When set and the 21140A is in a suspended state because a transmit buffer is unavailable, the 21140A performs a transmit automatic poll demand (Table 3–24). This field is not active in snooze mode. |
| 16    | **MBZ — Must Be Zero**  
This bit is not used by the 21140A and should always be programmed to zero. |
| 15:14 | **CAL — Cache Alignment**  
Programmable address boundaries for data burst stop (Table 3–26). If the buffer is not aligned, the 21140A executes the first transfer up to the address boundary. Then, all transfers are aligned to the specified boundary. These bits must be initialized after reset. |
| 13:8  | **PBL — Programmable Burst Length**  
Indicates the maximum number of longwords to be transferred in one DMA transaction. If reset, the 21140A burst is limited only by the amount of data stored in the receive FIFO (at least 16 longwords), or by the amount of free space in the transmit FIFO (at least 16 longwords) before issuing a bus request. The PBL can be programmed with permissible values 0, 1, 2, 4, 8, 16, or 32. After reset, the PBL default value is 0. |
| 7     | **BLE — Big/Little Endian**  
When set, the 21140A operates in big endian byte ordering mode. When reset, the 21140A operates in little endian byte ordering mode.  
Big endian is applicable only for data buffers.  
For example, the byte order in little endian of a data buffer is 12345678H, with each digit representing a nibble. In big endian, the byte orientation is 78563412H. |
| 6:2   | **DSL — Descriptor Skip Length**  
Specifies the number of longwords to skip between two descriptors. |

(continued on next page)
3.2 CSR Operation

Table 3–23 (Cont.) CSR0 Bus Mode Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>BAR — Bus Arbitration</strong></td>
</tr>
<tr>
<td></td>
<td>Selects the internal bus arbitration between the receive and transmit processes. When set, a round-robin arbitration scheme is applied resulting in equal sharing between processes. When reset, the receive process has priority over the transmit process, unless the 21140A is currently transmitting (Section 4.3.2).</td>
</tr>
<tr>
<td>0</td>
<td><strong>SWR — Software Reset</strong></td>
</tr>
<tr>
<td></td>
<td>When set, the 21140A resets all internal hardware with the exception of the configuration area; it does not change the port select setting (CSR6&lt;18&gt;).</td>
</tr>
</tbody>
</table>

Table 3–24 defines the transmit automatic polling bits.

Table 3–24 Transmit Automatic Polling Bits

<table>
<thead>
<tr>
<th>CSR0&lt;19:17&gt;</th>
<th>Time Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>No transmit automatic polling; CSR1 access should be used to poll the transmit descriptor list.</td>
</tr>
<tr>
<td>001</td>
<td>Transmit automatic polling every 200 $\mu$s.</td>
</tr>
<tr>
<td>010</td>
<td>Transmit automatic polling every 800 $\mu$s.</td>
</tr>
<tr>
<td>011</td>
<td>Transmit automatic polling every 1.6 ms.</td>
</tr>
<tr>
<td>100</td>
<td>Transmit automatic polling every 12.8 $\mu$s.</td>
</tr>
<tr>
<td>101</td>
<td>Transmit automatic polling every 25.6 $\mu$s.</td>
</tr>
<tr>
<td>110</td>
<td>Transmit automatic polling every 51.2 $\mu$s.</td>
</tr>
<tr>
<td>111</td>
<td>Transmit automatic polling every 102.4 $\mu$s.</td>
</tr>
</tbody>
</table>

Table 3–25 lists the CSR0 read and write access rules.

Table 3–25 CSR0 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FF 800000H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>To write, the transmit and receive processes must be stopped. If one or both of the processes is not stopped, the result is UNPREDICTABLE.</td>
</tr>
</tbody>
</table>

Table 3–26 defines the cache address alignment bits.
3.2 CSR Operation

Table 3–26 Cache Alignment Bits
<table>
<thead>
<tr>
<th>CSR0&lt;15:14&gt;</th>
<th>Address Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Reserved</td>
</tr>
<tr>
<td>01</td>
<td>8-longword boundary alignment</td>
</tr>
<tr>
<td>10</td>
<td>16-longword boundary alignment</td>
</tr>
<tr>
<td>11</td>
<td>32-longword boundary alignment</td>
</tr>
</tbody>
</table>

3.2.2.2 Transmit Poll Demand Register (CSR1–Offset 08H)

Figure 3-12 shows the CSR1 bit field and Table 3–27 describes the bit field.

Figure 3–12 CSR1 Transmit Poll Demand Register

Table 3–27 CSR1 Transmit Poll Demand Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:0</td>
<td><strong>TPD — Transmit Poll Demand (Write Only)</strong></td>
</tr>
<tr>
<td></td>
<td>When written with any value, the 21140A checks for frames to be transmitted. If no descriptor is available, the transmit process returns to the suspended state and CSR5&lt;2&gt; is not asserted. If the descriptor is available, the transmit process resumes.</td>
</tr>
</tbody>
</table>

Table 3–28 lists the CSR1 read and write access rules.

Table 3–28 CSR1 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FFFFFFFFH</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>Effective only if the transmit process is in the suspended state.</td>
</tr>
</tbody>
</table>
3.2 CSR Operation

3.2.2.3 Receive Poll Demand Register (CSR2–Offset 10H)

Figure 3–13 shows the CSR2 bit field and Table 3–29 describes the bit field.

Figure 3–13  CSR2 Receive Poll Demand Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:0</td>
<td>RPD — Receive Poll Demand (Write Only)</td>
</tr>
</tbody>
</table>

When written with any value, the 21140A checks for receive descriptors to be acquired. If no descriptor is available, the receive process returns to the suspended state and CSR5<7> is not asserted. If the descriptor is available, the receive process resumes.

Table 3–30 lists the access rules for CSR2.

Table 3–30  CSR2 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FFFFFFFFH</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>Effective only if the receive process is in the suspended state.</td>
</tr>
</tbody>
</table>

3.2.2.4 Descriptor List Address Registers (CSR3–Offset 18H and CSR4–Offset 20H)

The CSR3 descriptor list address register is used for receive buffer descriptors and the CSR4 descriptor list address register is used for transmit buffer descriptors. In both cases, the registers are used to point the 21140A to the start of the appropriate descriptor list.

Figure 3-14 shows the CSR3 bit field and Table 3-31 describes the bit field.
3.2 CSR Operation

**Note**

The descriptor lists reside in physical memory space and must be longword aligned. The 21140A behaves UNPREDICTABLY when the lists are not longword aligned.

Writing to either CSR3 or CSR4 is permitted only when its respective process is in the stopped state. When stopped, the CSR3 and CSR4 registers must be written before the respective START command is given (Section 3.2.2.6).

**Figure 3–14 CSR3 Receive List Base Address Register**

<table>
<thead>
<tr>
<th>31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</th>
<th>Start of Receive List</th>
</tr>
</thead>
</table>

**Table 3–31 CSR3 Receive List Base Address Register Description**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:2</td>
<td>Start of Receive List</td>
</tr>
<tr>
<td>1:0</td>
<td>Must be 00 for longword alignment.</td>
</tr>
</tbody>
</table>

Table 3–32 lists the access rules for CSR3.

**Table 3–32 CSR3 Access Rules**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>UNPREDICTABLE</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>Receive process stopped</td>
</tr>
</tbody>
</table>

Figure 3–15 shows the CSR4 bit field and Table 3–33 describes the bit field.
3.2 CSR Operation

Figure 3–15 CSR4 Transmit List Base Address Register

| 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |
| 31:2 Start of Transmit List |
| 1:0 Must be 00 for longword alignment. |

Table 3–33 CSR4 Transmit List Base Address Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:2 Start of Transmit List</td>
<td></td>
</tr>
<tr>
<td>1:0 Must be 00 for longword alignment.</td>
<td></td>
</tr>
</tbody>
</table>

Table 3–34 lists the access rules for CSR4.

Table 3–34 CSR4 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>UNPREDICTABLE</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>Transmit process stopped</td>
</tr>
</tbody>
</table>

3.2.2.5 Status Register (CSR5–Offset 28H)

The status register CSR5 contains all the status bits that the 21140A reports to the host. CSR5 is usually read by the driver during interrupt service routine or polling. Most of the fields in this register cause the host to be interrupted. CSR5 bits are not cleared when read. Writing 1 to these bits clears them; writing 0 has no effect. Each field can be masked (Section 3.2.2.7).

Figure 3–16 shows the CSR5 bit fields and Table 3–35 describes the bit fields.
3.2 CSR Operation

Figure 3–16  CSR5 Status Register

Table 3–35  CSR5 Status Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>25:23</td>
<td>EB — Error Bits (Read Only)</td>
</tr>
<tr>
<td></td>
<td>Indicates the type of error that caused system error. Valid only when fatal bus error CSR5&lt;13&gt; is set (Table 3–36).</td>
</tr>
<tr>
<td></td>
<td>This field does not generate an interrupt.</td>
</tr>
<tr>
<td>22:20</td>
<td>TS — Transmission Process State (Read Only)</td>
</tr>
<tr>
<td></td>
<td>Indicates the state of the transmit process (Table 3–37). This field does not generate an interrupt.</td>
</tr>
</tbody>
</table>

(continued on next page)
3.2 CSR Operation

Table 3–35 (Cont.) CSR5 Status Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19:17</td>
<td><strong>RS — Receive Process State (Read Only)</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates the state of the receive process (Table 3–38). This field does not generate an interrupt.</td>
</tr>
<tr>
<td>16</td>
<td><strong>NIS — Normal Interrupt Summary</strong></td>
</tr>
<tr>
<td></td>
<td>Normal interrupt summary bit. Its value is the logical OR of:</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;0&gt; — Transmit interrupt</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;2&gt; — Transmit buffer unavailable</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;6&gt; — Receive interrupt</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;11&gt; — General-purpose timer expired</td>
</tr>
<tr>
<td></td>
<td>Only unmasked bits affect the normal interrupt summary CSR5&lt;16&gt; bit.</td>
</tr>
<tr>
<td>15</td>
<td><strong>AIS — Abnormal Interrupt Summary</strong></td>
</tr>
<tr>
<td></td>
<td>Abnormal interrupt summary bits. Its value is the logical OR of:</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;1&gt; — Transmit process stopped</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;3&gt; — Transmit jabber timeout</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;5&gt; — Transmit underflow</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;7&gt; — Receive buffer unavailable</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;8&gt; — Receive process stopped</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;9&gt; — Receive watchdog timeout</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;10&gt; — Early transmit interrupt</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;13&gt; — Fatal bus error</td>
</tr>
<tr>
<td></td>
<td>Only unmasked bits affect the abnormal interrupt summary CSR5&lt;15&gt; bit.</td>
</tr>
<tr>
<td>13</td>
<td><strong>FBE — Fatal Bus Error</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates that a system error occurred (Table 3–36). If a system error occurs, the 21140A disables all bus accesses.</td>
</tr>
<tr>
<td>11</td>
<td><strong>GTE — General-Purpose Timer Expired</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates that the general-purpose timer (CSR11) counter has expired. This timer is mainly used by the software driver.</td>
</tr>
<tr>
<td>10</td>
<td><strong>ETI — Early Transmit Interrupt</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates that the packet to be transmitted was fully transferred into the chip’s internal transmit FIFOs. Transmit interrupt (CSR5&lt;0&gt;) automatically clears this bit.</td>
</tr>
</tbody>
</table>

(continued on next page)
3.2 CSR Operation

Table 3–35 (Cont.) CSR5 Status Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 9     | RWT — Receive Watchdog Timeout  
This bit reflects the line status and indicates that the receive watchdog timer has expired while another node is still active on the network. In case of overflow, the long packets may not be received. |
| 8     | RPS — Receive Process Stopped  
Asserts when the receive process enters the stopped state. |
| 7     | RU — Receive Buffer Unavailable  
Indicates that the next descriptor in the receive list is owned by the host and cannot be acquired by the 21140A. The reception process is suspended. To resume processing receive descriptors, the host should change the ownership of the descriptor and may issue a receive poll demand command. If no receive poll demand is issued, the reception process resumes when the next recognized incoming frame is received.  
After the first assertion, CSR5<7> is not asserted for any subsequent not owned receive descriptors fetches. CSR5<7> asserts only when the previous receive descriptor was owned by the 21140A. |
| 6     | RI — Receive Interrupt  
Indicates the completion of a frame reception. Specific frame status information has been posted in the descriptor. The reception process remains in the running state. |
| 5     | UNF — Transmit Underflow  
Indicates that the transmit FIFO had an underflow condition during the packet transmission. The transmit process is placed in the suspended state and underflow error TDES0<1> is set. |
| 3     | TJT — Transmit Jabber Timeout  
Indicates that the transmit jabber timer expired, meaning that the 21140A transmitter had been excessively active. The transmission process is aborted and placed in the stopped state. This event causes the transmit jabber timeout TDES0<14> flag to assert. |
| 2     | TU — Transmit Buffer Unavailable  
Indicates that the next descriptor on the transmit list is owned by the host and cannot be acquired by the 21140A. The transmission process is suspended.  
Table 4–14 explains the transmit process state transitions. To resume processing transmit descriptors, the host should change the ownership bit of the descriptor and then issue a transmit poll demand command, unless transmit automatic polling (Table 3–24) is enabled.  
(continued on next page)
3.2 CSR Operation

Table 3–35 (Cont.) CSR5 Status Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TPS — Transmit Process Stopped</td>
</tr>
<tr>
<td></td>
<td>Asserts when the transmit process enters the stopped state.</td>
</tr>
<tr>
<td>0</td>
<td>TI — Transmit Interrupt</td>
</tr>
<tr>
<td></td>
<td>Indicates that a frame transmission was completed, while TDES1&lt;31&gt; is asserted in the first descriptor of the frame.</td>
</tr>
</tbody>
</table>

Table 3–36 lists the bit codes for the fatal bus error bits.

Table 3–36 Fatal Bus Error Bits

<table>
<thead>
<tr>
<th>CSR5&lt;25:23&gt;</th>
<th>Process State</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Parity error(^1)</td>
</tr>
<tr>
<td>001</td>
<td>Master abort</td>
</tr>
<tr>
<td>010</td>
<td>Target abort</td>
</tr>
<tr>
<td>011</td>
<td>Reserved</td>
</tr>
<tr>
<td>1xx</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

\(^1\)The only way to recover from a parity error is by setting software reset (CSR0<0>=1).

Table 3–37 lists the bit codes for the transmit process state.

Table 3–37 Transmit Process State

<table>
<thead>
<tr>
<th>CSR5&lt;22:20&gt;</th>
<th>Process State</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Stopped — RESET command or transmit jabber expired</td>
</tr>
<tr>
<td>001</td>
<td>Running — Fetching transmit descriptor</td>
</tr>
<tr>
<td>010</td>
<td>Running — Waiting for end of transmission</td>
</tr>
<tr>
<td>011</td>
<td>Running — Reading buffer from memory and queuing the data into the transmit FIFO</td>
</tr>
<tr>
<td>100</td>
<td>Reserved</td>
</tr>
<tr>
<td>101</td>
<td>Running — Setup packet</td>
</tr>
<tr>
<td>110</td>
<td>Suspended — Transmit FIFO underflow, or an unavailable transmit descriptor</td>
</tr>
<tr>
<td>111</td>
<td>Running — Closing transmit descriptor</td>
</tr>
</tbody>
</table>
3.2 CSR Operation

Table 3–38 lists the bit codes for the receive process state.

Table 3–38  Receive Process State

<table>
<thead>
<tr>
<th>CSR5&lt;19:17&gt;</th>
<th>Process State</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Stopped — RESET or STOP RECEIVE command</td>
</tr>
<tr>
<td>001</td>
<td>Running — Fetching receive descriptor</td>
</tr>
<tr>
<td>010</td>
<td>Running — Checking for end of receive packet before prefetch of next descriptor</td>
</tr>
<tr>
<td>011</td>
<td>Running — Waiting for receive packet</td>
</tr>
<tr>
<td>100</td>
<td>Suspended — Unavailable receive buffer</td>
</tr>
<tr>
<td>101</td>
<td>Running — Closing receive descriptor</td>
</tr>
<tr>
<td>110</td>
<td>Running — Flushing the current frame from the receive FIFO because of unavailable receive buffer</td>
</tr>
<tr>
<td>111</td>
<td>Running — Queuing the receive frame from the receive FIFO into the receive buffer</td>
</tr>
</tbody>
</table>

Table 3–39 lists the access rules for CSR5.

Table 3–39  CSR5 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FC000000H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>CSR5 bits 0 through 16 are cleared by writing 1. Writing 0 to these bits has no effect. Writing to CSR5 bits 17 through 25 has no effect.</td>
</tr>
</tbody>
</table>

3.2.2.6 Operation Mode Register (CSR6—Offset 30H)

CSR6 establishes the receive and transmit operating modes and commands. CSR6 should be the last CSR to be written as part of initialization. Figure 3–17 shows the CSR6 bit fields and Table 3-40 describes the bit fields.
3.2 CSR Operation

Figure 3–17 CSR6 Operating Mode Register

SC - Special Capture Effect Enable
RA - Receive All
MBZ - Must Be Zero
MB1 - Must Be One
SCR - Scrambler Mode
PCS - PCS Function
TTM - Transmit Threshold Mode
SF - Store and Forward
MBZ - Must Be Zero
HBD - Heartbeat Disable
PS - Port Select
CA - Capture Effect Enable
MBZ - Must Be Zero
TR - Threshold Control Bits
ST - Start/Stop Transmission Command
FC - Force Collision Mode
OM - Operating Mode
FD - Full-Duplex Mode
MBZ - Must Be Zero
PM - Pass All Multicast
PR - Promiscuous Mode
SB - Start/Stop Backoff Counter
IF - Inverse Filtering
PB - Pass Bad Frames
HO - Hash-Only Filtering Mode
SR - Start/Stop Receive
HP - Hash/Perfect Receive Filtering Mode
### 3.2 CSR Operation

#### Table 3–40 CSR6 Operating Mode Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 31    | **SC — Special Capture Effect Enable**  
When set, enables the enhanced resolution of capture effect on the network (Section 6.6). Digital recommends that this bit be set together with CSR6<17>.  
When clear, the 21140A disables the enhanced resolution of capture effect on the network. |
| 30    | **RA — Receive All**  
When set, all incoming packets will be received, regardless of the destination address. The address match is checked according to Table 3–44, and is reported in RDES0<30>. |
| 28:26 | **MBZ — Must Be Zero**  
These bits are not used by the 21140A and should always be programmed to zero. |
| 25    | **MBO — Must Be One**  
This bit should always be programmed to one. |
| 24    | **SCR — Scrambler Mode**  
When set, the scrambler function is active and the MII/SYM port transmits and receives scrambled symbols.  
Changing this bit during operation may cause unpredictable behavior. |
| 23    | **PCS — PCS Function**  
When set, the PCS functions are active and the MII/SYM port operates in symbol mode. All MII/SYM port control signals are generated internally.  
When reset, the MII/SYM port is not selected and port select (CSR6<18>) is also reset.  
Changing this bit during operation may cause unpredictable behavior. |
| 22    | **TTM — Transmit Threshold Mode**  
Selects the transmit FIFO threshold to be either 10Mb/s or 100Mb/s (Table 3–41).  
When set, the threshold is 10Mb/s. When reset, the threshold is 100Mb/s.  
The transmit process must be in the stopped state to change this bit (Table 3–42). |

(continued on next page)
### 3.2 CSR Operation

#### Table 3–40 (Cont.) CSR6 Operating Mode Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21 SF — Store and Forward</td>
<td>When set, transmission starts when a full packet resides in the FIFO. When this occurs, the threshold values specified in CSR6&lt;15:14&gt; are ignored. The transmit process must be in the stopped state to change this bit.</td>
</tr>
<tr>
<td>20 MBZ — Must Be Zero</td>
<td>This bit is not used by the 21140A and should always be programmed to zero.</td>
</tr>
<tr>
<td>19 HBD — Heartbeat Disable</td>
<td>When set, the heartbeat signal quality (SQE) generator function is disabled. This bit should be set in the MII/SYM mode.</td>
</tr>
<tr>
<td>18 PS — Port Select</td>
<td>When reset, the SRL port is selected. When set, the MII/SYM port is selected. During a hardware reset, this bit automatically resets (Table 3-42). A software reset does not affect this bit. After this bit state is changed, a software reset should be performed and both the transmit and receive processes should be initialized.</td>
</tr>
<tr>
<td>17 CA — Capture Effect Enable</td>
<td>When set, enables the resolution of the capture effect on the network (Section 6.6). When reset, the 21140A disables the resolution of the capture effect on the network.</td>
</tr>
<tr>
<td>16 MBZ — Must Be Zero</td>
<td>This bit is not used by the 21140A and should always be programmed to zero.</td>
</tr>
<tr>
<td>15:14 TR — Threshold Control Bits</td>
<td>Controls the selected threshold level for the 21140A transmit FIFO. Four threshold levels are allowed (Table 3-41). The threshold value has a direct impact on the 21140A bus arbitration scheme (Section 4.3.2). Transmission starts when the frame size within the transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. The transmit process must be in the stopped state to change these bits (CSR6&lt;15:14&gt;).</td>
</tr>
</tbody>
</table>

(continued on next page)
3.2 CSR Operation

Table 3–40 (Cont.) CSR6 Operating Mode Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>ST — Start/Stop Transmission Command</td>
</tr>
<tr>
<td></td>
<td>When set, the transmission process is placed in the running state, and the 21140A checks the transmit list at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the transmit list base address set by CSR4, or from the position retained when the transmit process was previously stopped. If no descriptor can be acquired, the transmit process enters the suspended state. If the current descriptor is not owned by the 21140A, the transmission process enters the suspended state and transmit buffer unavailable CSR5&lt;2&gt; is set. The start transmission command is affected only when the transmission process is stopped. If the command is issued before setting CSR4, the 21140A will behave UNPREDICTABLY. When reset, the transmission process is placed in the stopped state after completing the transmission of the current frame. The next descriptor position in the transmit list is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only when the transmission process is in either the running or suspended state (Table 4-14).</td>
</tr>
</tbody>
</table>

| 12    | FC — Force Collision Mode |
|       | Allows the collision logic to be tested. Meaningful only in internal loopback mode. When set, a collision is forced during the next transmission attempt. This results in 16 transmission attempts with excessive collision reported in the transmit descriptor (TDES0<8>). |

| 11:10 | OM — Operating Mode |
|       | Selects the 21140A loopback operation modes (Table 3–43). |

| 9     | FD — Full-Duplex Mode |
|       | When set, the 21140A operates in a full-duplex mode (Section 6.5). The 21140A can transmit and receive simultaneously. Setting the 21140A to operate in full-duplex mode is allowed only if the transmit and receive processes are in the stopped state, and start/stop receive (CSR6<1>) and start/stop transmission commands (CSR6<13>) are both set to 0. While in full-duplex mode, heartbeat check is disabled, heartbeat fail TDES0<7> should be ignored, and internal loopback is not allowed. |

| 8     | MBZ — Must Be Zero |
|       | This bit is not used by the 21140A and should always be programmed to zero. |
### 3.2 CSR Operation

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><strong>PM — Pass All Multicast</strong>&lt;br&gt;When set, indicates that all the incoming frames with a multicast destination address (first bit in the destination address field is 1) are received. Incoming frames with physical address destinations are filtered according to the CSR6&lt;0&gt; bit.</td>
</tr>
<tr>
<td>6</td>
<td><strong>PR — Promiscuous Mode</strong>&lt;br&gt;When set, indicates that any incoming valid frame is received, regardless of its destination address.&lt;br&gt;After reset, the 21140A wakes up in promiscuous mode.</td>
</tr>
<tr>
<td>5</td>
<td><strong>SB — Start/Stop Backoff Counter</strong>&lt;br&gt;When set, indicates that the internal backoff counter stops counting when any carrier activity is detected. The 21140A backoff counter resumes when the carrier drops. The earliest the 21140A starts its transmission after carrier deassertion is 9.6 µs for 10Mb/s data rate or 0.96 µs for 100Mb/s data rate.&lt;br&gt;When reset, the internal backoff counter is not affected by the carrier activity.</td>
</tr>
<tr>
<td>4</td>
<td><strong>IF — Inverse Filtering (Read Only)</strong>&lt;br&gt;When set, the 21140A operates in an inverse filtering mode. This is valid only during perfect filtering mode (Table 3–44 and Table 4–8).</td>
</tr>
<tr>
<td>3</td>
<td><strong>PB — Pass Bad Frames</strong>&lt;br&gt;When set, the 21140A operates in pass bad frame mode. All incoming frames that passed the address filtering are received, including runt frames, collided fragments, or truncated frames caused by FIFO overflow.&lt;br&gt;If any received bad frames are required, promiscuous mode (CSR6&lt;6&gt;) should be set to 1.</td>
</tr>
<tr>
<td>2</td>
<td><strong>HO — Hash-Only Filtering Mode (Read Only)</strong>&lt;br&gt;When set, the 21140A operates in an imperfect address filtering mode for both physical and multicast addresses (Table 4–8).</td>
</tr>
</tbody>
</table>

(continued on next page)
3.2 CSR Operation

Table 3–40 (Cont.) CSR6 Operating Mode Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1     | **SR — Start/Stop Receive**  
When set, the receive process is placed in the running state. The 21140A attempts to acquire a descriptor from the receive list and processes incoming frames.  
Descriptor acquisition is attempted from the current position in the list, which is the address set by CSR3 or the position retained when the receive process was previously stopped. If no descriptor is owned by the 21140A, the receive process enters the suspended state and receive buffer unavailable (CSR5<7>) sets.  
The start reception command is affected only when the reception process has stopped. If the command was issued before setting CSR3, the 21140A behaves UNPREDICTABLY.  
When cleared, the receive process enters the stopped state after completing the reception of the current frame. The next descriptor position in the receive list is saved, and becomes the current position after the receive process is restarted. The stop reception command is effective only when the receive process is in running or suspended state (Table 4–13). |
| 0     | **HP — Hash/Perfect Receive Filtering Mode (Read Only)**  
When reset, the 21140A does a perfect address filter of incoming frames according to the addresses specified in the setup frame (Table 4–8).  
When set, the 21140A does imperfect address filtering of multicast incoming frames according to the hash table specified in the setup frame. If CSR6<2> is set, then physical addresses are imperfect address filtered too. If CSR6<2> is reset, physical addresses are perfect address filtered, according to a single physical address, as specified in the setup frame.  

Note  
A unique condition occurs in hash/perfect receive filtering mode when a physical address is perfect filtered by comparing it to a single address. In this case, while in perfect filtering mode, the driver sends a setup frame containing the single physical address. The driver then sets the pass all multicast mode. |

Table 3–41 lists the threshold values in bytes.
### 3.2 CSR Operation

#### Table 3–41 Transmit Threshold

<table>
<thead>
<tr>
<th>CSR6&lt;21&gt;</th>
<th>CSR6&lt;15:14&gt;</th>
<th>CSR6&lt;18&gt; = 0</th>
<th>CSR6&lt;18&gt; = 1</th>
<th>CSR6&lt;18&gt; = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CSR6&lt;22&gt; = X</td>
<td>CSR6&lt;22&gt; = 1</td>
<td>CSR6&lt;22&gt; = 0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>00</td>
<td>72</td>
<td>72</td>
<td>128</td>
</tr>
<tr>
<td>0</td>
<td>01</td>
<td>96</td>
<td>96</td>
<td>256</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>128</td>
<td>128</td>
<td>512</td>
</tr>
<tr>
<td>0</td>
<td>11</td>
<td>160</td>
<td>160</td>
<td>1024</td>
</tr>
<tr>
<td>1</td>
<td>XX</td>
<td>Store and</td>
<td>Store and</td>
<td>Store and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>forward</td>
<td>forward</td>
<td>forward</td>
</tr>
</tbody>
</table>

Table 3–42 lists the port and data rate selection.

#### Table 3–42 Port and Data Rate Selection

<table>
<thead>
<tr>
<th>CSR6&lt;18&gt;</th>
<th>CSR6&lt;22&gt;</th>
<th>CSR6&lt;23&gt;</th>
<th>CSR6&lt;24&gt;</th>
<th>Active Port</th>
<th>Data Rate</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>SRL</td>
<td>10M b/s</td>
<td>Conventional 10M b/s ENDEC interface</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>MII/SYM</td>
<td>10M b/s</td>
<td>MII with transmit FIFO thresholds appropriate for 10M b/s</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>MII/SYM</td>
<td>100M b/s</td>
<td>MII with transmit FIFO thresholds appropriate for 100M b/s</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>MII/SYM</td>
<td>100M b/s</td>
<td>PCS function for 100BASE-FX</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>MII/SYM</td>
<td>100M b/s</td>
<td>PCS and scrambler functions for 100BASE-TX</td>
</tr>
</tbody>
</table>

Table 3–43 selects the 21140A loopback operation modes.
3.2 CSR Operation

Table 3–43  Loopback Operation Mode

<table>
<thead>
<tr>
<th>CSR6&lt;11:10&gt;</th>
<th>Operation Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Normal</td>
</tr>
<tr>
<td>01&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Internal loopback</td>
</tr>
<tr>
<td>10</td>
<td>External loopback</td>
</tr>
</tbody>
</table>

<sup>1</sup>Internal loopback is performed on the serial and MII/SYM ports. If enabled by CSR6, it also tests the PCS functions (CSR6<23>) and the scrambler function (CSR6<24>). Note that when internal loopback is performed on the SYM port, symbols appear on the network. When internal loopback is performed on the MII port, the mii_txen signal is disabled.

Table 3–44 lists the codes to determine the filtering mode.

Table 3–44  Filtering Mode

<table>
<thead>
<tr>
<th>CSR6&lt;7&gt;</th>
<th>CSR6&lt;6&gt;</th>
<th>CSR6&lt;4&gt;</th>
<th>CSR6&lt;2&gt;</th>
<th>CSR6&lt;0&gt;</th>
<th>Filtering Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16 perfect filtering</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>512-bit hash + 1 perfect filtering</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>512-bit hash for multicast and physical addresses</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Inverse filtering</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>Promiscuous</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Promiscuous</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>Pass all multicast</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Pass all multicast</td>
</tr>
</tbody>
</table>

Note

When CSR6<30> is set (receive all mode), this table is used to generate the address match status reported in RDES0<30>.

Table 3–45 describes the only conditions that permit change to a field when modifying values to CSR6.
3.2 CSR Operation

Table 3–45  CSR6 Access Rules

<table>
<thead>
<tr>
<th>Category Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>32000040</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td></td>
</tr>
<tr>
<td>* CSR6&lt;22&gt;</td>
<td>Receive and transmit processes stopped</td>
</tr>
<tr>
<td>* CSR6&lt;21&gt;</td>
<td>Receive and transmit processes stopped</td>
</tr>
<tr>
<td>* CSR6&lt;17&gt;</td>
<td>Receive and transmit processes stopped</td>
</tr>
<tr>
<td>* CSR6&lt;16&gt;</td>
<td>Receive and transmit processes stopped</td>
</tr>
<tr>
<td>* CSR6&lt;15:14&gt;</td>
<td>Transmit process stopped</td>
</tr>
<tr>
<td>* CSR6&lt;12&gt;</td>
<td>Receive and transmit processes stopped</td>
</tr>
<tr>
<td>* CSR6&lt;11:10&gt;</td>
<td>Receive and transmit processes stopped</td>
</tr>
<tr>
<td>* CSR6&lt;9&gt;</td>
<td>Receive and transmit processes stopped</td>
</tr>
<tr>
<td>* CSR6&lt;8&gt;</td>
<td>Transmit process stopped</td>
</tr>
<tr>
<td>* CSR6&lt;5&gt;</td>
<td>Receive and transmit processes stopped</td>
</tr>
<tr>
<td>* CSR6&lt;3&gt;</td>
<td>Receive process stopped</td>
</tr>
<tr>
<td>* Start_Transmit CSR6&lt;13&gt;=1</td>
<td>CSR4 initialized</td>
</tr>
<tr>
<td>* Stop_Transmit CSR6&lt;13&gt;=0</td>
<td>Transmit running or suspended</td>
</tr>
<tr>
<td>* Start_Receive CSR6&lt;1&gt;=1</td>
<td>CSR3 initialized</td>
</tr>
<tr>
<td>* Stop_Receive CSR6&lt;1&gt;=0</td>
<td>Receive running or suspended</td>
</tr>
</tbody>
</table>

3.2.2.7 Interrupt Enable Register (CSR7–Offset 38H)

The interrupt enable register (CSR7) enables the interrupts reported by CSR5 (Section 3.2.2.5). Setting a bit to 1 enables a corresponding interrupt. After a hardware or software reset, all interrupts are disabled. Figure 3–18 shows the CSR7 bit fields and Table 3–46 describes the bit fields.
3.2 CSR Operation

Figure 3–18 CSR7 Interrupt Enable Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td><strong>NI — Normal Interrupt Summary Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set, normal interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset, no normal interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>This bit (CSR7&lt;16&gt;) enables the following bits:</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;0&gt; — Transmit interrupt</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;2&gt; — Transmit buffer unavailable</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;6&gt; — Receive interrupt</td>
</tr>
<tr>
<td></td>
<td>CSR5&lt;11&gt; — General-purpose timer expired</td>
</tr>
</tbody>
</table>

(continued on next page)
### 3.2 CSR Operation

#### Table 3–46 (Cont.) CSR7 Interrupt Enable Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| **15** | AI — Abnormal Interrupt Summary Enable  
When set, abnormal interrupt is enabled.  
When reset, no abnormal interrupt is enabled. This bit CSR7<15> enables the following bits:  
- CSR5<1> — Transmit process stopped  
- CSR5<3> — Transmit jabber timeout  
- CSR5<5> — Transmit underflow  
- CSR5<7> — Receive buffer unavailable  
- CSR5<8> — Receive process stopped  
- CSR5<9> — Receive watchdog timeout  
- CSR5<10> — Early transmit interrupt  
- CSR5<13> — Fatal bus error |
| **13** | FBE — Fatal Bus Error Enable  
When set together with abnormal interrupt summary enable (CSR7<15>) and fatal bus error (CSR5<13>), the interrupt is enabled.  
When reset and fatal bus error (CSR5<13>) is set, the interrupt is disabled. |
| **11** | GPT — General-Purpose Timer Enable  
When set together with abnormal interrupt summary enable (CSR7<15>) and general-purpose timer expired (CSR5<11>), the interrupt is enabled.  
When reset and general-purpose timer expired CSR5<11> is set, the interrupt is disabled. |
| **10** | ETE — Early Transmit Interrupt Enable  
When set together with normal interrupt summary enable (CSR7<16>) and early transmit interrupt enable (CSR5<10>), the interrupt is enabled.  
When reset and early transmit interrupt (CSR5<10>) is set, the interrupt is disabled. |
| **9** | RW — Receive Watchdog Timeout Enable  
When set together with abnormal interrupt summary enable (CSR7<15>) and receive watchdog timeout (CSR5<9>), the interrupt is enabled.  
When reset and receive watchdog timeout (CSR5<9>) is set, the interrupt is disabled. |

(continued on next page)
### 3.2 CSR Operation

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td><strong>RS — Receive Stopped Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set together with abnormal interrupt summary enable (CSR7&lt;15&gt;) and receive stopped (CSR5&lt;8&gt;), the interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset and receive stopped (CSR5&lt;8&gt;) is set, the interrupt is disabled.</td>
</tr>
<tr>
<td>7</td>
<td><strong>RU — Receive Buffer Unavailable Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set together with abnormal interrupt summary enable (CSR7&lt;15&gt;) and receive buffer unavailable (CSR5&lt;7&gt;), the interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset and receive buffer unavailable (CSR5&lt;7&gt;) is set, the interrupt is disabled.</td>
</tr>
<tr>
<td>6</td>
<td><strong>RI — Receive Interrupt Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set together with normal interrupt summary enable (CSR7&lt;16&gt;) and receive interrupt bit (CSR5&lt;6&gt;), the interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset and receive interrupt (CSR5&lt;6&gt;) is set, the interrupt is disabled.</td>
</tr>
<tr>
<td>5</td>
<td><strong>UN — Underflow Interrupt Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set together with abnormal interrupt summary enable (CSR7&lt;15&gt;) and transmit underflow (CSR5&lt;5&gt;), the interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset and transmit underflow (CSR5&lt;5&gt;) is set, the interrupt is disabled.</td>
</tr>
<tr>
<td>3</td>
<td><strong>TJ — Transmit Jabber Timeout Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set together with abnormal interrupt summary enable (CSR7&lt;15&gt;) and transmit jabber timeout (CSR5&lt;3&gt;), the interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset and transmit jabber timeout (CSR5&lt;3&gt;) is set, the interrupt is disabled.</td>
</tr>
<tr>
<td>2</td>
<td><strong>TU — Transmit Buffer Unavailable Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set together with normal interrupt summary enable (CSR7&lt;16&gt;) and transmit buffer unavailable (CSR5&lt;2&gt;), the interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset and transmit buffer unavailable (CSR5&lt;2&gt;) is set, the interrupt is disabled.</td>
</tr>
<tr>
<td>1</td>
<td><strong>TS — Transmit Stopped Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set together with abnormal interrupt summary enable (CSR7&lt;15&gt;) and transmission stopped (CSR5&lt;1&gt;), the interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset and transmission stopped (CSR5&lt;1&gt;) is set, the interrupt is disabled.</td>
</tr>
</tbody>
</table>

(continued on next page)
3.2 CSR Operation

Table 3–46 (Cont.)  CSR7 Interrupt Enable Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><strong>TI — Transmit Interrupt Enable</strong></td>
</tr>
<tr>
<td></td>
<td>When set together with normal interrupt summary enable (CSR7&lt;16&gt;) and transmit interrupt (CSR5&lt;0&gt;), the interrupt is enabled.</td>
</tr>
<tr>
<td></td>
<td>When reset and transmit interrupt (CSR5&lt;0&gt;) is set, the interrupt is disabled.</td>
</tr>
</tbody>
</table>

Table 3–47 lists the access rules for CSR7.

Table 3–47  CSR7 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FFFE0000H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.2.2.8 Missed Frames and Overflow Counter (CSR8–Offset 40H)

Figure 3–19 shows the CSR8 bit fields and Table 3–48 describes the bit fields.

Figure 3–19  CSR8 Missed Frames and Overflow Counter
3.2 CSR Operation

Table 3–48 CSR8 Missed Frames and Overflow Counter Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>Overflow Counter Overflow (Read Only)</td>
</tr>
<tr>
<td></td>
<td>Sets when the overflow counter overflows; resets when CSR8 is read.</td>
</tr>
<tr>
<td>27:17</td>
<td>Overflow Counter (Read Only)</td>
</tr>
<tr>
<td></td>
<td>Indicates the number of frames discarded because of overflow. The counter clears when read.</td>
</tr>
<tr>
<td>16</td>
<td>Missed Frame Overflow (Read Only)</td>
</tr>
<tr>
<td></td>
<td>Sets when the missed frame counter overflows; resets when CSR8 is read.</td>
</tr>
<tr>
<td>15:0</td>
<td>Missed Frame Counter (Read Only)</td>
</tr>
<tr>
<td></td>
<td>Indicates the number of frames discarded because no host receive descriptors were available. The counter clears when read.</td>
</tr>
</tbody>
</table>

Table 3–49 lists the access rules for CSR8.

Table 3–49 CSR8 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>00000000H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>Not possible</td>
</tr>
</tbody>
</table>

3.2.2.9 Boot ROM, Serial ROM, and MII Management Register (CSR9–Offset 48H)

This register provides an interface to the boot ROM, serial ROM, and MII management. It selects the device and contains both the commands and data to be read from and stored in the boot ROM and serial ROM. The MII management selects an operation mode for reading and writing the MII.

Figure 3–20 shows the boot ROM, serial ROM, and MII management register and Table 3–50 describes the register bit fields.
3.2 CSR Operation

Table 3–50 CSR9 Boot ROM, Serial ROM, and MII Management Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>MDI — MII Management Data_In</td>
</tr>
<tr>
<td></td>
<td>Used by the 21140A to read data from the PHY.</td>
</tr>
<tr>
<td>18</td>
<td>MII — MII Management Operation Mode</td>
</tr>
<tr>
<td></td>
<td>Defines the operation mode (read or write) of the PHY.</td>
</tr>
<tr>
<td>17</td>
<td>MDO — MII Management Write Data</td>
</tr>
<tr>
<td></td>
<td>Specifies the value of the data that the 21140A writes to the PHY.</td>
</tr>
<tr>
<td>16</td>
<td>MDC — MII Management Clock</td>
</tr>
<tr>
<td></td>
<td>MII management data clock (mii_mdc) is an output signal to the PHY. It is used as a timing reference.</td>
</tr>
<tr>
<td>14</td>
<td>RD — Read Operation</td>
</tr>
<tr>
<td></td>
<td>Read control bit. When set together with CSR9&lt;12&gt;, the 21140A performs read cycles from the boot ROM, the serial ROM, and the external register.</td>
</tr>
</tbody>
</table>

(continued on next page)
### 3.2 CSR Operation

Table 3-50 (Cont.)  CSR9 Boot ROM, Serial ROM, and MII Management

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 13    | WR — Write Operation  
Write control bit. When set together with CSR9<12>, the 21140A performs write cycles to the boot ROM, the serial ROM, and the external register. |
| 12    | BR — Boot ROM Select  
When set, the 21140A selects the boot ROM. |
| 11    | SR — Serial ROM Select  
When set, together with either serial ROM read operation (CSR9<14>) or serial ROM write operation (CSR9<13>), the 21140A selects the serial ROM. |
| 10    | REG — External Register Select  
When set, the 21140A selects an external register (Section 7.5). |
| 7:0   | DATA — Boot ROM Data or Serial ROM Control  
If the boot ROM is selected (CSR9<12> is set), this field contains the data to be read from and written to the boot ROM.  
If the serial ROM is selected, CSR9<3:0> bits are connected to the serial ROM control pins as follows:  
  - **Bit 3, Data Out** — When set, serially shifts the read data from the serial ROM device to the 21140A.  
  - **Bit 2, Data In** — When set, serially shifts the write data from the 21140A to the serial ROM device.  
  - **Bit 1, Serial ROM Clock** — When set, it provides a serial clock output to the serial ROM.  
  - **Bit 0, Serial ROM Chip Select** — When set, it provides a serial ROM chip select to the serial ROM. |

Table 3-51 lists the access rules for CSR9.
3.2 CSR Operation

Table 3–51 CSR9 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FFF097FFH</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.2.2.10 Boot ROM Programming Address Register (CSR10–Offset 50H)

The boot ROM programming address register (CSR10) contains the 18-bit boot ROM address.

Figure 3–21 shows the CSR10 bit field and Table 3–52 describes the bit field.

Figure 3–21  CSR10 Boot ROM Programming Address Register

Table 3–52 CSR10 Boot ROM Programming Address Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:0</td>
<td>Boot ROM Address</td>
</tr>
<tr>
<td></td>
<td>Contains a pointer to the boot ROM</td>
</tr>
</tbody>
</table>

Table 3–53 lists the access rules for CSR10.

Table 3–53 CSR10 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>UNPREDICTABLE</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>
3.2 CSR Operation

3.2.2.11 General-Purpose Timer Register (CSR11–Offset 58H)

This register contains a 16-bit general-purpose timer. It is used mainly by the software driver for timing functions not supplied by the operating system. After this timer is loaded, it starts counting down. The expiration of the timer causes an interrupt in CSR5<11>. If the timer expires with the CON bit on, the counter will load itself automatically with the last value. The value that is read by the host in this register is the current count value. The timer is not active in snooze mode. The timer is asynchronous and the precision is ±1 bit.

The timer operation is based on the existing serial clock. The cycle time of the timer depends on the port that is selected.

Figure 3–22 shows the CSR11 bit fields and Table 3–54 describes the bit fields.

**Figure 3–22 CSR11 General-Purpose Timer Register**

![CSR11 General-Purpose Timer Register Diagram]

**Table 3–54 CSR11 General-Purpose Timer Register Description**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td><strong>CON — Continuous Mode</strong></td>
</tr>
<tr>
<td></td>
<td>When set, the general-purpose timer is in continuous operating mode. When reset, the general-purpose timer is in one-shot operating mode.</td>
</tr>
<tr>
<td>15:0</td>
<td><strong>Timer Value</strong></td>
</tr>
<tr>
<td></td>
<td>Contains the number of iterations of the general-purpose timer. Each iteration duration is:</td>
</tr>
<tr>
<td></td>
<td>10Mb/s port is 204.8 µs.</td>
</tr>
<tr>
<td></td>
<td>100Mb/s MII is 81.92 µs.</td>
</tr>
<tr>
<td></td>
<td>10Mb/s MII is 819.2 µs.</td>
</tr>
</tbody>
</table>

Table 3–55 lists the access rules for CSR11.
3.2 CSR Operation

Table 3–55  CSR11 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FFFE0000H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>

3.2.2.12 General-Purpose Port Register (CSR12–Offset 60H)

The 21140A has an 8-pin general-purpose port that is controlled by CSR12.

Figure 3–23 shows the CSR12 bit fields and Table 3–56 describes the bit fields.

Figure 3–23  CSR12 General-Purpose Port Register

GPC - General-Purpose Control
MD - General-Purpose Mode and Data

LJ-04706.AI5
### 3.2 CSR Operation

#### Table 3–56 CSR12 General-Purpose Port Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 8     | GPC — General-Purpose Control  
Determines whether accessing CSR12<7:0> affects either the direction of each pin (input or output) or the data of each pin (1 or 0). The interaction of this bit and CSR12<7:0> is described in the following field. |
| 7:0   | MD — General-Purpose Mode and Data  
When CSR12<8> is set, the value that is written by the host to CSR12<7:0> sets the direction of each pin to be either an input pin or an output pin. For example, if CSR12<1> is 1, then gep<1> is an output pin. If CSR12<1> is 0, then gep<1> is an input pin.  
When a hardware reset is initiated, all gep pins become input pins.  
When CSR12<8> is reset, any host write access to CSR12<7:0> sets values on the pins that are configured as output pins. For example, if CSR12<1> is 1 (and is defined as an output pin), then gep<1> is 1. If CSR12<1> is 0 (and is defined as an output pin), then gep<1> is 0.  
Any host read access to CSR12<7:0> reflects the input values on any pins designated as input pins and output values on any pins designated as output pins.  
The application of the general-purpose pins in board design should be correlated with the way the port driver software is using it. |

**Note**  
Refer to the 21140A application notes for the details regarding a particular application.

Table 3–57 lists the access rules for CSR12.

#### Table 3–57 CSR12 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FFFFFFFXXH</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>
3.2 CSR Operation

3.2.2.13 Watchdog Timer Register (CSR15–Offset 78H)

Figure 3–24 shows the CSR15 bit fields and Table 3–58 describes the bit fields. This register is mainly used for diagnostic purposes.

Figure 3–24 CSR15 Watchdog Timer Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>MBZ - Must Be Zero</td>
</tr>
<tr>
<td>9–8</td>
<td>RWR - Receive Watchdog Release</td>
</tr>
<tr>
<td>5–4</td>
<td>RWD - Receive Watchdog Disable</td>
</tr>
<tr>
<td>2–1</td>
<td>JCK - Jabber Clock</td>
</tr>
<tr>
<td>0</td>
<td>HUJ - Host Unjab</td>
</tr>
<tr>
<td></td>
<td>JBD - Jabber Disable</td>
</tr>
</tbody>
</table>

LJ04094A.A05
3.2 CSR Operation

Table 3–58 CSR15 Watchdog Timer Register Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 8     | MBZ — Must Be Zero  
This bit is not used by the 21140A and should always be programmed to zero. |
| 5     | RWR — Receive Watchdog Release  
Defines the time interval no carrier from receive watchdog expiration until reenabling the receive channel. When set, the receive watchdog is released 40- to 48-bit-times from the last carrier deassertion. When reset, the receive watchdog is released 16- to 24-bit-times from the last carrier deassertion. |
| 4     | RWD — Receive Watchdog Disable  
When set, the receive watchdog counter is disabled. When reset, receive carriers longer than 2560 bytes are guaranteed to cause the watchdog counter to time out. Packets shorter than 2048 bytes are guaranteed to pass. |
| 2     | JCK — Jabber Clock  
When set, transmission is cut off after a range of 2048 bytes to 2560 bytes is transmitted.  
When reset, transmission for the 10Mb/s port is cut off after a range of 26 ms to 33 ms.  
When reset, transmission for the 100Mb/s port is cut off after a range of 2.6 ms to 3.3 ms. |
| 1     | HUJ — Host Unjab  
Defines the time interval between transmit jabber expiration until reenabling of the transmit channel. When set, the transmit channel is released immediately after the jabber expiration.  
When reset, the transmit jabber is released 365 ms to 420 ms after jabber expiration for the 10Mb/s port.  
When reset, the transmit jabber is released 36.5 ms to 42 ms after jabber expiration for the 100Mb/s port. |
| 0     | JBD — Jabber Disable  
When set, the transmit jabber function is disabled. |

Table 3–59 lists the access rules for CSR15.
3.2 CSR Operation

Table 3–59  CSR15 Access Rules

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value after reset</td>
<td>FFFFFFFEC8H</td>
</tr>
<tr>
<td>Read access rules</td>
<td>—</td>
</tr>
<tr>
<td>Write access rules</td>
<td>—</td>
</tr>
</tbody>
</table>
This chapter describes descriptor lists and data buffers, which are collectively called the host communication area, that manage the actions and status related to buffer management. Commands and signals that control the functional operation of the 21140A are also described.

4.1 Data Communication

The 21140A and the driver communicate through two data structures:

- Control and status registers (CSRs), described in Chapter 3.
- Descriptor lists and data buffers, described in this chapter.

4.2 Descriptor Lists and Data Buffers

The 21140A transfers received data frames to the receive buffers in host memory and transmits data from the transmit buffers in host memory. Descriptors that reside in the host memory act as pointers to these buffers.

There are two descriptor lists, one for receive and one for transmit. The base address of each list is written into CSR3 and CSR4, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both the receive and transmit descriptors (RDES1<24> and TDES1<24>). The descriptor lists reside in the host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, and not contiguous in memory (Figure 4–1).

A data buffer consists of either an entire frame or part of a frame, but it cannot exceed a single frame. Buffers contain only data; buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. Data chaining can be enabled or disabled. Data buffers reside in host physical memory space.
4.2 Descriptor Lists and Data Buffers

Figure 4-1 Descriptor Ring and Chain Structure Examples

Ring Structure

```
| Descriptor 0 | Buffer 1 |
| Descriptor 1 | Buffer 2 |
| Descriptor n | Buffer 1 |
|              | Buffer 2 |
```

Chain Structure

```
| Descriptor 0 | Buffer 1 |
| Descriptor 1 | Buffer 1 |
| Next Descriptor |
```

4.2.1 Receive Descriptors

Figure 4-2 shows the receive descriptor format.

**Note**

Descriptors and receive buffers addresses must be longword aligned.
4.2 Descriptor Lists and Data Buffers

Providing two buffers, two byte-count buffers, and two address pointers in each descriptor enables the adapter port to be compatible with various types of memory-management schemes.

Figure 4–2 Receive Descriptor Format

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
</tr>
<tr>
<td>OWN</td>
<td>WN</td>
<td>Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDES0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDES1</td>
<td>Control Bits</td>
<td>Byte-Count Buffer 2</td>
<td>Byte-Count Buffer 1</td>
<td></td>
</tr>
<tr>
<td>RDES2</td>
<td></td>
<td>Buffer Address 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDES3</td>
<td></td>
<td>Buffer Address 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2.1.1 Receive Descriptor 0 (RDES0)

RDES0 contains the received frame status, the frame length, and the descriptor ownership information. Figure 4–3 shows the RDES0 bit fields and Table 4–1 describes the bit fields.
4.2 Descriptor Lists and Data Buffers

Figure 4–3  RDES0 Receive Descriptor 0

OWN - Own Bit
FF - Filtering Fail
FL - Frame Length
ES - Error Summary
DE - Descriptor Error
DT - Data Type
RF - Runt Frame
MF - Multicast Frame
FS - First Descriptor
LS - Last Descriptor
TL - Frame Too Long
CS - Collision Seen
FT - Frame Type
RW - Receive Watchdog
RE - Report on MII Error
DB - Dribbling Bit
CE - CRC Error
ZER - Zero
4.2 Descriptor Lists and Data Buffers

Table 4–1  RDES0 Receive Descriptor 0 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 31    | OWN — Own Bit  
When set, indicates that the descriptor is owned by the 21140A. When reset, indicates that the descriptor is owned by the host. The 21140A clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full. |
| 30    | FF — Filtering Fail  
When set, this indicates that the frame failed the address recognition filtering. This bit can be set only when receive all (CSR6<30>) is set. Otherwise, this bit is reset. |
| 29:16 | FL — Frame Length  
Indicates the length, in bytes, of the received frame, including the cyclic redundancy check (CRC).  
This field is valid only when last descriptor (RDES0<8>) is set and descriptor error (RDES0<14>) is reset. |
| 15    | ES — Error Summary  
Indicates the logical OR of the following RDES0 bits:  
RDES0<1> — CRC error  
RDES0<6> — Collision seen  
RDES0<7> — Frame too long  
RDES0<11> — Runt frame  
RDES0<14> — Descriptor error  
This bit is valid only when last descriptor (RDES0<8>) is set. |
| 14    | DE — Descriptor Error  
When set, indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the 21140A does not own the next descriptor. The frame is truncated.  
This bit is valid only when last descriptor (RDES0<8>) is set. |

(continued on next page)
### 4.2 Descriptor Lists and Data Buffers

#### Table 4–1 (Cont.) RDES0 Receive Descriptor 0 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>13:12 DT — Data Type</strong></td>
<td>Indicates the type of frame the buffer contains:</td>
</tr>
<tr>
<td>00</td>
<td>Serial received frame.</td>
</tr>
<tr>
<td>01</td>
<td>Internal loopback frame.</td>
</tr>
<tr>
<td>10</td>
<td>External loopback frame or serial received frame. The 21140A does not differentiate between loopback and serial received frames; therefore, this information is global and reflects only the operating mode (CSR6&lt;11:10&gt;).</td>
</tr>
<tr>
<td>11</td>
<td>Reserved. This field is valid only when last descriptor (RDES0&lt;8&gt;) is set.</td>
</tr>
<tr>
<td><strong>11 RF — Runt Frame</strong></td>
<td>When set, indicates that this frame was damaged by a collision or premature termination before the collision window had passed. Runt frames are passed on to the host only if the pass bad frames bit (CSR6&lt;3&gt;) is set. This bit is valid only when last descriptor (RDES0&lt;8&gt;) is set and overflow (RDES0&lt;0&gt;) is reset.</td>
</tr>
<tr>
<td><strong>10 MF — Multicast Frame</strong></td>
<td>When set, indicates that this frame has a multicast address. This bit is valid only when last descriptor (RDES0&lt;8&gt;) is set.</td>
</tr>
<tr>
<td><strong>9 FS — First Descriptor</strong></td>
<td>When set, indicates that this descriptor contains the first buffer of a frame. If the buffer size of the first buffer is 0, the second buffer contains the beginning of the frame. If the buffer size of the second buffer is also 0, the second descriptor contains the beginning of the frame.</td>
</tr>
<tr>
<td><strong>8 LS — Last Descriptor</strong></td>
<td>When set, indicates that the buffers pointed to by this descriptor are the last buffers of the frame.</td>
</tr>
<tr>
<td><strong>7 TL — Frame Too Long</strong></td>
<td>When set, indicates that the frame length exceeds the maximum Ethernet-specified size of 1518 bytes. This bit is valid only when last descriptor (RDES0&lt;8&gt;) is set. (continued on next page)</td>
</tr>
</tbody>
</table>
Table 4–1 (Cont.)  RDES0 Receive Descriptor 0 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

_________________________ Note ___________________________

Frame too long is only a frame length indication and does not cause any frame truncation.

_________________________

6  CS — Collision Seen
When set, indicates that the frame was damaged by a collision that occurred after the 64 bytes following the start frame delimiter (SFD). This is a late collision.
This bit is valid only when last descriptor (RDES0<8>) is set.

5  FT — Frame Type
When set, indicates that the frame is an Ethernet-type frame (frame length field is greater than 1500 bytes). When clear, indicates that the frame is an IEEE 802.3 frame.
This bit is not valid for runt frames of less than 14 bytes.
This bit is valid only when last descriptor (RDES0<8>) is set.

4  RW — Receive Watchdog
When set, indicates that the receive watchdog timer expired while receiving the current packet with length greater than 2048 bytes through 2560 bytes. Receive watchdog timeout (CSR5<9>) is set.
When RDES0<4> is set, the frame length field in RDES0<30:16> is not valid.
This bit is valid only when last descriptor (RDES0<8>) is set.

3  RE — Report on MII Error
When set, indicates that a receive error in the physical layer was reported during the frame reception.

2  DB — Dribbling Bit
When set, indicates that the frame contained a noninteger multiple of 8 bits.
This error is reported only if the number of dribbling bits in the last byte is 4 in MII/SYM operating mode, or at least 3 in 10Mb/s serial operating mode. This bit is not valid if either collision seen (RDES0<6>) or runt frame (RDES0<11>) is set. If set, and the CRC error (RDES0<1>) is reset, then the packet is valid.
This bit is valid only when last descriptor (RDES0<8>) is set.

(continued on next page)
4.2 Descriptor Lists and Data Buffers

Table 4–1 (Cont.) RDES0 Receive Descriptor 0 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1     | CE — CRC Error  
When set, indicates that a cyclic redundancy check (CRC) error occurred on the received frame. This bit is also set when the mii_err pin is asserted during the reception of a receive packet even though the CRC may be correct. This bit is valid only when last descriptor (RDES0<8>) is set. |
| 0     | ZER — Zero  
This bit is always zero for a packet with a legal length. |

4.2.1.2 Receive Descriptor 1 (RDES1)

Figure 4–4 shows the RDES1 bit fields and Table 4–2 describes the bit fields.

Figure 4–4 RDES1 Receive Descriptor 1

| Bit 31 | Bit 30 | Bit 29 | Bit 28 | Bit 25 | Bit 24 | Bit 23 | Bit 22 | Bit 21 | Bit 20 | Bit 19 | Bit 18 | Bit 17 | Bit 16 | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9  | Bit 8  | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| RER - Receive End of Ring |
| RCH - Second Address Chained |
| RBS2 - Buffer 2 Size |
| RBS1 - Buffer 1 Size |
4.2 Descriptor Lists and Data Buffers

Table 4–2 RDES1 Receive Descriptor 1 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 25    | **RER — Receive End of Ring**  
When set, indicates that the descriptor list reached its final descriptor.  
The 21140A returns to the base address of the list (Section 3.2.2.4),  
creating a descriptor ring. |
| 24    | **RCH — Second Address Chained**  
When set, indicates that the second address in the descriptor is the next  
descrptor address, rather than the second buffer address.  
RDES1<25> takes precedence over RDES1<24>. |
| 21:11 | **RBS2 — Buffer 2 Size**  
Indicates the size, in bytes, of the second data buffer. If this field is 0, the  
21140A ignores this buffer and fetches the next descriptor.  
The buffer size must be a multiple of 4.  
This field is not valid if RDES1<24> is set. |
| 10:0  | **RBS1 — Buffer 1 Size**  
Indicates the size, in bytes, of the first data buffer. If this field is 0, the  
21140A ignores this buffer and uses buffer 2.  
The buffer size must be a multiple of 4. |

4.2.1.3 Receive Descriptor 2 (RDES2)

Figure 4–5 shows the RDES2 bit field and Table 4–3 describes the bit field.

Figure 4–5 RDES2 Receive Descriptor 2

Table 4–3 RDES2 Receive Descriptor 2 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 31:0  | **Buffer Address 1**  
Indicates the physical address of buffer 1. The buffer must be longword  
aligned (RDES2<1:0> = 00). |
4.2 Descriptor Lists and Data Buffers

4.2.1.4 Receive Descriptor 3 (RDES3)

Figure 4–6 shows the RDES3 bit field and Table 4–4 describes the bit field.

**Figure 4–6 RDES3 Receive Descriptor 3**

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>25</th>
<th>24</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 4–4 RDES3 Receive Descriptor 3 Description**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:0</td>
<td><strong>Buffer Address 2</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates the physical address of buffer 2. The buffer must be longword aligned (RDES3&lt;1:0&gt; = 00).</td>
</tr>
</tbody>
</table>

4.2.1.5 Receive Descriptor Status Validity

Table 4–5 lists the validity of the receive descriptor status bits in relation to the reception completion status.
4.2 Descriptor Lists and Data Buffers

Table 4–5 Receive Descriptor Status Validity

<table>
<thead>
<tr>
<th>Reception Status</th>
<th>RF</th>
<th>CS</th>
<th>FT</th>
<th>FF</th>
<th>DB</th>
<th>CE</th>
<th>RE</th>
<th>(ES, DE, DT, FS, LS, FL, OF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overflow</td>
<td>0</td>
<td>0</td>
<td>V</td>
<td>V</td>
<td>NV</td>
<td>NV</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Collision after 512 bits</td>
<td>0</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Runt frame</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Runt frame less than 14 bytes</td>
<td>V</td>
<td>V</td>
<td>NV</td>
<td>NV</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Watchdog timeout</td>
<td>0</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>NV</td>
<td>NV</td>
<td>V</td>
<td>V</td>
</tr>
</tbody>
</table>

List of table abbreviations

- RF — Runt frame (RDES0<11>)
- CS — Collision seen (RDES0<6>)
- FT — Frame type (RDES0<5>)
- FF — Filtering Fail (RDES0<3>)
- DB — Dribbling bit (RDES0<2>)
- CE — CRC error (RDES0<1>)
- RE — Report on MII Error (RDES0<1>)
- ES — Error summary (RDES0<15>)
- DE — Descriptor error (RDES0<14>)
- DT — Data type (RDES0<13:12>)
- FS — First descriptor (RDES0<9>)
- LS — Last descriptor (RDES0<8>)
- FL — Frame length (RDES0<30:16>)
- OF — Overflow (RDES0<0>)
- V — Valid
- NV — Not valid

4.2.2 Transmit Descriptors

Figure 4–7 shows the Transmit descriptor format.

---

**Note**

Descriptor addresses must be longword aligned.

---

Providing two buffers, two byte-count buffers, and two address pointers in each descriptor enables the adapter port to be compatible with various types of memory-management schemes.
4.2 Descriptor Lists and Data Buffers

4.2.2.1 Transmit Descriptor 0 (TDES0)

TDES0 contains transmitted frame status and descriptor ownership information. Figure 4-8 shows the TDES0 bit fields and Table 4-6 describes the bit fields.

Figure 4–8 TDES0 Transmit Descriptor 0
4.2 Descriptor Lists and Data Buffers

Table 4–6  TDES0 Transmit Descriptor 0 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>OWN — Own Bit</td>
</tr>
<tr>
<td></td>
<td>When set, indicates that the descriptor is owned by the 21140A. When cleared, indicates that the descriptor is owned by the host. The 21140A clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the first descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between the 21140A fetching a descriptor and the driver setting an ownership bit.</td>
</tr>
<tr>
<td>15</td>
<td>ES — Error Summary</td>
</tr>
<tr>
<td></td>
<td>Indicates the logical OR of the following bits:</td>
</tr>
<tr>
<td></td>
<td>TDES0&lt;1&gt; — Underflow error</td>
</tr>
<tr>
<td></td>
<td>TDES0&lt;8&gt; — Excessive collisions</td>
</tr>
<tr>
<td></td>
<td>TDES0&lt;9&gt; — Late collision</td>
</tr>
<tr>
<td></td>
<td>TDES0&lt;10&gt; — No carrier</td>
</tr>
<tr>
<td></td>
<td>TDES0&lt;11&gt; — Loss of carrier</td>
</tr>
<tr>
<td></td>
<td>TDES0&lt;14&gt; — Transmit jabber timeout</td>
</tr>
<tr>
<td>14</td>
<td>TO — Transmit Jabber Timeout</td>
</tr>
<tr>
<td></td>
<td>When set, indicates that the transmit jabber timer timed out and that the 21140A transmitter was still active. The transmit jabber timeout interrupt CSR5&lt;3&gt; is set. The transmission process is aborted and placed in the STOPPED state. When TDES0&lt;14&gt; is set, any heartbeat fail indication (TDES0&lt;7&gt;) is not valid.</td>
</tr>
<tr>
<td>11</td>
<td>LO — Loss of Carrier</td>
</tr>
<tr>
<td></td>
<td>When set, indicates loss of carrier during transmission. Not valid in internal loopback mode (CSR6&lt;11:10&gt;=01).</td>
</tr>
<tr>
<td>10</td>
<td>NC — No Carrier</td>
</tr>
<tr>
<td></td>
<td>When set, indicates that the carrier signal from the transceiver was not present during transmission. Not valid in internal loopback mode (CSR6&lt;11:10&gt;=01).</td>
</tr>
<tr>
<td>9</td>
<td>LC — Late Collision</td>
</tr>
<tr>
<td></td>
<td>When set, indicates that the frame transmission was aborted due to collision occurring after the collision window of 64 bytes. Not valid if underflow error (TDES0&lt;1&gt;) is set.</td>
</tr>
</tbody>
</table>

(continued on next page)
4.2 Descriptor Lists and Data Buffers

Table 4–6 (Cont.) TDES0 Transmit Descriptor 0 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td><strong>EC — Excessive Collisions</strong>&lt;br&gt;When set, indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame.</td>
</tr>
<tr>
<td>7</td>
<td><strong>HF — Heartbeat Fail</strong>&lt;br&gt;This bit is effective only in 10Mb/s operating mode. When set, this bit indicates a heartbeat collision check failure (the transceiver failed to return a collision pulse as a check after the transmission). For transceivers that do not support heartbeat collision check, heartbeart fail is set but is not valid.&lt;br&gt;This bit is not valid if underflow error (TDES0&lt;1&gt;) is set.&lt;br&gt;On the second transmission attempt, after the first transmission was aborted due to a collision, the 21140A does not check heartbeat fail (TDES0&lt;7&gt;) and is reset.</td>
</tr>
<tr>
<td>6:3</td>
<td><strong>CC — Collision Count</strong>&lt;br&gt;This 4-bit counter indicates the number of collisions that occurred before the frame was transmitted.&lt;br&gt;Not valid when the excessive collisions bit (TDES0&lt;8&gt;) is also set.</td>
</tr>
<tr>
<td>2</td>
<td><strong>LF — Link Fail Report</strong>&lt;br&gt;When set, indicates that the link test failed before the frame was transmitted through the symbol port. This bit is valid only while using the symbol mode (CSR6&lt;23&gt; is set.)</td>
</tr>
<tr>
<td>1</td>
<td><strong>UF — Underflow Error</strong>&lt;br&gt;When set, indicates that the transmitter aborted the message because data arrived late from memory. Underflow error indicates that the 21140A encountered an empty transmit FIFO while transmitting a frame. The transmission process enters the suspended state and sets both transmit underflow (CSR5&lt;5&gt;) and transmit interrupt (CSR5&lt;0&gt;).</td>
</tr>
<tr>
<td>0</td>
<td><strong>DE — Deferred</strong>&lt;br&gt;When set, indicates that the 21140A had to defer while ready to transmit a frame because the carrier was asserted.</td>
</tr>
</tbody>
</table>

4.2.2.2 Transmit Descriptor 1 (TDES1)<br>Figure 4–9 shows the TDES1 bit fields and Table 4–7 describes the bit fields.
4.2 Descriptor Lists and Data Buffers

Figure 4–9  TDES1 Transmit Descriptor 1

Table 4–7  TDES1 Transmit Descriptor 1 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>IC — Interrupt on Completion</td>
</tr>
<tr>
<td></td>
<td>When set, the 21140A sets transmit interrupt (CSR5&lt;0&gt;) after the present frame has been transmitted. It is valid only when last segment (TDES1&lt;30&gt;) is set or when it is a setup packet.</td>
</tr>
<tr>
<td>30</td>
<td>LS — Last Segment</td>
</tr>
<tr>
<td></td>
<td>When set, indicates that the buffer contains the last segment of a frame.</td>
</tr>
<tr>
<td>29</td>
<td>FS — First Segment</td>
</tr>
<tr>
<td></td>
<td>When set, indicates that the buffer contains the first segment of a frame.</td>
</tr>
<tr>
<td>28</td>
<td>FT1 — Filtering Type</td>
</tr>
<tr>
<td></td>
<td>Table 4–8 lists the filtering types.</td>
</tr>
</tbody>
</table>

(continued on next page)
4.2 Descriptor Lists and Data Buffers

Table 4–7 (Cont.)  TDES1 Transmit Descriptor 1 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 27    | SET — Setup Packet  
When set, indicates that the current descriptor is a setup frame descriptor (Section 4.2.3). |
| 26    | AC — Add CRC Disable  
When set, the 21140A does not append the cyclic redundancy check (CRC) to the end of the transmitted frame. This field is valid only when first segment (TDES1<29>) is set. |
| 25    | TER — Transmit End of Ring  
When set, indicates that the descriptor pointer has reached its final descriptor. The 21140A returns to the root address of the list (Section 3.2.2.4). This creates a descriptor ring. |
| 24    | TCH — Second Address Chained  
When set, indicates that the second address in the descriptor is the next descriptor address, rather than the second buffer address.  
Transmit end of ring (TDES1<25>) takes precedence over second address chained (TDES1<24>). |
| 23    | DPD — Disabled Padding  
When set, the 21140A does not automatically add a padding field, to a packet shorter than 64 bytes.  
When reset, the 21140A automatically adds a padding field and also a CRC field to a packet shorter than 64 bytes. The CRC field is added despite the state of the add CRC disable (TDES1<26>) flag. |
| 22    | FT0 — Filtering Type  
Table 4–8 lists the filtering types. |
| 21:11 | TBS2 — Buffer 2 Size  
Indicates the size, in bytes, of the second data buffer. If this field is 0, the 21140A ignores this buffer and fetches the next descriptor.  
This field is not valid if second address chained (TDES1<24>) is set. |
| 10:0  | TBS1 — Buffer 1 Size  
Indicates the size, in bytes, of the first data buffer. If this field is 0, the 21140A ignores this buffer and uses buffer 2. |

Table 4–8 lists the filtering types. Section 3.2.2.6 provides additional information on filtering.
4.2 Descriptor Lists and Data Buffers

Table 4–8 Filtering Type

<table>
<thead>
<tr>
<th>FT1</th>
<th>FT0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td><strong>Perfect Filtering</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 21140A interprets the descriptor buffer as a setup perfect table of 16 addresses, and sets the 21140A filtering mode to perfect filtering. This field is valid only when setup packet (TDES1&lt;27&gt;) is set (Table 3–44).</td>
</tr>
<tr>
<td>01</td>
<td></td>
<td><strong>Hash Filtering</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 21140A interprets the descriptor buffer as a setup hash table of 512-bit-plus-one perfect address. If an incoming receive packet destination address is a multicast address, the 21140A executes an imperfect address filtering compared with the hash table. However, if the incoming receive packet destination address is a physical address, the 21140A executes a perfect filtering compared with the perfect address. This field is valid only when setup packet (TDES1&lt;27&gt;) is set (Table 3–44).</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td><strong>Inverse Filtering</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 21140A interprets the descriptor buffer as a setup perfect table of 16 addresses and sets the 21140A filtering mode to inverse filtering. The 21140A receives the incoming frames with destination addresses not matching the perfect addresses and rejects the frames with destination addresses matching one of the perfect addresses. This field is valid only when setup packet (TDES1&lt;27&gt;) is set (Table 3–44).</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Hash-Only Filtering</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 21140A interprets the descriptor buffer as a setup 512-bit hash table. If an incoming receive packet destination address is multicast or physical, the 21140A executes an imperfect address filtering against the hash table. This field is valid only when setup packet (TDES1&lt;27&gt;) is set (Table 3–44).</td>
</tr>
</tbody>
</table>

4.2.2.3 Transmit Descriptor 2 (TDES2)

Figure 4–10 shows the TDES2 bit field and Table 4–9 describes the bit field.

Figure 4–10  TDES2 Transmit Descriptor 2
4.2 Descriptor Lists and Data Buffers

Table 4–9  TDES2 Transmit Descriptor 2 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:0</td>
<td>Buffer Address 1</td>
</tr>
<tr>
<td></td>
<td>Physical address of buffer 1. There are no limitations on the buffer address alignment.</td>
</tr>
</tbody>
</table>

4.2.2.4 Transmit Descriptor 3 (TDES3)

Figure 4–11 shows the TDES3 bit field and Table 4–10 describes the bit field.

Figure 4–11  TDES3 Transmit Descriptor 3

```
 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
```

Table 4–10  TDES3 Transmit Descriptor 3 Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:0</td>
<td>Buffer Address 2</td>
</tr>
<tr>
<td></td>
<td>Physical address of buffer 2. There are no limitations on the buffer address alignment.</td>
</tr>
</tbody>
</table>

4.2.2.5 Transmit Descriptor Status Validity

Table 4–11 lists the validity of the transmit descriptor status bits during transmission completion status.
4.2 Descriptor Lists and Data Buffers

Table 4–11 Transmit Descriptor Status Validity

<table>
<thead>
<tr>
<th>Transmission Status</th>
<th>LO</th>
<th>NC</th>
<th>LC</th>
<th>EC</th>
<th>HF</th>
<th>CC</th>
<th>(ES, TO, UF, DE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Underflow</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Excessive collisions</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>NV</td>
<td>V</td>
</tr>
<tr>
<td>Watchdog timeout</td>
<td>NV</td>
<td>V</td>
<td>NV</td>
<td>NV</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Internal loopback</td>
<td>NV</td>
<td>NV</td>
<td>V</td>
<td>V</td>
<td>NV</td>
<td>V</td>
<td>V</td>
</tr>
</tbody>
</table>

List of table abbreviations

LO — Loss of carrier (TDES0<11>)
NC — No carrier (TDES0<10>)
LC — Late collision (TDES0<9>)
EC — Excessive collisions (TDES0<8>)
HF — Heartbeat fail (TDES0<7>)
CC — Collision count (TDES0<6:3>)
ES — Error summary (TDES0<15>)
TO — Transmit jabber timeout (TDES0<14>)
UF — Underflow error (TDES0<1>)
DE — Deferred (TDES0<0>)
V — Valid
NV — Not valid

4.2.3 Setup Frame

A setup frame defines the 21140A Ethernet addresses that are used to filter all incoming frames. The setup frame is never transmitted on the Ethernet wire nor is it looped back to the receive list. When processing the setup frame, the receiver logic temporarily disengages from the Ethernet wire. The setup frame size must be exactly 192 bytes.

Note

The setup frame must be allocated in a single buffer that is longword aligned. First segment (TDES1<29>) and last segment (TDES1<30>) must both be 0.

When the setup frame load is completed, the 21140A closes the setup frame descriptor by clearing its ownership bit and setting all other bits to 1.
4.2 Descriptor Lists and Data Buffers

4.2.3.1 First Setup Frame
A setup frame must be processed before the reception process is started, except when it operates in promiscuous filtering mode.

4.2.3.2 Subsequent Setup Frames
Subsequent setup frames may be queued to the 21140A despite the reception process state. To ensure correct setup frame processing, these packets may be queued at the beginning of the transmit descriptor’s ring or following a descriptor with a zero-length buffer. For the descriptor with a zero-length buffer, it should contain the following information:

- $TDES0<31> = 1$ (Adapter-owned descriptor)
- $TDES1<30> = 0$ (Last segment bit 0)
- $TDES1<29> = 0$ (First segment bit 0)
- $TDES1<21:11> = 0$ (Transmit buffer 2 empty)
- $TDES1<10:0> = 0$ (Transmit buffer 1 empty)

Setup packet ($TDES1<27>$) may also be set. If so, the address filtering bits ($TDES1<22>$ and $TDES1<28>$) should be the same as in the previous packet.

For setup frame processing, the transmission process must be running. The setup frame is processed after all preceding frames have been transmitted and the current frame reception, if any, is completed.

The setup frame does not affect the reception process state, but during setup frame processing, the 21140A is disengaged from the Ethernet wire.

4.2.3.3 Perfect Filtering Setup Frame Buffer
This section describes how the 21140A interprets a setup frame buffer in perfect filtering mode ($CSR6<0> = 0$).

The 21140A can store 16 destination addresses (full 48-bit Ethernet addresses). The 21140A compares the addresses of any incoming frame to these addresses, and also tests the status of the inverse filtering ($CSR6<4>$). It rejects addresses that:

- Do not match if not inverse filtering ($CSR6<4> = 0$).
- Match if inverse filtering ($CSR6<4> = 1$).

The setup frame must always supply all 16 addresses. Any mix of physical and multicast addresses can be used. Unused addresses should duplicate one of the valid addresses.
4.2 Descriptor Lists and Data Buffers

Figure 4–12 shows the perfect filtering setup frame buffer format of the addresses.

**Figure 4–12  Perfect Filtering Setup Frame Buffer Format**

<table>
<thead>
<tr>
<th>31</th>
<th>16</th>
<th>15</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;3:0&gt;</td>
<td>Physical Address 00 (Bytes &lt;1:0&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;7:4&gt;</td>
<td>Physical Address 00 (Bytes &lt;3:2&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;11:8&gt;</td>
<td>Physical Address 00 (Bytes &lt;5:4&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 03</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Address 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;183:180&gt;</td>
<td>Physical Address 15 (Bytes &lt;1:0&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;187:184&gt;</td>
<td>Physical Address 15 (Bytes &lt;3:2&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;191:188&gt;</td>
<td>Physical Address 15 (Bytes &lt;5:4&gt;)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

XXXXXX = Don’t care

The low-order bit of the low-order bytes is the multicast bit of the address.
4.2 Descriptor Lists and Data Buffers

Example 4–1 shows a perfect filtering setup buffer (fragment).

**Example 4–1  Perfect Filtering Buffer**

Ethernet addresses to be filtered:

1. A8-09-65-12-34-76
2. 09-BC-87-DE-03-15

Setup frame buffer fragment while in little endian byte ordering:

2. xxxx09A8
   xxxx1265
   xxxx7634
   xxxxBC09
   xxxxDE87
   xxxx1503

Setup frame buffer fragment while in big endian byte ordering:

3. A809xxxx
   6512xxxx
   3476xxxx
   098Cxxxx
   870Exxxx
   0315xxxx

1. Displays two Ethernet addresses written according to the Ethernet specification for address display.

2. Displays two addresses as they would appear in the buffer in little endian format.

3. Displays two addresses as they would appear in the buffer in big endian format.

4.2.3.4 Imperfect Filtering Setup Frame Buffer

This section describes how the 21140A interprets a setup frame buffer in imperfect filtering mode (CSR6<0> is set). Figure 4–13 shows imperfect filtering.
The 21140A can store 512 bits serving as hash bucket heads, and one physical 48-bit Ethernet address. Incoming frames with multicast destination addresses are subjected to imperfect filtering. Frames with physical destination addresses are checked against the single physical address.

For any incoming frame with a multicast destination address, the 21140A applies the standard Ethernet cyclic redundancy check (CRC) function to the first 6 bytes containing the destination address, then it uses the most significant 9 bits of the result as a bit index into the table. If the indexed bit is set, the frame is accepted. If the bit is cleared, the frame is rejected. (Appendix C provides an example of a hash index for a given Ethernet address.)

This filtering mode is called imperfect because multicast frames not addressed to this station may slip through, but it still decreases the number of frames that the host can receive.

Figure 4–14 shows the format for the hash table and the physical address.
### 4.2 Descriptor Lists and Data Buffers

#### Figure 4–14  Imperfect Filtering Setup Frame Format

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;3:0&gt;</td>
<td>Hash Filter (Bytes &lt;1:0&gt;)</td>
</tr>
<tr>
<td>&lt;7:4&gt;</td>
<td>Hash Filter (Bytes &lt;3:2&gt;)</td>
</tr>
<tr>
<td>&lt;127:124&gt;</td>
<td>Hash Filter (Bytes &lt;61:60&gt;)</td>
</tr>
<tr>
<td>&lt;131:128&gt;</td>
<td>Hash Filter (Bytes &lt;63:62&gt;)</td>
</tr>
<tr>
<td>&lt;159:156&gt;</td>
<td>Physical Address</td>
</tr>
<tr>
<td>&lt;163:160&gt;</td>
<td>Physical Address</td>
</tr>
<tr>
<td>&lt;167:164&gt;</td>
<td>Physical Address</td>
</tr>
<tr>
<td>&lt;171:168&gt;</td>
<td>Physical Address</td>
</tr>
<tr>
<td>&lt;191:188&gt;</td>
<td>Physical Address</td>
</tr>
</tbody>
</table>

`XXXXXX = Don't care`

Bits are sequentially numbered from right to left and down the hash table. For example, if the CRC (destination address) <8:0> = 33, the 21140A examines bit 1 in the fourth longword.
4.2 Descriptor Lists and Data Buffers

Example 4–2 shows an imperfect filtering setup frame buffer.

**Example 4–2  Imperfect Filtering Buffer**

Ethernet addresses to be filtered:

1. 25-00-25-00-27-00
   A3-C5-62-3F-25-87
   D9-C2-C0-99-0B-82
   7D-48-4D-FD-CC-0A
   E7-C1-96-36-89-DD
   61-CC-28-55-03-C7
   6B-46-0A-55-2D-7E

2. A8-12-34-35-76-08

Setup frame buffer while in little endian byte ordering:

3. xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx1000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx0000
   xxxx4000
   xxxx0080
   xxxx0000
   xxxx0000
   xxxx0010

(continued on next page)
4.2 Descriptor Lists and Data Buffers

Example 4–2 (Cont.) Imperfect Filtering Buffer

xxxx0000
xxxx0000
xxxx0000
xxxx1000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0001
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx0000
xxxx12A8
xxxx3534
xxxx0876
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxx (continued on next page)
Example 4–2 (Cont.) Imperfect Filtering Buffer

Setup frame buffer while in big endian byte ordering:

```
  0000xxxx
  0000xxxx
  0000xxxx
  0010xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0100xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  0000xxxx
  4000xxxx
  xxxxxxxxxx
  xxxxxxxxxx
  xxxxxxxxxx
  xxxxxxxxxx
  xxxxxxxxxx
  xxxxxxxxxx
  xxxxxxxxxx
  xxxxxxxxxx

(continued on next page)```
4.2 Descriptor Lists and Data Buffers

Example 4–2 (Cont.) Imperfect Filtering Buffer

1. A812xxxx
   3435xxxx
   7608xxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   xxxxxxxx
   Displays Ethernet multicast addresses written according to the Ethernet specification for address display.

2. Displays an Ethernet physical address.

3. Displays the first part of an imperfect filter setup frame buffer, in little endian byte ordering, with set bits for the multicast addresses as in 1.

4. Displays the second part of the buffer with the physical address as in 2, in little endian byte ordering.

5. Displays the first part of an imperfect filter setup frame buffer, in big endian byte ordering, with set bits for the multicast addresses as in 1.

6. Displays the second part of the buffer with the physical address as in 2, in big endian byte ordering.

4.3 Functional Description

This section describes the reset commands, interrupt handling, and startup. It also describes the transmit and receive processes.

The functional operation of the 21140A is controlled by the driver interface located in the host communication area. The driver interface activity is controlled by control and status registers (CSRs), descriptor lists, and data buffers.

Descriptor lists and data buffers, collectively referred to as the host communication area, reside in host memory. These data structures process the actions and status related to buffer management. The 21140A transfers frame data to and from the receive and transmit buffers in host memory. Descriptors resident in the host memory point to these buffers.
4.3 Functional Description

4.3.1 Reset Commands

The following two commands are available to reset the 21140A hardware and software:

- Assert `rst_l`, to initiate a hardware reset.
- Assert CSR0<0>, to initiate a software reset.

For a proper reset operation, both clocks (`pci_clk`, and, depending on the operation mode, either `mii/sym_tclk` or `srl_tclk`) should operate normally. For both the hardware and software reset commands, the 21140A aborts all processing and starts the reset sequence. The 21140A initializes all internal states and registers.

__________________________ Note __________________________

No internal states are retained, no descriptors are owned, and all the host-visible registers are set to the reset values. However, a software reset command has no effect on the configuration registers.

The 21140A does not explicitly disown any owned descriptor; descriptor-owned bits can be left in a state indicating 21140A ownership. Section 4.2.1.1 and Section 4.2.2.1 provide a detailed description of own bits.

__________________________ __________________________

After either a hardware or software reset command, the first bus transaction to the 21140A should not be initiated for at least 50 PCI clock cycles. When the reset sequence completes, the 21140A can accept host commands. The receive and transmit processes are placed in the stopped state (Table 4–13 and Table 4–14). It is permissible to issue successive reset commands (hardware or software).

4.3.2 Arbitration Scheme

The arbitration scheme is used by the 21140A to grant precedence to the receive process instead of the transmit process (CSR0<1>). Table 4–12 lists a description of the arbitration scheme. The technical expressions used in this table are described in the following list:

- `Txreq` — Specifies a DMA request for the transmit process to:
  - Fetch descriptor.
  - Close descriptor.
  - Process setup packet.
4.3 Functional Description

- Allocate sufficient space in the transmit FIFO for a full data burst.
- Rxreq — Specifies a DMA request for the receive process to:
  - Fetch descriptor.
  - Close descriptor.
  - Enter sufficient data into the receive FIFO for a full data burst, or indicate the end of receive packet and the data in the receive FIFO is less than a full burst.
- TxEN — Specifies 21140A is currently transmitting.
- RxF>thr — Specifies the amount of free bytes left in the receive FIFO. The values are taken from the programmed threshold values in CSR6<15:14>. Table 3-41 lists the coding for the programmed values.
- TxF>thtx — Specifies the amount of free bytes left in the transmit FIFO. The values are taken from the programmed threshold values in Table 3-41.

<table>
<thead>
<tr>
<th>Txreq</th>
<th>Rxreq</th>
<th>TxEN</th>
<th>RxF&gt;thr</th>
<th>TxF&gt;thtx</th>
<th>Chosen Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>Receive process</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>Receive process</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>Transmit process</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>Transmit process</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>—</td>
<td>—</td>
<td>Receive process</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Transmit process</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Transmit process</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Transmit process</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Receive process</td>
</tr>
</tbody>
</table>

In addition to the arbitration scheme listed in Table 4-12, two other factors must be considered:

- The transmit process obtains a window for one burst between two consecutive receive packets.
4.3 Functional Description

- The receive process obtains a window for one burst between two consecutive transmit packets.

4.3.3 Interrupts

Interrupts can be generated as a result of various events. CSR5 contains all the status bits that might cause an interrupt. The following list contains the events that cause interrupts:

- CSR5<0> — Transmit interrupt
- CSR5<1> — Transmit process stopped
- CSR5<2> — Transmit buffer unavailable
- CSR5<3> — Transmit jabber timeout
- CSR5<5> — Transmit underflow
- CSR5<6> — Receive interrupt
- CSR5<7> — Receive buffer unavailable
- CSR5<8> — Receive process stopped
- CSR5<9> — Receive watchdog timeout
- CSR5<10> — Early transmit interrupt
- CSR5<11> — General-purpose timer expired
- CSR5<13> — Fatal bus error

Interrupt bits are cleared by writing a 1 to the bit position. This enables additional interrupts from the same source.

Interrupts are not queued, and if the interrupting event recurs before the driver has responded to it, no additional interrupts are generated. For example, receive interrupt (CSR5<6>) indicates that one or more received frames were delivered to host memory. The driver must scan all descriptors, from the last recorded position to the first one owned by the 21140A.

An interrupt is generated only once for simultaneous, multiple interrupting events. The driver must scan CSR5 for the interrupt cause or causes. The interrupt is not generated again, unless a new interrupting event occurs after the driver has cleared the appropriate CSR5 bits.

For example, transmit interrupt (CSR5<0>) and receive interrupt (CSR5<6>) are set simultaneously. The host acknowledges the interrupt, and the driver begins executing by reading CSR5. Next, receive buffer unavailable (CSR5<7>) is set. The driver writes back its copy of CSR5, clearing transmit interrupt and receive interrupt. The interrupt line is deasserted for one cycle and then asserted again with receive buffer unavailable.
4.3 Functional Description

4.3.4 Startup Procedure

The following sequence of checks and commands must be performed by the driver to prepare the 21140A for operation:

1. Wait 50 PCI clock cycles for the 21140A to complete its reset sequence.

2. Update configuration registers (Section 3.1):
   a. Read the configuration ID and revision registers to identify the 21140A and its revision.
   b. Write the configuration interrupt register (if interrupt mapping is necessary).
   c. Write the configuration base address registers to map the 21140A I/O or memory address space into the appropriate processor address space.
   d. Write the configuration command register.
   e. Write the configuration latency counter to match the system latency guidelines.

3. Write CSR0 to set global host bus operating parameters (Section 3.2.2.1).

4. Write CSR7 to mask unnecessary (depending on the particular application) interrupt causes.

5. The driver must create the transmit and receive descriptor lists. Then, it writes to both CSR3 and CSR4, providing the 21140A with the starting address of each list (Section 3.2.2.4). The first descriptor on the transmit list may contain a setup frame (Section 4.2.3).

   __________________________ Caution __________________________
   If address filtering (either perfect or imperfect) is desired, the receive process should only be started after the setup frame has been processed (Section 4.2.3).

6. Write CSR6 (Section 3.2.2.6) to set global serial parameters and start both the receive and transmit processes. The receive and transmit processes enter the running state and attempt to acquire descriptors from the respective descriptor lists. Then the receive and transmit processes begin processing incoming and outgoing frames. The receive and transmit processes are independent of each other and can be started and stopped separately.
4.3 Functional Description

4.3.5 Receive Process

While in the running state, the receive process polls the receive descriptor list, attempting to acquire free descriptors. Incoming frames are processed and placed in acquired descriptors’ data buffers. Status information is written to receive descriptor 0.

4.3.5.1 Descriptor Acquisition

The 21140A always attempts to acquire an extra descriptor in anticipation of incoming frames. Descriptor acquisition is attempted if any of the following conditions are satisfied:

- When start/stop receive (CSR6<1>) sets immediately after being placed in the running state.
- When the 21140A begins writing frame data to a data buffer pointed to by the current descriptor, and the buffer ends before the frame ends.
- When the 21140A completes the reception of a frame, and the current receive descriptor has been closed.
- When the receive process is suspended because of a host-owned buffer (RDES0<31>=0), and a new frame is received.
- When receive poll demand is issued (Section 3.2.2.3).

4.3.5.2 Frame Processing

As incoming frames arrive, the 21140A recovers the incoming data and clock pulses, and then sends them to the receive engine. The receive engine strips the preamble bits and stores the frame data in the receive FIFO. Concurrently, the receive section performs address filtering depending on the results of inverse filtering (CSR6<6>), hash/perfect receive filtering mode (CSR6<0>), and hash-only receive filtering mode (CSR6<2>), and also its internal filtering table. If the frame fails the address filtering, it is ignored and purged from the FIFO. Frames that are shorter than 64 bytes, because of collision or premature termination, are also ignored and purged from the FIFO (unless pass bad frames bit CSR6<3> is set).

After 64 bytes have been received, the 21140A requests the PCI bus to begin transferring the frame data to the buffer pointed to by the current descriptor. While waiting for the PCI bus, the 21140A continues to receive and store the data in the FIFO. After receiving the PCI bus, the 21140A sets first descriptor (RDES0<9>), to delimit the frame. Then, the descriptors are released when the OWN (RDES0<31>) bit is reset to 0 either as the data buffers fill up or as the last segment of a frame is transferred to a buffer. If a frame is contained in a single descriptor, both last descriptor (RDES0<8>) and first descriptor (RDES0<9>) are set.
4.3 Functional Description

The 21140A sets last descriptor (RDE50<8>), fetches the next descriptor, and releases the RDE50 status bits in the last frame descriptor. After the last frame descriptor is released, the 21140A sets receive interrupt (CSR5<6>) and fetches the next descriptor. The same process repeats unless the 21140A encounters a descriptor flagged as being owned by the host. If this occurs, the receive process sets receive buffer unavailable (CSR5<7>) and then enters the suspended state. The position in the receive list is retained.

4.3.5.3 Receive Process Suspended

If any frames enter while the receive process is suspended, the 21140A fetches the current descriptor in host memory. If the descriptor is now owned by the 21140A, the receive process reenters the running state and starts the frame reception.

If a receive frame arrives while the receive process is suspended, the 21140A refetches the next descriptor. If the descriptor is still owned by the host, the 21140A increments the missed frames counter (CSR8<15:0>) and discards the current frame in the receive FIFO. If more than one frame is stored in the receive FIFO, the process repeats.

4.3.5.4 Receive Process State Transitions

Table 4–13 lists the receive process state transitions and the resulting actions.

Table 4–13 Receive Process State Transitions

<table>
<thead>
<tr>
<th>From State</th>
<th>Event</th>
<th>To State</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stopped</td>
<td>Start receive command.</td>
<td>Running</td>
<td>Receive polling begins from last list position or from the list head, if this is the first start receive command issued, or if the receive descriptor list address (CSR3) was modified by the driver.</td>
</tr>
<tr>
<td>Running</td>
<td>The 21140A attempts to acquire a descriptor owned by the host.</td>
<td>Suspended</td>
<td>Receive buffer unavailable (CSR5&lt;7&gt;) sets when the last acquired descriptor buffer is consumed. The position in the list is retained.</td>
</tr>
</tbody>
</table>

(continued on next page)
4.3 Functional Description

Table 4–13 (Cont.) Receive Process State Transitions

<table>
<thead>
<tr>
<th>From State</th>
<th>Event</th>
<th>To State</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>Stop receive command.</td>
<td>Stopped</td>
<td>Receive process is stopped after the current frame, if any, is completely transferred to data buffers. Receive process stopped (CSR5&lt;8&gt;) sets. The position in the list is retained.</td>
</tr>
<tr>
<td>Running</td>
<td>Memory or host bus parity error encountered.</td>
<td>Running</td>
<td>The 21140A operation is stopped and fatal bus error (CSR5&lt;13&gt;) sets. The 21140A remains in the running state. A software reset must be issued to release the 21140A.</td>
</tr>
<tr>
<td>Running</td>
<td>Reset command.</td>
<td>Stopped</td>
<td>Receive capability is cut off.</td>
</tr>
<tr>
<td>Suspended</td>
<td>Receive poll demand or incoming frame and available descriptor.</td>
<td>Running</td>
<td>Receive polling resumes from last list position.</td>
</tr>
<tr>
<td>Suspended</td>
<td>Stop receive command.</td>
<td>Stopped</td>
<td>Receive process stopped (CSR5&lt;8&gt;) sets.</td>
</tr>
<tr>
<td>Suspended</td>
<td>Reset command.</td>
<td>Stopped</td>
<td>None.</td>
</tr>
</tbody>
</table>

4.3.6 Transmit Process

While in the running state, the transmit process polls the transmit descriptor list for frames requiring transmission. After polling starts, it continues in either sequential descriptor ring order or chained order. When it completes frame transmission, status information is written into transmit descriptor 0 (TDES0). If the 21140A detects a descriptor flagged as owned by the host, or if an error condition occurs, the transmit process is suspended and both transmit buffer unavailable (CSR5<2>) and normal interrupt summary (CSR5<16>) are set.

Transmit interrupt (CSR5<0>) is set after completing transmission of a frame that has interrupt on completion (TDES1<31>) set in its last descriptor. When this occurs, the transmission process continues to run.

While in the running state, the transmit process can simultaneously acquire two frames. As the transmit process completes copying the first frame, it immediately polls the transmit descriptor list for the second frame. If the second frame is valid, the transmit process copies the frame before writing the status information of the first frame.
4.3 Functional Description

4.3.6.1 Frame Processing

Frames can be data-chained and span several buffers. Frames must be delimited by the first descriptor (TDES1<29>) and the last descriptor (TDES1<30>), respectively.

As the transmit process starts execution, the first descriptor must have TDES1<29> set. When this occurs, frame data transfers from the host buffer to the internal FIFO. Concurrently, if the current frame has the last descriptor TDES1<30> clear, the transmit process attempts to acquire the next descriptor. The transmit process expects this descriptor to have TDES1<29> clear. If TDES1<30> is clear, it indicates an intermediary buffer. If TDES1<30> is set, it indicates the last buffer of the frame.

After the last buffer of the frame has been transmitted, the 21140A writes back the final status information to the transmit descriptor 0 (TDES0) word of the descriptor that has the last segment set in transmit descriptor 1 (TDES1<30>). At this time, if interrupt on completion (TDES1<31>) was set, the transmit interrupt (CSR5<0>) is set, the next descriptor is fetched, and the process repeats.

Actual frame transmission begins after the internal FIFO has reached either a programmable threshold CSR6<15:14> (Table 3–41), or a full frame is contained in the FIFO. Descriptors are released (OWN bit TDES0<31> clears) when the 21140A completes the packet transmission.

4.3.6.2 Transmit Polling Suspended

Transmit polling can be suspended by either of the following conditions:

- The 21140A detects a descriptor owned by the host (TDES0<31>=0). To resume, the driver must give descriptor ownership to the 21140A and then issue a poll demand command.
- A frame transmission is aborted when a locally induced error is detected. The appropriate transmit descriptor 0 (TDES0) bit is set.

If either of the previous two conditions occur, both abnormal interrupt summary (CSR5<15>) and transmit interrupt (CSR5<0>) are set, and the information is written to transmit descriptor 0, causing the suspension.

In both of the cases previously described, the position in the transmit list is retained. The retained position is that of the descriptor following the last descriptor closed (set to host ownership) by the 21140A.
4.3 Functional Description

Note
The 21140A does not automatically poll the transmit descriptor list. The driver must explicitly issue a transmit poll demand command after rectifying the suspension cause, unless the transmit automatic polling (CSR0<18:17>) field is set to 1.

4.3.6.3 Transmit Process State Transitions
Table 4–14 lists the transmit process state transitions and the resulting actions.

Table 4–14 Transmit Process State Transitions

<table>
<thead>
<tr>
<th>From State</th>
<th>Event</th>
<th>To State</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stopped</td>
<td>Start transmit command.</td>
<td>Running</td>
<td>Transmit polling begins from one of the following positions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>· The last list position.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>· The head of the list, if this is the first start command issued after</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CSR4 was initialized or modified.</td>
</tr>
<tr>
<td>Running</td>
<td>The 21140A attempts acquisition of a</td>
<td>Suspended</td>
<td>Transmit buffer unavailable (CSR5&lt;2&gt;) is set.</td>
</tr>
<tr>
<td></td>
<td>descriptor owned by the host.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running</td>
<td>Frame transmission aborts because a locally</td>
<td>Suspended</td>
<td>The following bits are set:</td>
</tr>
<tr>
<td></td>
<td>induced underflow error (TDES0&lt;1&gt;) is</td>
<td></td>
<td>· TDES0&lt;1&gt; — Underflow error</td>
</tr>
<tr>
<td></td>
<td>detected (Section 4.2.2.1).</td>
<td></td>
<td>· CSR5&lt;5&gt; — Transmit underflow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>· CSR5&lt;15&gt; — Abnormal interrupt summary</td>
</tr>
<tr>
<td>Running</td>
<td>Stop transmit command.</td>
<td>Stopped</td>
<td>Transmit process is stopped after the current frame, if any, is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>transmitted. (continued on next page)</td>
</tr>
</tbody>
</table>
4.3 Functional Description

<table>
<thead>
<tr>
<th>From State</th>
<th>Event</th>
<th>To State</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>Frame transmission aborts because a transmit jabber timeout (TDES0&lt;14&gt;) was detected (Section 4.2.2.1).</td>
<td>Stopped</td>
<td>The following bits are set:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TDES0&lt;14&gt; — Transmit jabber time out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CSR5&lt;1&gt; — Transmit process stopped</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CSR5&lt;3&gt; — Transmit jabber time out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CSR5&lt;15&gt; — Abnormal interrupt summary</td>
</tr>
<tr>
<td>Running</td>
<td>Parity error detected by memory or host bus.</td>
<td>Running</td>
<td>Transmission is cut off and fatal bus error (CSR5&lt;13&gt;) is set. The 21140A remains in the running state. If a software reset occurs, normal operation continues.</td>
</tr>
<tr>
<td>Running</td>
<td>Reset command.</td>
<td>Stopped</td>
<td>Transmission is cut off. If CSR4 was not changed, the position in the list is retained. If CSR4 was changed, the next descriptor address is fetched from the header list (CSR4) when the poll demand command is issued. Transmit process stopped (CSR5&lt;1&gt;) is set.</td>
</tr>
<tr>
<td>Suspended</td>
<td>Transmit poll demand command issued.</td>
<td>Running</td>
<td>Transmit polling resumes from the last list position.</td>
</tr>
<tr>
<td>Suspended</td>
<td>Stop transmit command.</td>
<td>Stopped</td>
<td>Transmit process stopped (CSR5&lt;1&gt;) is set.</td>
</tr>
<tr>
<td>Suspended</td>
<td>Reset command.</td>
<td>Stopped</td>
<td>None.</td>
</tr>
</tbody>
</table>
This chapter describes the commands and operations of read and write cycles for a bus slave and a bus master. It also explains the initiation of termination cycles by the bus master or bus slave.

5.1 Overview

The peripheral component interconnect (PCI) is the physical interconnection used between highly integrated peripheral controller components and the host system. The 21140A uses the PCI bus to communicate with the host CPU and memory.

The 21140A is directly compatible with the PCI Local Bus Specification, Revision 2.0 and Revision 2.1. The 21140A supports a subset of the PCI-bus cycles (transactions). When communicating with the host, the 21140A operates as a bus slave; when communicating with the memory, as a bus master.

All signals are sampled on the rising edge of the clock. Each signal has a setup and hold aperture with respect to the rising clock edge. Refer to the Digital Semiconductor 21140A PCI Fast Ethernet LAN Controller Data Sheet for detailed timing information. Table 5-1 lists the codes for bus commands.

Note

The term clock cycle, as used in this chapter, refers to the PCI bus clock period specification.
5.2 Bus Commands

Table 5–1 lists the bus commands.

<table>
<thead>
<tr>
<th>c_be_l&lt;3:0&gt;</th>
<th>Command</th>
<th>Type of Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Interrupt acknowledge</td>
<td>Not supported</td>
</tr>
<tr>
<td>0001</td>
<td>Special cycle</td>
<td>Not supported</td>
</tr>
<tr>
<td>0010</td>
<td>I/O read</td>
<td>Supported as target</td>
</tr>
<tr>
<td>0011</td>
<td>I/O write</td>
<td>Supported as target</td>
</tr>
<tr>
<td>0100</td>
<td>Reserved</td>
<td>—</td>
</tr>
<tr>
<td>0101</td>
<td>Reserved</td>
<td>—</td>
</tr>
<tr>
<td>0110</td>
<td>Memory read</td>
<td>Supported as initiator and target</td>
</tr>
<tr>
<td>0111</td>
<td>Memory write</td>
<td>Supported as initiator and target</td>
</tr>
<tr>
<td>1000</td>
<td>Reserved</td>
<td>—</td>
</tr>
<tr>
<td>1001</td>
<td>Reserved</td>
<td>—</td>
</tr>
<tr>
<td>1010</td>
<td>Configuration read</td>
<td>Supported as target</td>
</tr>
<tr>
<td>1011</td>
<td>Configuration write</td>
<td>Supported as target</td>
</tr>
<tr>
<td>1100</td>
<td>Memory read multiple</td>
<td>Supported as initiator and target</td>
</tr>
<tr>
<td>1101</td>
<td>Dual-address cycle</td>
<td>Not supported</td>
</tr>
<tr>
<td>1110</td>
<td>Memory read line</td>
<td>Supported as target</td>
</tr>
<tr>
<td>1111</td>
<td>Memory write and invalidate</td>
<td>Supported as target</td>
</tr>
</tbody>
</table>

Initiator support for this command is controlled by CSR0<21>.

5.3 Bus Slave Operation

All host accesses to CSRs and configuration registers in the 21140A are executed with the 21140A acting as the slave. The bus slave operations include the following:

- I/O read
- I/O write
- Configuration read
- Configuration write
- Memory read
5.3 Bus Slave Operation

- Memory write
- Memory read/write (includes memory write and invalidate, memory read line, and memory read multiple)

Note
The 21140A does not support the following bus transactions:
- Interrupt acknowledge
- Special cycle
- Dual-address cycle

If the 21140A is targeted for a burst I/O or memory operation, it responds with a retry on the second data transaction.

5.3.1 Slave Read Cycle (I/O or Memory Target)

Figure 5–1 shows a typical slave read cycle. The 21140A I/O read cycle is executed as follows:

1. The host initiates the slave read cycle by asserting the frame_l signal, driving the address on the ad lines and driving the bus command (slave read operation) on the c_be_l lines.

2. The 21140A samples the address and the bus command on the next clock edge.

3. The host deasserts frame_l signal and asserts irdy_l signal.

4. The 21140A asserts devsel_l, and, at the next cycle, drives the data on the ad lines.

5. The read transaction completes when both irdy_l and trdy_l are asserted by the host and the 21140A, respectively, on the same clock edge.

   The 21140A assumes that c_be_l lines are 0000 (longword access).

   If the c_be_l lines are 1111, the ad bus read is 00000000H with correct parity.

6. The host and the 21140A terminates the cycle by deasserting irdy_l and trdy_l, respectively.
5.3 Bus Slave Operation

Figure 5–1 Slave Read Cycle

5.3.2 Slave Write Cycle (I/O or Memory Target)

Figure 5–2 shows a typical slave write cycle. The 21140A slave write cycle is executed as follows:

1. The host initiates the slave write cycle by asserting the frame_l signal, driving both the address on the ad lines and the bus command (slave write operation) on the c_be_l lines.

2. The 21140A samples the address and the bus command on the next clock edge.

3. The host deasserts frame_l and drives the data on the ad lines along with irdy_l.

4. The 21140A samples the data, and also asserts both devsel_l and trdy_l.

5. The host and the 21140A complete the write transaction by asserting both irdy_l and trdy_l, respectively, on the same clock edge.

The 21140A assumes that c_be_l lines are 0000 (longword access). If the c_be_l lines are 1111, the write transaction completes normally on the bus, but the write transaction to the CSR is not executed.
5.3 Bus Slave Operation

6. The host and the 21140A terminate the cycle by deasserting \texttt{irdy\_l} and \texttt{trdy\_l}, respectively.

**Figure 5–2 Slave Write Cycle**

5.3.3 Configuration Read and Write Cycles

The 21140A provides a way for software to analyze and configure the system before defining any address assignments or mapping. The 21140A provides 256 bytes of configuration registers. Section 3.1 describes these registers.

\begin{center}
\begin{tabular}{c}
\textbf{Note} \\
Configuration space accesses provide support for \texttt{c\_be\_l} lines.
\end{tabular}
\end{center}

Figure 5–3 shows a configuration read cycle. The host selects the 21140A by asserting \texttt{idsel}. The 21140A responds by asserting \texttt{devsel\_l}. The remainder of the read cycle is similar to the slave read cycle (Section 5.3.1).
5.3 Bus Slave Operation

Figure 5–3 Configuration Read Cycle

5.4 Bus Master Operation

All memory accesses are completed with the 21140A as the master on the PCI bus. The bus master operations include the following:

- Bus arbitration
- Memory read cycle
- Memory write cycle
- Termination cycles
5.4 Bus Master Operation

5.4.1 Bus Arbitration

The 21140A uses the PCI central arbitration mechanism with its unique request (req_l) and grant (gnt_l) signals. Figure 5–4 shows the bus arbitration mechanism. The 21140A bus arbitration is executed as follows:

1. The 21140A requests the bus by asserting req_l.
2. The arbiter, in response, asserts gnt_l (gnt_l can be deasserted on any clock).
3. The 21140A ensures that its gnt_l is asserted on the clock edge that it wants to drive frame_l. (If gnt_l is deasserted, the 21140A does not proceed.)
4. The 21140A deasserts req_l on the cycle that it asserts frame_l.

Figure 5–4 Bus Arbitration

```
<table>
<thead>
<tr>
<th>clk</th>
<th>req_l</th>
<th>gnt_l</th>
<th>frame_l</th>
<th>ad</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Address Data Data
5.4 Bus Master Operation

The 21140A uses gnt_l according to the following rules:

• If gnt_l is deasserted together with the assertion of frame_l, the 21140A continues its bus transaction.

• If gnt_l is asserted while frame_l remains deasserted, the arbiter can deassert gnt_l at any time. The 21140A does not assert frame_l until it is granted again.

5.4.2 Memory Read Cycle

Figure 5–5 shows the memory read cycle. The memory read cycle is executed as follows:

1. The 21140A initiates the memory read cycle by asserting frame_l signal. It also drives the address on the ad lines and the appropriate bus command (read operation) on the c_be_l lines.

2. The memory controller samples the address and the bus command on the next clock edge.

3. The 21140A asserts irdy_l until the end of the read transaction.

4. During the data transfer cycles, c_be_l indicates which byte lines are involved in each cycle. The 21140A drives 0000 on the c_be_l lines (longword access).

5. The memory controller drives the data on the ad lines and asserts trdy_l.

6. The 21140A samples the data on each rising clock edge when both irdy_l and trdy_l are asserted.

7. The previous two steps can be repeated a number of times.

8. The cycle is terminated when frame_l is deasserted by the 21140A.

9. Signal irdy_l is deasserted by the 21140A and trdy_l is deasserted by the memory controller.
5.4 Bus Master Operation

5.4.3 Memory Write Cycle

Figure 5–6 shows the memory write cycle. The memory write cycle is executed as follows:

1. The 21140A initiates the memory write cycle by asserting frame_l. It also drives both the address on the ad lines and the write operation bus command on the c_be_l lines.

2. The 21140A asserts irdy_l until the end of the transaction and drives the data on the ad lines.

3. The memory controller samples the address and the bus command on the next clock edge and asserts devsel_l.

4. During the data transfer cycles, the c_be_l lines indicate which byte lanes are involved in each cycle. The 21140A drives 0000 on the c_be_l lines (longword access).

5. The memory controller samples the data and asserts trdy_l. Each data cycle is completed on the rising clock edge when both irdy_l and trdy_l are asserted.

6. The previous two steps can be repeated a number of times.
5.4 Bus Master Operation

7. The 21140A terminates the cycle by deasserting frame_l.
8. The 21140A deasserts irdy_l and the memory controller deasserts trdy_l.

Figure 5–6 Memory Write

5.5 Termination Cycles

Termination cycles can be initiated during either slave or master cycles.

5.5.1 Slave-Initiated Termination

The 21140A initiates termination in slave mode when it is accessed by the host with I/O or memory burst cycles. The 21140A asserts stop_l to request the host to terminate the transaction. After stop_l is asserted, it remains asserted until frame_l is deasserted.

Figure 5–7 shows the retried device (the host) releasing the bus. The host retries the last data transaction after acquiring the bus in a different arbitration.
5.5 Termination Cycles

Figure 5–7 21140A-Initiated Retry Cycle

5.5.2 Master-Initiated Termination

A master-initiated termination can occur when the 21140A operates as a
master device on the PCI bus. Terminations can be issued by either the
21140A or the memory controller.

Terminations by the 21140A include the following:

- Normal completion
- Timeout
- Master abort

Memory-controller terminations (target) include the following:

- Target abort
- Target disconnect
- Target retry
5.5 Termination Cycles

5.5.2.1 21140A-Initiated Termination
A 21140A-initiated termination occurs when frame_l is deasserted and irdy_l is asserted. This indicates to the memory controller that the final data phase is in progress. The final data transfer occurs when both irdy_l and trdy_l assert. The transaction completes when both frame_l and irdy_l deassert. This is an idle bus condition.

5.5.2.1.1 Normal Completion  Figure 5–8 shows a normal completion cycle termination. This indicates that the 21140A successfully completed its intended transaction.

Figure 5–8  Normal Completion

5.5.2.1.2 Timeout  A timeout cycle termination occurs when the gnt_l line has been deasserted by the arbiter and the 21140A internal latency timer has expired. However, the intended transaction has not completed. A maximum of two additional data phases are permitted and then the 21140A performs a normal transaction completion.
5.5 Termination Cycles

5.5.2.1.3 Master Abort  If the target does not assert devsel_l within five cycles from the assertion of frame_l, the 21140A performs a normal completion. It then releases the bus and asserts both master abort (CFCS<29>) and fatal bus error (CSR5<13>). Figure 5–9 shows the 21140A master abort termination.

Figure 5–9 Master Abort

5.5.2.2 Memory-Controller-Initiated Termination
The memory controller or target can initiate certain terminations when the 21140A is the bus master.

5.5.2.2.1 Target Abort  The 21140A aborts the bus transaction when the target asserts stop_l and deasserts devsel_l. This indicates that the target wants the transaction to be aborted. The 21140A releases the bus and asserts both received target abort (CFCS<28>) and fatal bus error (CSR5<15>). Figure 5–10 shows the 21140A target abort.
5.5 Termination Cycles

**Figure 5–10 Target Abort**

- **clk**: 1 2 3 4 5
- **frame_l**: 
- **irdy_l**: 
- **trdy_l**: 
- **stop_l**: 
- **devsel_l**: 

**5.5.2.2.2 Target Disconnect Termination**

The 21140A terminates the bus transaction when the target asserts `stop_l`, which remains asserted until `frame_l` is deasserted. The 21140A releases the bus. Then, it retries at least the last data transaction after regaining the bus in another arbitration. Figure 5–11 shows the 21140A target disconnect.
5.5 Termination Cycles

Figure 5–11  Target Disconnect

---

5.5.2.2.3 Target Retry  The 21140A retries the bus transaction when the target asserts \texttt{stop\_l} and deasserts \texttt{trdy\_l}; \texttt{stop\_l} remains asserted until \texttt{frame\_l} is deasserted. The 21140A releases the bus. Then, it retries at least the last two data transactions after regaining the bus in another arbitration. Figure 5–12 shows the 21140A target retry.
5.5 Termination Cycles

Figure 5–12 Target Retry

5.6 Parity

The 21140A supports parity generation on all address, data, and command bits. Parity is always checked and generated on the 32-bit address and data bus (ad) as well as on the four command (c_be_l) lines. The 21140A always transfers stable values (1 or 0) on all the ad and c_be_l lines. If a data parity error is detected or perr_l is asserted when the 21140A is a bus master, the 21140A asserts detected parity error (CFCS<31>) and abnormal interrupt summary (CSR5<15>).

Figure 5–13 shows an example of parity generation on a memory write burst transaction. Note that valid parity is generated one cycle after the address and data segments were generated on the bus. One cycle after the assertion of the address parity, serr_l is asserted for one cycle because of an address parity error during slave operation. One cycle after the assertion of the data parity, perr_l is asserted because of a parity data error in either slave write or master read operations.
5.6 Parity

Figure 5–13 Parity Operation

5.7 Parking

Parking in the PCI bus allows the central arbiter to pause any selected agent. The 21140A enters the parking state when the arbiter asserts its gnt_l line while the bus is idle.
Network Interface Operation

This chapter describes the operation of the MII/SYM port and the serial (also referred to as SRL) port. It also describes media access control (MAC), loopback, and full-duplex operations.

6.1 MII/SYM Port

This section provides a description of the 100BASE-T terminology, the interface, the signals used, and the operating modes.

6.1.1 100BASE-T Terminology

This subsection provides a description of the 100BASE-T terminology used for the MII/SYM port. The following is a list of these terms:

- **Media-independent interface (MII)** is defined between the media access control (MAC) sublayer and the physical layer protocol (PHY) layer.
- **Physical coding sublayer (PCS)** is a sublayer within the PHY defined by 100BASE-T. The PCS implements the higher level functions of the PHY.
- **100BASE-T** is a generic term that refers to all members in the IEEE 802.3 family of 100Mb/s carrier-sense multiple access with collision detection (CSMA/CD) standards.
- **100BASE-T4** is the standard IEEE 802.3 for 100Mb/s, using unshielded twisted-pair (UTP) category 3 (CAT3) cables. The PHY requires four pairs.
- **100BASE-X** refers to all members of the IEEE 802.3 family contained in the 100Mb/s CSMA/CD standard. It implements a specific physical medium attachment (PMA) and PCS. Members of this family include 100BASE-TX and 100BASE-FX.
- **100BASE-TX** refers to the IEEE 802.3 PHY layer, which includes the 100BASE-X PCS and PMA together with the physical layer medium dependent (PMD). It uses UTP category 5 (CAT5) cables and STP cables.
6.1 MII/SYM Port

- 100BASE-FX refers to the IEEE 802.3 PHY layer, which includes the 100BASE-X PCS and PMA together with the PMD. It uses multimode fiber.

6.1.2 Interface Description

The MII port is an IEEE 802.3 compliant interface that provides a simple, inexpensive, and easily implemented interconnection between the MAC sublayer and the PHY layer. It also interconnects the PHY layer devices and station management (STA) entities. This interface has the following characteristics:

- Supports both 100Mb/s and 10Mb/s data rates
- Contains data and delimiters that are synchronous to clock references
- Provides independent, 4-bit-wide transmit and receive data paths
- Uses transistor–transistor (TTL) signal levels, compatible with common CMOS application-specific integrated circuit (ASIC) processes
- Provides a simple management interface
- Provides capability to drive a limited length of shielded cable

6.1.2.1 Signal Standards

Table 6–1 provides the standards that reference the MII/SYM port signal names with the appropriate IEEE 802.3 signal names.

<table>
<thead>
<tr>
<th>MII/SYM Signals</th>
<th>IEEE 802.3 Signals</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>mii_clsn</td>
<td>COL</td>
<td>Collision detect is asserted by the PHY layer when it detects a collision on the medium. It remains asserted while this condition persists. For the 10Mb/s implementation, collision is derived from the signal quality error of the PMA. For the 100Mb/s implementation, collision is defined for each PHY layer separately.</td>
</tr>
<tr>
<td>mii_crs</td>
<td>CRS</td>
<td>Carrier sense is asserted by the PHY layer when either the transmit or receive medium is active (not idle).</td>
</tr>
</tbody>
</table>

(continued on next page)
### 6.1 MII/SYM Port

Table 6–1 (Cont.) IEEE 802.3 and MII/SYM Signals

<table>
<thead>
<tr>
<th>MII/SYM Signals</th>
<th>IEEE 802.3 Signals</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>mii_dv</td>
<td>RX_DV</td>
<td>Receive data valid is asserted by the PHY layer when the first received preamble nibble is driven over the MII and remains asserted for the remainder of the frame.</td>
</tr>
<tr>
<td>mii_err</td>
<td>RX_ERR</td>
<td>Receive error is asserted by the PHY layer to indicate either a coding error or any other type of error that the MAC cannot detect was received. This error was detected on the frame currently being received and transferred over the MII.</td>
</tr>
<tr>
<td>mii_mdc</td>
<td>MDC</td>
<td>Management data clock is the clock reference for the mii_mdio signal.</td>
</tr>
<tr>
<td>mii_mdio</td>
<td>MDIO</td>
<td>Management data input/output is used to transfer control signals between the PHY layer and STA entity. The 21140A is capable of initiating the transfer of control signals to and from the PHY device by using this line.</td>
</tr>
<tr>
<td>mii/sym_rclk</td>
<td>RX_CLK</td>
<td>Receive clock synchronizes all receive signals.</td>
</tr>
<tr>
<td>mii/sym_rxd&lt;3:0&gt;</td>
<td>RXD&lt;3:0&gt;</td>
<td>These lines provide receive data.</td>
</tr>
<tr>
<td>mii/sym_tclk</td>
<td>TX_CLK</td>
<td>Transmit clock synchronizes all transmit signals.</td>
</tr>
<tr>
<td>mii/sym_txd&lt;3:0&gt;</td>
<td>TXD&lt;3:0&gt;</td>
<td>These lines provide transmit data.</td>
</tr>
<tr>
<td>mii_txen</td>
<td>TX_EN</td>
<td>Transmit enable is asserted by the MAC sublayer when the first transmit preamble nibble is driven over the MII and remains asserted for the remainder of the frame.</td>
</tr>
</tbody>
</table>

**Note**

The remaining three signals are activated when the MII/SYM port uses either the 100BASE-TX or 100BASE-FX applications.
6.1 MII/SYM Port

<table>
<thead>
<tr>
<th>MII/SYM Signals</th>
<th>IEEE 802.3 Signals</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>mii_sd</td>
<td>—</td>
<td>Signal detect indication is supplied by an external PMD device.</td>
</tr>
<tr>
<td>sym_rxd&lt;4&gt;</td>
<td>—</td>
<td>This line is used for receive data.</td>
</tr>
<tr>
<td>sym_txd&lt;4&gt;</td>
<td>—</td>
<td>This line is used for transmit data.</td>
</tr>
</tbody>
</table>

6.1.2.2 Operating Modes

The 21140A implements the MII/SYM port signals (Table 6–1) to support the following operating modes:

- **100Mb/s mode**—The 21140A implements the MII with a data rate of 100Mb/s and both the receive clock mii/sym_rclk and the transmit clock mii/sym_tclk operate at 25 MHz. In this mode, the 21140A can be used with any device that implements the 100BASE-T PHY layer (for example, 100BASE-TX, 100BASE-FX, or 100BASE-T4).

- **10Mb/s mode**—The 21140A implements the MII with a data rate of 10Mb/s and both the receive clock mii/sym_rclk and the transmit clock mii/sym_tclk operate at 2.5 MHz. In this mode, the 21140A can be used with any device that implements the 10Mb/s PHY layer and an MII.

- **100BASE-TX mode**—The 21140A implements certain functions of the PCS for STP PMD and UTP CAT5 PMD. The receive symbols are 5 bits wide and are transferred over the mii/sym_rxd<3:0> and sym_rxd<4> lines. The transmit symbols are also 5 bits wide and are transferred over the mii/sym_txd<3:0> and sym_txd<4> lines. These functions include the following:
  - 4-bit and 5-bit decoding and encoding
  - Start-of-stream delimiter (SSD) and end-of-stream delimiter (ESD) detection and generation
  - Bit alignment
  - Carrier detect
  - Collision detect
  - Symbol error detection
  - Scrambling and descrambling
  - Link timer
6.1 MII/SYM Port

This mode enables a direct interface with existing fiber distributed data interface (FDDI) TP-PMD devices that implement the physical functions.

- **100BASE-FX mode**—The 21140A implements certain functions of the PCS sublayer for multimode fiber. The receive symbols are 5 bits wide and are transferred over the \texttt{mii/sym\_rxd<3:0>} and \texttt{sym\_rxd<4>} lines. The transmit symbols are also 5 bits wide and are transferred over the \texttt{mii/sym\_txd<3:0>} and \texttt{sym\_txd<4>} lines. These functions include the following:
  - 4-bit and 5-bit decoding and encoding
  - SSD and ESD detection and generation
  - Bit alignment
  - Carrier detect
  - Collision detect
  - Symbol error detection
  - Link timer

This mode enables a direct interface with existing FDDI TP-PMD devices that implement the physical functions.

\begin{center} \textbf{Note} \end{center}

The SSD detection logic compares the incoming data to JK and not to IJK (this complies with IEEE 802.3, draft number 2).

6.2 Serial Port

The serial port consists of seven signals that provide a conventional interface to the existing 10Mb/s Ethernet ENDEC components.

6.3 Media Access Control Operation

The 21140A supports a full implementation of the MAC sublayer of IEEE 802.3. It can operate in half-duplex mode, full-duplex mode, and loopback mode.
6.3 Media Access Control Operation

Transmission
In half-duplex mode, the 21140A checks the line condition before starting to transmit. If the condition is clear to transmit (for example, the line is idle for an IPG time duration and the PMA link status is okay), the 21140A starts transmitting. Transmit enable (mii/sym_txen) asserts and data is being transferred. Depending on the operating mode selected, the data is transferred either through the MII/SYM port or the serial port interface.

During transmission, the 21140A monitors the line condition. If a collision is detected, the 21140A continues to transmit for a predetermined time as specified in IEEE 802.3 (for example, jam), and then it stops the transmission. The 21140A then implements the truncated binary backoff algorithm as defined in IEEE 802.3.

Depending on the operating mode, a collision is defined as follows:

- In serial mode, the signal srl_clsn asserts.
- In MII mode, the signal mii/sym_clsn asserts.
- In either 100BASE-TX or 100BASE-FX modes, the receive input is active while the 21140A transmits.

If the 21140A fails to transmit a frame due to collisions after 16 attempts, the 21140A reports an excessive collision and stops transmitting the frame.

In full-duplex mode, the 21140A starts transmitting a frame provided that IPG duration time has elapsed since its previous transmission. There is no collision in full-duplex mode, so the transmission is guaranteed to end successfully.

Reception
In both half-duplex and full-duplex modes, the 21140A monitors the line for a new frame transmission. When a frame is detected, the 21140A starts to assemble the frame. Depending on the 21140A operating mode, a new frame transmission is defined as follows:

- In serial mode, receive enable (srl_rxen) asserts.
- In MII mode, both data valid (mii_dv) and carrier sense (mii_crs) assert.
- In either 100BASE-TX or 100BASE-FX modes, the receive input becomes active.

While the frame is being assembled, the 21140A continues to monitor the line condition. The reception ends with an error if the frame is not a valid Ethernet or IEEE 802.3 MAC frame (Section 6.3.1), or if one of the following conditions occur:
6.3 Media Access Control Operation

- The 21140A is operating in an MII mode and receive error (mii_err) asserts during a frame reception.
- The 21140A is operating in either 100BASE-TX or 100BASE-FX mode and one of the following conditions occur:
  - An invalid symbol is detected.
  - The frame does not start with the symbol J followed by the symbol K.
  - The frame does not end with the symbol T followed by the symbol R.
  - Between the JK and TR symbols, the frame contains any symbol that does not belong to the data code groups defined in IEEE 802.3, clause 24.

6.3.1 MAC Frame Format

The 21140A handles both IEEE 802.3 and Ethernet MAC frames. While operating in either the 100BASE-FX mode or 100BASE-TX mode, the 21140A encapsulates the frames it transmits according to the IEEE 802.3, clause 24. Receive frames are encapsulated according to the IEEE 802.3, clause 24.

The changes between a MAC frame (Section 6.3.1.1) and the encapsulation used when operating either in 100BASE-TX or 100BASE-FX mode are listed as follows:

1. The first byte of the preamble in the MAC frame is replaced with the JK symbol pair.
2. After the frame check sequence (FCS) byte of the MAC frame, the TR symbol pair is inserted.
6.3 Media Access Control Operation

6.3.1 Ethernet and IEEE 802.3 Frames

Ethernet is the generic name for the network type. An Ethernet frame has a minimum length of 64 bytes and a maximum length of 1518 bytes, exclusive of the preamble and the SFD.

An Ethernet frame format consists of the following:
- Preamble
- Start frame delimiter (SFD)
- Two address fields
- Type or length field
- Data field
- Frame check sequence (CRC value)

6.3.1.2 Ethernet Frame Format Description

Figure 6–1 shows the Ethernet frame format and Table 6–2 describes the byte fields.

Figure 6–1 Ethernet Frame Format

<table>
<thead>
<tr>
<th>Preamble</th>
<th>SFD</th>
<th>Destination Address</th>
<th>Source Address</th>
<th>Type/Length</th>
<th>Data (46…1500)</th>
<th>CRC (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(7)</td>
<td>(1)</td>
<td>(6)</td>
<td>(6)</td>
<td>(2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Numbers in parentheses indicate field length in bytes.
6.3 Media Access Control Operation

Table 6–2 Ethernet Frame Format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>A 7-byte field of 56 alternating 1s and 0s, beginning with a 0.</td>
</tr>
<tr>
<td>SFD — Start frame delimiter</td>
<td>A 1-byte field that contains the value 10101011; the most significant bit is transmitted and received first.</td>
</tr>
<tr>
<td>Destination address</td>
<td>A 6-byte field that contains either a specific station address, the broadcast address, or a multicast (logical) address where this frame is directed.</td>
</tr>
<tr>
<td>Source address</td>
<td>A 6-byte field that contains the specific station address where this frame originated.</td>
</tr>
<tr>
<td>Type/length</td>
<td>A 2-byte field that indicates whether the frame is in IEEE 802.3 format or Ethernet format (Table 6–3).</td>
</tr>
<tr>
<td></td>
<td>A field greater than 1500 is interpreted as a type field, which defines the type of protocol of the frame.</td>
</tr>
<tr>
<td></td>
<td>A field smaller than or equal to 1500 (05-DC) is interpreted as a length field, which indicates the number of data bytes in the frame.</td>
</tr>
<tr>
<td>Data</td>
<td>A data field consists of 46 to 1500 bytes of information that is fully transparent because any arbitrary sequence of bits can occur.</td>
</tr>
<tr>
<td></td>
<td>A data field shorter than 46 bytes, which is specified by the length field, is allowed. Unless padding is disabled (TDES1&lt;23&gt;), it is added by the 21140A when transmitting to fill the data field up to 46 bytes.</td>
</tr>
<tr>
<td>CRC</td>
<td>A frame check sequence is a 32-bit cyclic redundancy check (CRC) value that is computed as a function of the destination address field, source address field, type field, and data field. The FCS is appended to each transmitted frame, and is used at reception to determine if the received frame is valid.</td>
</tr>
</tbody>
</table>

Table 6–3 lists the possible values for the frame format. The values are expressed in hexadecimal notation and the 2-byte field is displayed with a hyphen separating the 2 bytes. The byte on the left of the hyphen is the most significant byte and is transmitted first.

Table 6–3 Frame Format Table

<table>
<thead>
<tr>
<th>Frame Format</th>
<th>Length or Type</th>
<th>Hexadecimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.3</td>
<td>Length field</td>
<td>00-00 to 05-DC</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Type field</td>
<td>05-DD to FF-FF</td>
</tr>
</tbody>
</table>
6.3 Media Access Control Operation

The CRC polynomial, as specified in the Ethernet specification, is as follows:

\[ FCS(X) = X^{31} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1 \]

The 32 bits of the CRC value are placed in the FCS field so that the \( X^{31} \) term is the right-most bit of the first octet, and the \( X^0 \) term is the left-most bit of the last octet. The bits of the CRC are thus transmitted in the order \( X^{31}, X^{30}, \ldots, X^1, X^0 \).

6.3.2 Ethernet Reception Addressing

The 21140A can be set up to recognize any one of the Ethernet receive address groups described in Table 6–4. Each group is separate and distinct from the other groups.

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16-address perfect filtering</td>
</tr>
</tbody>
</table>

The 21140A provides support for the perfect filtering of up to 16 Ethernet physical or multicast addresses. Any mix of addresses can be used for this perfect filter function of the 21140A. The 16 addresses are issued in setup frames to the 21140A.

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>One physical address, unlimited multicast addresses imperfect filtering</td>
</tr>
</tbody>
</table>

The 21140A provides support for one, single physical address to be perfectly filtered with an unlimited number of multicast addresses to be imperfectly filtered. This case supports the needs of applications that require one, single physical address to be filtered as the station address, while enabling reception of more than 16 multicast addresses, without suffering the overhead of pass-all-multicast mode. The single physical address, for perfect filtering, and a 512-bit mask, for imperfect filtering using a hash algorithm, are issued in a setup frame to the 21140A. When hash hits are detected, the 21140A delivers the received frame (Section 4.2.3).
6.3 Media Access Control Operation

Table 6–4 (Cont.)  Ethernet Receive Address Groups

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3     | Unlimited physical addresses, unlimited multicast addresses imperfect filtering  
The 21140A provides support for unlimited physical addresses to be imperfectly filtered with an unlimited number of multicast addresses to be imperfectly filtered as well. This case supports applications that require more than one physical address to be filtered as the station address, while enabling the reception of more than 16 multicast addresses, without suffering the overhead of pass-all-multicast mode. A 512-bit mask, for imperfect filtering using a hash algorithm, is issued in a setup frame to the 21140A. When hash hits are detected, the 21140A delivers the received frame (Section 4.2.3). |
| 4     | Promiscuous Ethernet reception  
The 21140A provides support for reception of all frames on the network regardless of their destination. This function is controlled by a CSR bit. This group is typically used for network monitoring. |
| 5     | 16-address perfect filtering and reception of all multicast Ethernet addresses  
This group augments the receive address Group 1 and also receives all frames on the Ethernet with a multicast address. |
| 6     | 16-address inverse filtering  
In this mode, the 21140A applies the reverse filter of Group 1. The 21140A provides support for the rejection of up to 16 Ethernet physical or multicast addresses. Any mix of addresses may be used for this filter function of the 21140A. The 16 addresses are issued in setup frames to the 21140A. |

6.3.3 Detailed Transmit Operation

This section describes the transmit operation in detail, as supported by the 21140A. This description includes the specific control register definitions, setup frame definitions, and a mechanism used by the host processor software to manipulate the transmit list (that is, the descriptors and buffers that can be found in Section 4.2).
6.3 Media Access Control Operation

6.3.3.1 Transmit Initiation

The host CPU initiates a transmit by storing the entire information content of the frame to be transmitted in one or more buffers in memory. The host processor software prepares a companion transmit descriptor, also in host memory, for the transmit buffer and signals the 21140A to take it. After the 21140A has been notified of this transmit list, the 21140A starts to move the data bytes from the host memory to the internal transmit FIFO.

When the transmit FIFO is adequately filled to the programmed threshold level, or when there is a full frame buffered into the transmit FIFO, the 21140A begins to encapsulate the frame.

The threshold level can be programmed with various quantities (Table 3–41). The lower threshold is for low bus latency systems and the high threshold is for high bus latency systems.

The transmit encapsulation is performed by the transmit state machine, which delays the actual transmission of the data onto the network until the network has been idle for a minimum interpacket gap (IPG) time.

6.3.3.2 Frame Encapsulation

The transmit data frame encapsulation stream consists of appending the 56 preamble bits together with the SFD to the basic frame beginning and the FCS (for example, CRC), to the basic frame end.

The basic frame read from the host memory includes the destination address field, the source address field, the type/length field, and the data field. If the data field length is less than 46 bytes, and padding (TDES1<23>) is enabled, the 21140A pads the basic frame with the pattern 00 for up to 46 bytes before appending the FCS field to the end.

While operating either in 100BASE-FX mode or 100BASE-TX mode, the 21140A encapsulates the frames it transmits according to IEEE 802.3, clause 24 and the receive frame is encapsulated as defined in IEEE 802.3, clause 24.

The changes between a MAC frame (Section 6.3.1) and the encapsulation used when operating either in 100BASE-TX or 100BASE-FX modes are listed as follows:

1. The first byte of the preamble in the MAC frame is replaced with the JK symbol pair.
2. After the FCS byte of the MAC frame, the TR symbol pair is inserted.
6.3 Media Access Control Operation

6.3.3 Initial Deferral

The 21140A constantly monitors the line and can initiate a transmission any time the host CPU requests it. Actual transmission of the data onto the network occurs only if the network has been idle for a 96-bit time period, and any backoff time requirements have been satisfied.

The IPG time is divided into two parts: IPS1 and IPS2.

1. IPS1 time (60-bit time): the 21140A monitors the network for an idle state. If a carrier is sensed on the serial line during this time, the 21140A defers and waits until the line is idle again before restarting the IPS1 time count.

2. IPS2 time (36-bit time): the 21140A continues to count time even though a carrier has been sensed on the network, and thus forces collisions on the network. This enables all network stations to have access to the serial line.

6.3.3.4 Collision

A collision occurs when concurrent transmissions from two or more Ethernet nodes take place. Depending on the mode of operation, the 21140A detects a collision event in one of the following ways:

- In serial mode, when srl_clsn asserts.
- In MII mode, when mii/sym_clsn asserts.
- In either 100BASE-TX or 100BASE-FX mode, when the receive input is active while the 21140A transmits.

When the 21140A detects a collision while transmitting, it halts the transmission of the data, and instead, transmits a jam pattern consisting of hexadecimal AAAAAAAA. At the end of the jam transmission, the 21140A begins the backoff wait period.

If the collision was detected during the preamble transmission, the jam pattern is transmitted after completing the preamble (if the 21140A is in 100BASE-FX or 100BASE-TX operating modes, this includes the JK symbol pair). This results in a minimum 96-bit fragment.
6.3 Media Access Control Operation

The 21140A scheduling of retransmission is determined by a controlled randomization process called truncated binary exponential backoff. The delay is an integer multiple of slot times. The number of slot times of delay before the $n^{th}$ retransmission attempt is chosen as a uniformly distributed random integer $r$ in the range:

$$0 \leq r < 2^k$$

$$k = \min(n, N) \text{ and } N = 10$$

When 16 attempts have been made at transmission and all have been terminated by a collision, the 21140A sets an error status bit in the descriptor (TDES0<8>) and, if enabled, issues a normal transmit termination (CSR5<0>) interrupt to the host.

Note
The jam pattern is a fixed pattern that is not compared with the actual frame CRC. This has the very low probability (0.5^32) of having a jam pattern equal to the CRC.

6.3.3.5 Terminating Transmission

A specific frame transmission is terminated by any of the following conditions:

- Normal—The frame has been transmitted successfully. When the last byte is serialized, the pad and CRC are optionally appended and transmitted, thus concluding frame transmission.

- Underflow—Transmit data is not ready when needed for transmission. The underflow status bits (TDES0<1> and CSR5<5>) are set, and the packet is terminated on the network with a bad CRC.

- Excessive collisions—If a collision occurs for the 15th consecutive retransmission attempt of the same frame, TDES0<8> is set.

- Jabber timer expired—If the timer expires (TDES0<14> sets) while transmission continues, the programmed interval transmission is cut off.

- Memory error—This generic error indicates either a host bus timeout or a host memory error.

- Late collision—If a collision occurs after the collision window (transmitting at least 64 bytes), transmission is cut off and TDES0<9> sets.
6.3 Media Access Control Operation

At the completion of every frame transmission, status information about the frame is written into the transmit descriptor. Status information is written into CSR5 if an error occurs during the operation of the transmit machine itself. If a normal interrupt summary (CSR7<16>) is enabled, the 21140A issues a normal transmit termination interrupt (CSR5<0>) to the host.

6.3.3.6 Transmit Parameter Values

Table 6–5 lists the transmit parameter values for both the 10Mb/s and 100Mb/s serial bit rates.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defer time</td>
<td>IPS1+IPS2=96-bit time</td>
<td>—</td>
</tr>
<tr>
<td>IPS1</td>
<td>—</td>
<td>60-bit time period</td>
</tr>
<tr>
<td>IPS2</td>
<td>—</td>
<td>36-bit time period</td>
</tr>
<tr>
<td>Slot time interval</td>
<td>—</td>
<td>512-bit time period</td>
</tr>
<tr>
<td>Network acquisition time</td>
<td>—</td>
<td>512-bit time period</td>
</tr>
<tr>
<td>Transmission attempts</td>
<td>—</td>
<td>16</td>
</tr>
<tr>
<td>Backoff limit</td>
<td>—</td>
<td>10</td>
</tr>
<tr>
<td>Jabber timer</td>
<td>Default</td>
<td>16000- to 20000-bit time</td>
</tr>
<tr>
<td></td>
<td>Programmable range</td>
<td>26000- to 32000-bit time</td>
</tr>
</tbody>
</table>

6.3.4 Detailed Receive Operation

This section describes the detailed receive operation as supported by the 21140A. This description includes the specific control register definitions, setup frame definitions, and a mechanism used by the host processor software to manipulate the receive list (that is, the descriptors and buffers that can be found in Section 4.2).

6.3.4.1 Receive Initiation

The 21140A continuously monitors the network when reception is enabled. When activity is recognized, it starts to process the incoming data. Depending on the mode of operation, the 21140A detects activity in one of the following ways:

- In serial mode, when \texttt{srl\_rxen} asserts.
6.3 Media Access Control Operation

- In MII mode, when both *mii_dv* and *mii_crs* assert.
- In either 100BASE-TX or 100BASE-FX mode, when the receive input is active.

After detecting receive activity on the line, the 21140A starts to process the preamble bytes based on the mode of operation.

6.3.4.2 Preamble Processing

Preamble processing varies depending on the 21140A operating mode. The next two subsections describe how this processing is handled.

6.3.4.2.1 MII or Serial Mode Preambles

In either MII or serial mode, the preamble, as defined by Ethernet, can be up to 64 bits (8 bytes) long.

The 21140A allows any arbitrary preamble length. However, depending on the mode, there is a minimum preamble length:

- In MII/SYM mode, at least 8 bits are required to recognize a preamble.
- In SRL mode, at least 16 bits are required to recognize a preamble.
- While in snooze mode, at least 20 bits are required to recognize a preamble for both MII/SYM and SRL modes.

Recognition occurs as follows:

- In MII/SYM mode, the 21140A checks for the start frame delimiter (SFD) byte content of 10101011.
- In SRL mode:
  1. The first 8 preamble bits are ignored.
  2. The 21140A checks for the start frame delimiter (SFD) byte content of 10101011.

While checking for SFD, if the 21140A receives a 11 (before receiving 14 bytes in SRL mode or 6 bytes in MII mode) or a 00 (everywhere), the reception of the current frame is aborted. The frame is not received, and the 21140A waits until the network activity stops (Section 6.3.4.1) before monitoring the network activity for a new preamble.

Figure 6–2 shows the preamble recognition sequence bit fields.
6.3 Media Access Control Operation

Figure 6–2  Preamble Recognition Sequence in SRL Mode

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>X X 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1</td>
<td>X X X</td>
</tr>
</tbody>
</table>

Neglected SFD

6.3.4.2.2 100BASE-TX or 100BASE-FX Mode Preambles  When operating in either 100BASE-TX or 100BASE-FX mode, the 21140A expects the frame to start with the symbol pair JK followed by the preamble, as specified in Section 6.3.4.2.1. If a JK symbol pair is not detected, the reception of the current frame is aborted (not received), and the 21140A waits until the network activity stops before monitoring the network activity for a new preamble.

6.3.4.3 Address Matching

Ethernet addresses consist of two 6-byte fields: one field for the destination address and one for the source address. The first bit of the destination address signifies whether it is a physical address or a multicast address (Table 6–6).

Table 6–6 Destination Address Bit 1

<table>
<thead>
<tr>
<th>Bit 1</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Station address (physical)</td>
</tr>
<tr>
<td>1</td>
<td>Multicast address</td>
</tr>
</tbody>
</table>

The 21140A filters the frame based on the Ethernet receive address group (Section 6.3.2) filtering mode that has been enabled.

If the frame address passes the filter, the 21140A removes the preamble and delivers the frame to the host processor memory. If, however, the address does not pass the filter when the mismatch is recognized, the 21140A terminates its reception. In this case, no data is sent to the host memory nor is any receive buffer consumed.

6.3.4.4 Frame Decapsulation

The 21140A checks the CRC bytes of all received frames before releasing them to the host processor. When operating in either 100BASE-TX or 100BASE-FX mode, the 21140A also checks that the frame ends with the TR symbol pair: if not, the 21140A reports an FCS error in the packet reception status.
6.3 Media Access Control Operation

6.3.4.5 Terminating Reception
Reception of a specific frame is terminated when any of the following conditions occur:

- Normal termination—The network activity (Section 6.3.4.1) stops for the various operating modes.
- Overflow—The receive DMA cannot empty the receive FIFO into host processor memory as rapidly as it is filled, and an error occurs as frame data is lost. The overflow status bit (RDES0<0>) is set.
- Watchdog timer expired—If the timer expires (CSR5<9> and RDES0<4> both set) while reception is still in process, and reception is cut off.
- Collision—If a late collision occurs after the reception of 64 bytes of the packet, the collision seen status bit RDES0<6> is set.

6.3.4.6 Frame Reception Status
When reception terminates, the 21140A determines the status of the received frame and loads it into the receive status word in the buffer descriptor. An interrupt is issued if enabled. The 21140A may report the following conditions at the end of frame reception:

- Overflow—The 21140A receive FIFO overflowed.
- CRC error—The 32-bit CRC transmitted with the frame did not match the CRC calculated upon reception. The CRC check is always executed and is independent of any other errors. In addition, the 21140A reports a CRC error in any of the following cases:
  - The mii_err signal asserts during frame reception over the MII when operating in one of the MII operating modes.
  - The 21140A is operating in either the 100BASE-TX or 100BASE-FX mode and one of the following occur:
    * An invalid symbol is received in the middle of the frame.
    * The frame does not end with the symbol T followed by the symbol R.
    * Between the JK and TR symbol pairs, the frame contains any symbol that does not belong to the data code groups defined in IEEE 802.3 clause 24.
6.3 Media Access Control Operation

- **Dribbling bits error**—This indicates the frame did not end on a byte boundary. The 21140A signals a dribbling bits error only if the number of dribbling bits in the last byte is 4 in MII/SYM operating mode, or at least 3 in 10Mb/s serial operating mode. Only whole bytes are run through the CRC check. This means that although up to 7 dribbling bits may have occurred and a framing error was signaled, the frame might nevertheless have been received correctly.

- **Alignment error**—A CRC error and a dribbling bit error occur together. This means that the frame did not contain an integral number of bytes and the CRC check failed.

- **Frame too short (runt frame)**—A frame containing less than 64 bytes was received (including CRC). Reception of runt frames is optionally selectable. The 21140A defaults to inhibit reception of runts.

- **Frame too long**—A frame containing more than 1500 bytes was received. Reception of frames too long completes with an error indication.

- **Collision seen**—A frame collision occurred after the 64 bytes following the start frame delimiter (SFD) were received. Reception of such frames is completed and an error bit is set in the descriptor.

- **Descriptor error**—An error was found in one of the receive descriptors, which disabled the correct reception of an incoming frame.

6.4 Loopback Operations

The 21140A supports two loopback modes: internal loopback and external loopback. The default value of the loopback data rate is 10Mb/s. When the MII/SYM port is enabled, the loopback data rate is 10/100Mb/s.

6.4.1 Internal Loopback Mode

Internal loopback mode is normally used to verify that the internal logic operations function correctly. Internal loopback mode is enabled according to CSR6<11:10>. Internal loopback mode includes all the internal functions. In loopback mode, the 21140A disengages from the Ethernet wire.

6.4.2 External Loopback Mode

External loopback mode is normally used to verify that the logic operations up to the Ethernet wire function correctly. External loopback mode is enabled according to CSR6<11:10>.
6.5 Full-Duplex Operation

6.5 Full-Duplex Operation

The 21140A activates the transmit and receive processes simultaneously. It also supports receive back-to-back packets with an interpacket gap (IPG) of 96-bit times in parallel with transmit back-to-back packets with an IPG of 96-bit times.

The 21140A implements a programmable full-duplex operating mode (CSR6<9>) bit that commands the MAC to ignore both the carrier and the collision detect signal.

The driver must take the following actions to enter full-duplex operation.

1. Stop the receive and transmit processes by writing 0 to CSR6<1> and CSR6<13> fields, respectively. The driver must wait for any previously scheduled frame activity to cease by polling the transmit process state (<22:20>) and receive process state (<19:17>) fields in CSR5.

2. Prepare appropriate transmit and receive descriptor lists in host memory. These lists can use the existing lists at the point of suspension, or can create new lists that must be identified to the 21140A by referencing the receive list base address in CSR3 and the transmit list base address in CSR4.

3. Set full-duplex mode (CSR6<9>).

4. Place the transmit and receive processes in the running state by using the start commands.

5. Resume normal processing. Execute any 21140A interrupts.

6.6 Capture Effect—A Value-Added Feature

As a value-added feature, the 21140A provides a complete solution to an unsolved Ethernet and IEEE 802.3 problem referred to as capture effect. This solution is not part of the IEEE 802.3 standard. A device implementing this feature deviates from the IEEE 802.3 standard backoff algorithm. Therefore, this feature is optional and can be enabled or disabled using the CSR6<17> control bit.
6.6 Capture Effect—A Value-Added Feature

6.6.1 What Is Capture Effect?

Consider two stations on the line, station A and station B. Each station has a significant amount of data ready to transmit. Each station is able to satisfy the minimum IPG rules (both from transmit-to-transmit and from receive-to-transmit operations). The following steps show how station A captures the line (Table 6–7):

1. Station A (with data A1) and station B (with data B1) both attempt to transmit simultaneously within a slot time of 51.2 µs. Each station has an initial collision count set to 0.
2. The stations experience a collision. Both stations increment their collision count to 1.
3. Each station picks a backoff time value that is uniformly distributed from 0 to (2^n)–1 slots. In this example, station B selects a backoff of 1 (a 50% probability), and station A selects a backoff of 0.
4. Station A successfully transmits its A1 data packet. Station B waits for data A1 to be transmitted before attempting to retransmit data B1.
5. Collision count at station B remains at 1, while collision count at station A is reset to 0.
6. If station A has another packet (data A2) ready to transmit while station B still wants to transmit its packet (data B1), the stations both contend for the line again.
7. If these stations collide, the backoff value available for station A is 0 or 1 slots. The backoff value available for station B is 0, 1, 2, or 3 slots because the collision count is now at 2 (station A's collision count is at 1). Station A is more likely to succeed and transmit data A2, while data B1 from station B begins the deferral of completing its backoff interval.
8. It is possible, with this type of behavior between stations, that in the 2-node Ethernet, a station can capture the channel for an unfair amount of time. One station can transmit a significant number of packets back to back, while the other station continues to backoff further and further.
9. This process could continue until station B reaches the maximum number of collisions, 16, while attempting to transmit data B1. At this time, station B would access the line and transmit data B1.
6.6 Capture Effect—A Value-Added Feature

Note

If station A completes the transmitting of a stream of packets during this type of capture, and station B is still in backoff, potentially for a long time, the line is idle for this period of time.

<table>
<thead>
<tr>
<th>Station A</th>
<th>Line</th>
<th>Station B</th>
<th>Collision Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit packet A1</td>
<td>Collision</td>
<td>Transmit packet B1</td>
<td>0 0</td>
</tr>
<tr>
<td>Backoff 0, 1</td>
<td></td>
<td>Backoff 0, 1</td>
<td>1 1</td>
</tr>
<tr>
<td>Transmit packet A1</td>
<td>Packet A1</td>
<td>Backoff</td>
<td>0 1</td>
</tr>
<tr>
<td>Transmit packet A2</td>
<td>Collision</td>
<td>Transmit packet B1</td>
<td>0 1</td>
</tr>
<tr>
<td>Backoff 0, 1</td>
<td></td>
<td>Backoff 0, 1, 2, 3</td>
<td>1 2</td>
</tr>
<tr>
<td>Transmit packet A2</td>
<td>Packet A2</td>
<td>Backoff</td>
<td>0 2</td>
</tr>
<tr>
<td>Transmit packet A3</td>
<td>Collision</td>
<td>Transmit packet B1</td>
<td>0 2</td>
</tr>
<tr>
<td>Backoff 0, 1</td>
<td></td>
<td>Backoff 0, 1, 2, ... 7</td>
<td>1 3</td>
</tr>
</tbody>
</table>

6.6.2 Resolving Capture Effect

The 21140A generally resolves the capture effect by having the station use, after a successful transmission of a frame by a station, a 2–0 backoff algorithm on the next transmit frame. If the station senses a frame on the network before it attempts to transmit the next frame, regardless of whether the sensed frame destination address matches the station’s source address, the station returns to use the standard truncated binary exponential backoff algorithm (Section 6.3.3.4).

When the station executes the 2–0 backoff algorithm, it always waits for a 2-slot period on the first collision, and for a 0-slot period on the second collision. For retransmission attempts greater than 2, it uses the standard truncated, binary exponential backoff algorithm.
6.6 Capture Effect—A Value-Added Feature

Table 6–8 summarizes the 2-0 backoff algorithm.

Table 6–8  2–0 Backoff Algorithm

<table>
<thead>
<tr>
<th>Retransmission Attempts</th>
<th>Backoff Period (Number of Slot Times)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 1</td>
<td>Backoff = 2 slots</td>
</tr>
<tr>
<td>n = 2</td>
<td>Backoff = 0 slots</td>
</tr>
<tr>
<td>n = 3 to 15</td>
<td>Backoff = $0 \leq r &lt; 2^k$</td>
</tr>
<tr>
<td></td>
<td>$k = \min(n, N)$ and $N = 10$</td>
</tr>
<tr>
<td></td>
<td>$r$ = uniformly distributed random integer</td>
</tr>
</tbody>
</table>

6.6.3 Enhanced Resolution for Capture Effect

The 21140A offers an enhanced resolution for capture effect. The enhancement is made by incorporating a stopped backoff algorithm (with the 2-0 backoff algorithm) to reduce collision while maintaining the key properties of the 2-0 backoff algorithm.

When the enhanced resolution for the capture effect bit is set (CSR6<31>), the 21140A activates the stopped backoff algorithm as follows: in a back-to-back transmit, while in backoff after the first collision ($n=1$, where $n$ is the retransmission attempts), the 21140A stops its backoff timer for the duration when the channel is busy. It continues its backoff timer when the channel is idle. For any other collision cases, the backoff timer is not stopped.

6.7 Power-Saving Modes

The 21140A incorporates two different power-saving modes: sleep mode and snooze mode. The following subsections describe these power-saving modes.

6.7.1 Sleep Power-Saving Mode

Sleep mode can be activated when the 21140A is not being used (for example, not connected to the network) and it is important to reduce its power dissipation. While in sleep mode, most of the circuitries are disabled. This includes the DMA machine, FIFOs, RxM, TxM, and the general-purpose timer. The PCI section is not affected and access to the 21140A configuration registers remains possible. Access to the 21140A CSRs is not allowed.
6.7 Power-Saving Modes

To enter sleep mode, the driver must take the following actions:
1. Stop the receive and transmit processes by writing 0 to the CSR6<1> and CSR6<13> fields, respectively. The driver must wait for any previously scheduled frame activity to cease. This is done by polling the transmit process state (CSR5<22:20>) and the receive process state (CSR5<19:17>).
2. Set the CFDD<31> bit.

To exit sleep mode, the driver must take the following actions:
2. Wait 10 ms.
3. Start the receive and transmit processes by writing 1 to the CSR6<1> and CSR6<13> fields respectively.

6.7.2 Snooze Power-Saving Mode

Snooze mode is a dynamic power-saving mode. When the snooze mode bit (CFDD<30>) is set, the 21140A reduces its power dissipation unless one or more of the following conditions is true:
- PCI slave or master access is conducted
- Transmit process is in the running state
- Receive process is in the running state but not waiting for a packet
- Receive FIFO is not empty
- MAC receive engine is not idle
- Carrier is sensed
- Link pass or link fail interrupt occurred

When none of these conditions is true, the 21140A disables all its internal circuitries except for the PCI interface (not including the Manchester decoder that uses the 100-MHz phases). The 21140A automatically and immediately reenables all its circuitries when at least one of the following occurs:
- PCI slave access is conducted
- Carrier is sensed
- Link pass or link fail interrupt occurred

This results in the 21140A dynamically getting into and out of low-power mode, and overall power dissipation is reduced.
6.7 Power-Saving Modes

To activate snooze mode, the driver should set CFDD<30>, which is the snooze mode bit. To stop snooze mode, the driver should clear CFDD<30>.

Note
Snooze mode should not be used with boot ROM or serial ROM accesses. Also, snooze mode affects the general-purpose timer and the automatic poll demand functions and should not be used in this mode. While in snooze mode, at least 20 bits are required to recognize a preamble for both MII/SYM and RL modes.

6.8 Jabber and Watchdog Timers

The jabber timer monitors the time of each packet transmission. The watchdog timer monitors the time of each packet reception. If a single packet transmission or reception exceeds a programmable value (Section 3.2.2.13), the jabber and watchdog circuitry automatically disables both the transmit and receive path. The transmit jabber timer provides the jabber function by cutting off transmission and asserting the collision signal to the MAC.

The packet descriptor closes with both transmit jabber timeout (TDES0<14>) and late collision (TDES0<9>) setting if the jabber timer expires on a transmit packet.
This chapter describes the interface and operation of the boot ROM, the MicroWire serial ROM, the general purpose port, and the network activity LEDs. This chapter also describes how to connect an external register to the boot ROM port.

### 7.1 Overview

The 21140A provides a boot ROM interface that may be optionally used on the adapter. The boot ROM (expansion ROM) contains a code that can be executed for device-specific initialization and, possibly, a system boot function. During machine boot, the BIOS looks for bootable devices by searching a specific signature (55AA). Once found, the BIOS copies the code from the boot ROM to a shadow RAM in the host memory and executes the code from the RAM.

The boot ROM interface supports:
- 5-V or 12-V flash memory for code upgrade
- 120-ns EEPROM or faster
- Up to 256KB address space

The 21140A provides a software-controlled, serial port interface suitable for MicroWire and other common serial ROM devices. The serial ROM contains the IEEE address and, optionally, other system parameters.

### 7.2 Boot ROM and Serial ROM Connection

Figure 7-1 shows the connection of a 256KB boot ROM and the serial ROM. The two 9-bit edge trigger latches are used to latch the boot ROM addresses <17:2> and the oe_l and we_l control signals.
7.2 Boot ROM and Serial ROM Connection

Figure 7–1 Boot ROM, Serial ROM, and External Register Connection
7.3 Boot ROM Operations

7.3 Boot ROM Operations

Access to the boot ROM is done in two ways:

- Byte access (read/write) using CSR9 and CSR10.
- Dword (32-bit) read access from the PCI expansion ROM address space.

The following sections describe these accesses. For each, the boot ROM must be set to the desired mode (read or write) prior to the actual access for the read or write transaction. For additional information about how this is done, refer to the specific ROM device documentation.

Any mixture between byte access and Dword access is allowed, providing that byte access followed by Dword access will be separated by at least 15 PCI clock cycles.

7.3.1 Byte Read

Figure 7-2 shows the 21140A byte read cycle. It is executed as follows:

1. The host initiates a byte read cycle to the boot ROM by writing the boot ROM offset to CSR10 and by setting a read command in CSR9 (CSR9<14>) and CSR9<12> = 1.

2. The 21140A drives the boot ROM address bits <7:2> and the signals oe_l and we_l on the br_ad lines, drives address bit 17 on the br_a<0> line, and sets br_a<1>. Signal br_a<1> is used as a latch_enable to latch the address, oe_l, and we_l in the upper edge trigger latch.

3. The 21140A clears br_a<1>.

4. The 21140A drives the boot ROM address bits <15:8> on the br_ad lines, drives address bit 16 on the br_a<0> line, and sets br_a<1>. Address bits <16:8> are latched in the upper edge trigger latch while the previous address bits (17, <7:2>) and the control signals (oe_l and we_l) are latched in the lower edge trigger latch.

5. The 21140A drives address bits <1:0> on br_a<1> and br_a<0> respectively and asserts the br_ce_l pin.

6. In response, the boot ROM drives the data on the br_ad lines.

7. The 21140A terminates the byte read cycle by sampling the data, by placing it in CSR9<7:0>, and by deasserting the br_ce_l signal.
7.3 Boot ROM Operations

8. The driver can read the data from CSR9 after at least 20 PCI clock cycles passed since this CSR was previously written. Note that the results of trying to read the data earlier are UNPREDICTABLE.

Figure 7–2 Boot ROM Byte Read Cycle

7.3.2 Byte Write

Before performing a write operation, all the boot ROM entries must be 1. This is achieved by using the erase command.

Figure 7–3 shows the 21140A byte write cycle. It is executed as follows:

1. The host initiates a byte write cycle to the boot ROM by writing the boot ROM offset to CSR10, setting a write command in CSR9 (CSR9<13> and CSR9<12> = 1), and by writing the data to CSR9<7:0>.

2. The 21140A drives the boot ROM address bits <7:2> and the signals oe_l and we_l on the br_ad lines, drives address bit 17 on the br_a<0> line, and sets br_a<1>. Signal br_a<1> is used as a latch_enable to latch the address, oe_l, and we_l in the upper edge trigger latch.

3. The 21140A clears br_a<1>.

4. The 21140A drives the boot ROM address bits <15:8> on the br_ad lines, drives address bit 16 on the br_a<0> line, and sets br_a<1>. Address bits <16:8> are latched in the upper edge trigger latch while the previous address bits (<17>, <7:2>) and the control signals (oe_l and we_l) are latched in the lower edge trigger latch.
7.3 Boot ROM Operations

5. The 21140A drives address bits <1:0> on $\text{br_a<1>}$ and $\text{br_a<0>}$ respectively, drives the data on the $\text{br_ad}$ lines, and asserts the $\text{br_ce_l}$ pin.

6. The boot ROM samples the data.

7. The 21140A terminates the byte write cycle by deasserting the $\text{br_ce_l}$ signal.

Figure 7–3 Boot ROM Byte Write Cycle

7.3.3 Dword Read

Figure 7–4 shows the Dword read cycle. The host initiates a Dword read cycle by executing a typical read cycle to the expansion ROM address space. The $\text{ad}$ lines contain the expansion ROM address (base address and offset). Prior to the assertion of the $\text{trdy}_l$ signal, the 21140A takes the following steps:

1. The 21140A drives the boot ROM address bits <7:2> and the control signals $\text{oe}_l$ and $\text{we}_l$ on the $\text{br_ad}$ lines, drives address bit 17 on the $\text{br_a<0>}$ line, and sets $\text{br_a<1>}$. Signal $\text{br_a<1>}$ is used as a $\text{latch_enable}$ to latch the address, $\text{oe}_l$, and $\text{we}_l$ in the upper edge trigger latch.

2. The 21140A clears $\text{br_a<1>}$.

3. The 21140A drives the boot ROM address bits <15:8> on the $\text{br_ad}$ lines, drives address bit 16 on the $\text{br_ad<0>}$ line, and sets $\text{br_a<1>}$. Address bits <16:8> are latched in the upper edge trigger latch while the previous address bits (17, <7:2>) and the control signals $\text{oe}_l$ and $\text{we}_l$ are latched in the lower edge trigger latch.
7.3 Boot ROM Operations

4. The 21140A remains \( \text{br}_a<1> \) high, drives \( \text{br}_a<0> \) to high, and asserts the \( \text{br}_c e_l \) pin.

5. In response, the boot ROM drives the data on the \( \text{br}_a d \) lines (byte 3).

6. The 21140A samples the data (byte 3).

7. The 21140A remains \( \text{br}_a<1> \) high, drives \( \text{br}_a<0> \) to low, and asserts the \( \text{br}_c e_l \) pin.

8. In response, the boot ROM drives the data on the \( \text{br}_a d \) lines (byte 2).

9. The 21140A samples the data (byte 2).

10. The 21140A drives \( \text{br}_a<1> \) to low, drives \( \text{br}_a<0> \) high, and asserts the \( \text{br}_c e_l \) pin.

11. In response, the boot ROM drives the data on the \( \text{br}_a d \) lines (byte 1).

12. The 21140A samples the data (byte 1).

13. The 21140A remains \( \text{br}_a<1> \) low, drives \( \text{br}_a<0> \) to low, and asserts the \( \text{br}_c e_l \) pin.

14. In response, the boot ROM drives the data on the \( \text{br}_a d \) lines (byte 0).

15. The 21140A samples the data and deasserts the \( \text{br}_c e_l \) signal.

16. The 21140A assembles the 4 bytes, drives the data on the \( \text{ad} \) lines, and asserts \( \text{trd}_y_l \).

Figure 7–4  Boot ROM Dword Read Cycle
7.4 Serial ROM Operations

There are four serial ROM interface pins (Table 3–50):

- Serial ROM data out (CSR9<3>)
- Serial ROM data in (CSR9<2>)
- Serial ROM serial clock (CSR9<1>)
- Serial ROM chip select (CSR9<0>)

All EEPROM access sequences and timing are handled by software. Serial ROM operations include the following: read and write. In addition, the erase EEPROM operation is also supported and is handled similarly to the read and write operations.

7.4.1 Read Operation

Read operations consist of three phases:

1. Command phase—3 bits (binary code of 110)
2. Address phase—6 bits for 256-bit to 1Kb ROMs, 8 bits for 2Kb to 4Kb ROMs.
3. Data phase—16 bits

Figure 7–5 and Figure 7–6 show a typical read cycle that describes the action steps that need to be taken by the driver to execute a read cycle. The timing (for example, 30 ns, 50 ns, and so on) specifies the minimum time that the driver must wait before advancing to the next action.

During both the address phase in Figure 7–5 and the data phase in Figure 7–6, 1 bit is handled during each phase cycle. Therefore, the address phase should be repeated 6 or 8 times depending on the address length and the data phase should be repeated 16 times. Note that the value DX is the current data bit.
7.4 Serial ROM Operations

Figure 7–5  Read Cycle (Page 1 of 2)

1. Write CSR9<2:0> = 000#2
   30 ns

2. Write CSR9<2:0> = 001#2
3. Write CSR9<2:0> = 011#2
4. Write CSR9<2:0> = 001#2
   100 ns

5. Write CSR9<2:0> = 101#2
   150 ns
6. Write CSR9<2:0> = 111#2
7. Write CSR9<2:0> = 101#2
8. Write CSR9<2:0> = 111#2
9. Write CSR9<2:0> = 101#2
10. Write CSR9<2:0> = 001#2
11. Write CSR9<2:0> = 011#2
12. Write CSR9<2:0> = 001#2
   100 ns

13. Write CSR9<2:0> = X01#2
14. Write CSR9<2:0> = X11#2
15. Write CSR9<2:0> = X01#2
   150 ns

X = A0?

Yes

Wait 150 ns

No

Command

X = Current Address Bit

Address

A

LJ-04049.AI
7.4 Serial ROM Operations

Figure 7–6  Read Cycle (Page 2 of 2)

16. Write CSR9<2:0> = 011#2  100 ns
17. Read CSR9<3> = DX  150 ns
18. Write CSR9<2:0> = 001#2  250 ns

DX = D0 ?

No

19. Write CSR9<2:0> = 000#2

Yes

Wait 100 ns Until Ready

End
7.4 Serial ROM Operations

Figure 7–7 shows the read operation timing of the address and data.

Figure 7–7  Read Operation

7.4.2 Write Operation

Write operations consist of three phases:

1. Command phase—3 bits (binary code of 101)
2. Address phase—6 bits for 256-bit to 1Kb ROMs, 8 bits for 2Kb to 4Kb ROMs.
3. Data phase—16 bits

Figure 7–8 and Figure 7–9 show a typical write cycle that describes the action steps that need to be taken by the driver to execute a write cycle. The timing (for example, 30 ns, 50 ns, and so on) specifies the minimum time that the driver must wait before advancing to the next action.

During both the address phase in Figure 7–8 and the data phase in Figure 7–9, 1 bit is handled during each phase cycle. Therefore, the address phase should be repeated 6 or 8 times depending on the address length and the data phase should be repeated 16 times.
7.4 Serial ROM Operations

Figure 7–8 Write Cycle (Page 1 of 2)

1. Write CSR9<2:0> = 000#2  
   30 ns

2. Write CSR9<2:0> = 001#2  
   50 ns
3. Write CSR9<2:0> = 011#2  
   250 ns
4. Write CSR9<2:0> = 001#2  
   100 ns
5. Write CSR9<2:0> = 101#2  
   150 ns
6. Write CSR9<2:0> = 111#2  
   250 ns
7. Write CSR9<2:0> = 101#2  
   100 ns
8. Write CSR9<2:0> = 001#2  
   150 ns
9. Write CSR9<2:0> = 011#2  
   250 ns
10. Write CSR9<2:0> = 001#2 
    100 ns
11. Write CSR9<2:0> = 101#2 
    150 ns
12. Write CSR9<2:0> = 111#2 
    250 ns
13. Write CSR9<2:0> = 101#2 
    100 ns
14. Write CSR9<2:0> = X01#2 
    150 ns
15. Write CSR9<2:0> = X11#2 
    250 ns
16. Write CSR9<2:0> = X01#2 
    100 ns

X = A0?

Yes

A

Command

Address

X = Current Address Bit

X = A0?

No
7.4 Serial ROM Operations

Figure 7–9 Write Cycle (Page 2 of 2)

17. Write CSR9<2:0> = DX01#2 150 ns
18. Write CSR9<2:0> = DX11#2 250 ns
19. Write CSR9<2:0> = DX01#2 100 ns

DX = Current Data Bit

DX = D0?

No

Yes

20. Write CSR9<2:0> = 000#2 250 ns
21. Write CSR9<2:0> = 001#2 250 ns
22. Read CSR9<3>

Wait 1 ms

Busy

CSR9<3> = 1?

No

Yes

23. Write CSR9<2:0> = 000#2

End

Data

DX = Current Data Bit
7.4 Serial ROM Operations

Figure 7–10 shows the write operation timing of the address and data. The time period indicated by \textit{twp} is the actual write cycle time.

![Figure 7–10 Write Operation](image)

7.5 External Register Operation

The 21140A provides the ability to connect an external 8-bit register to the boot ROM port. Figure 7–1 illustrates the signals for this connection.

\textbf{Note}

CSR10 must be 0 before any external register access is done.

To read from the external register, the driver should set the read command (CSR9\textless14\textgreater) and select the external register (CSR9\textless10\textgreater=1). The 21140A performs the same steps as described in Section 7.3.1. The only differences are that now the 21140A drives 1 on both the \textit{we} and \textit{oe} boot ROM inputs and drives 0 on \textit{br_a\textless0}. This, together with the assertion of \textit{br_ce}, performs the actual read operation. The data is sampled by the 21140A and is placed in CSR9\textless7:0\textgreater.
7.5 External Register Operation

To write to the external register, the driver should set the write command (CSR<13>), select the external register (CSR9<10>=1), and write the data to CSR9<7:0>. The 21140A performs the same steps as described in Section 7.3.2. The only differences are that now the 21140A drives 1 on both the we_l and oe_l boot ROM inputs and drives 1 on br_a<0>. This, together with the assertion of br_ce_l, performs the actual write operation.

7.6 General-Purpose Port Register (CSR12)

The 21140A has an 8-pin general-purpose port that can read from or write to pins on external devices. This port is controlled by CSR12 and must be customized for each application.

See Section 3.2.2.12 for more information about CSR12.

7.7 LED Support

The 21140A supports two network event LEDs. Table 7-1 describes the LEDs and pin connections.

Table 7-1 LED Connection

<table>
<thead>
<tr>
<th>Signal</th>
<th>Pin Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rcv_match</td>
<td>122</td>
<td>Receive address match</td>
</tr>
<tr>
<td>sym_link</td>
<td>124</td>
<td>SYM port link</td>
</tr>
</tbody>
</table>
Joint Test Action Group—Test Logic

This appendix describes the joint test action group (JTAG) test logic and the associated registers (instruction, bypass, and boundary scan).

A.1 General Description

JTAG test logic supports testing, observing, and modifying circuit activity during the components normal operation. As a PCI device, the 21140A supports the IEEE standard 1149.1 Test Access Port and Boundary Scan Architecture. The IEEE 1149.1 standard specifies the rules and permissions that govern the design of the 21140A JTAG test logic support. Inclusion of JTAG test logic allows boundary scan to be used to test both the device and the board where it is installed. The JTAG test logic consists of the following four signals to serially interface within the 21140A (Table 2–1):

- **tck** — JTAG clock
- **tdi** — Test data and instructions in
- **tdo** — Test data and instructions out
- **tms** — Test mode select

Note: If JTAG test logic is not implemented, the **tck** pin should be connected to ground, and both the **tms** and **tdi** pins should be connected high. The **tdo** signal should remain unconnected.

These test pins operate in the same electrical environment as the 21140A PCI I/O buffers.

The system vendor is responsible for the design and operation of the 1149.1 serial chains (rings) required in the system. Typically, an 1149.1 ring is created by connecting one device’s **tdo** pin to another device’s **tdi** pin to create a serial chain of devices. In this application, the 21140A receives the same **tck** and **tms** signals as the other devices. The entire 1149.1 ring is connected to
A.1 General Description

either a motherboard test connector for test purposes or to a resident 1149.1 controller.

Note

To understand the description of the 21140A JTAG test logic in this section, the system designer should be familiar with the IEEE 1149.1 standard.

A.2 Registers

In JTAG test logic design, three registers are implemented through the 21140A pads:

- Instruction register
- Bypass register
- Boundary-scan register

A.2.1 Instruction Register

The 21140A JTAG test logic instruction register is a 3-bit (IR<2:0>) scan-type register that is used to direct the JTAG machine to the appropriate operating JTAG mode (Table A–1). Its contents are interpreted as test instructions. The test instructions select the boundary-scan registers for serial transfer of test data by using the \texttt{tdi} and \texttt{tdo} pins. These instructions also control the operation of the selected test features.

Table A–1 Instruction Register

<table>
<thead>
<tr>
<th>IR&lt;2&gt;</th>
<th>IR&lt;1&gt;</th>
<th>IR&lt;0&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>EXTEST mode (mandatory instruction) allows testing of the 21140A board-level interconnections. Test data is shifted into the boundary-scan register of the 21140A and then is transferred in parallel to the output pins.</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Sample-preload mode (mandatory instruction) allows the 21140A JTAG boundary-scan register to be initialized prior to selecting other instructions such as EXTEST. It is also possible to capture data at system pins while the system is running, and to shift that data out for examination.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

(continued on next page)
Table A–1 (Cont.) Instruction Register

<table>
<thead>
<tr>
<th>IR&lt;2&gt;</th>
<th>IR&lt;1&gt;</th>
<th>IR&lt;0&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Reserved.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Reserved.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Tristate mode (optional instruction) allows the 21140A to enter power-saving mode. When this occurs, the PCI and serial ROM port pads are tristated. The MII and SRL ports continue to operate normally without any power reduction.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Continuity mode (optional instruction) allows the 21140A continuity test while in production.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Bypass mode (mandatory instruction) allows the test features on the 21140A J TAG test logic to be bypassed. This instruction selects the bypass register to be connected between tdi and tdo. When the bypass mode is selected, the operation of the test logic has no effect on the operation of the system logic. Bypass mode is selected automatically when power is applied.</td>
</tr>
</tbody>
</table>

A.2.2 Bypass Register

The bypass register is a 1-bit shift register that provides a single-bit serial connection between the tdi and tdo signals when either no other test data register in the 21140A J TAG test logic registers is selected, or the test logic in the 21140A J TAG is bypassed. When power is applied, J TAG test logic resets and then is set to bypass mode.

A.2.3 Boundary-Scan Register

The J TAG boundary-scan register consists of cells located at the PCI and serial ROM pads. This register provides an interconnections test. It also provides additional control and observation of the 21140A pins during the testing phases. For example, the 21140A boundary-scan register can observe the output enable control signals of the I/O pads: ad_oe, cbe_oe, and so on. When these signals are programmed to be 1 during EXTEST mode, data is applied to the output from the selected boundary-scan cells.
A.2 Registers

The following listing contains the order of the boundary-scan register pads:

- tdi -> int_l -> rst_l -> pci_clk -> gnt_l
- req_l -> ad_oe -> ad<31:24> -> cbe_oe -> c_be_l<3>
- idsel -> ad<23:16> -> c_be_l<2> -> frame_oe -> frame_l
- irdy_oe -> irdy_l -> trdy_oe -> trdy_l -> devsel_oe
- devsel_l -> stop_oe -> stop_l -> perr_oe -> perr_l
- serr_l -> par_oe -> par -> c_be_l<1> -> ad<15:8>
- c_be_l<0> -> ad<7:0> -> sr_do -> sr_di -> sr_ck
- sr_cs -> br_ad<0> -> br_ad<1> -> br_ad<2> -> br_ad<3>
- br_ad<4> -> br_ad<5> -> br_ad<6> -> br_ad<7>
- br_ce_l -> br_a<0> -> br_a<1> -> tdo

A.2.4 Test Access Port Controller

The test access port (TAP) controller interprets IEEE P1149.1 protocols received on the tms pin. The TAP controller generates clocks and control signals to control the operation of the test logic. The TAP controller consists of a state machine and control dispatch logic. The 21140A fully implements the TAP state machine as described in the IEEE P1149.1 standard.
This appendix describes the 21140A features that support the driver when implementing and reporting the specified counters and events. CSMA/CD specified events can be reported by the driver based on these features. Table B–1 lists the counters and features.

<table>
<thead>
<tr>
<th>Counter</th>
<th>21140A Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time since creation counter</td>
<td>Supported by the host driver.</td>
</tr>
<tr>
<td>Bytes received</td>
<td>Driver must add the frame length (RDES0&lt;29:16&gt;) fields of all successfully received frames.</td>
</tr>
<tr>
<td>Bytes sent</td>
<td>Driver must add the buffer 1 size (TDES1&lt;10:0&gt;) and buffer 2 size (TDES1&lt;21:11&gt;) fields of all successfully transmitted buffers.</td>
</tr>
<tr>
<td>Frames received</td>
<td>Driver must count the successfully received frames in the receive descriptor list.</td>
</tr>
<tr>
<td>Frames sent</td>
<td>Driver must count the successfully transmitted frames in the transmit descriptor list.</td>
</tr>
<tr>
<td>Multicast bytes received</td>
<td>Driver must add the frame length (RDES0&lt;29:16&gt;) fields of all successfully received frames with multicast frame (RDES0&lt;10&gt;) set.</td>
</tr>
<tr>
<td>Multicast frames received</td>
<td>Driver must count the successfully received frames with multicast frame (RDES&lt;10&gt;) set. (continued on next page)</td>
</tr>
</tbody>
</table>

---

2 Carrier-sense multiple access with collision detection
<table>
<thead>
<tr>
<th>Counter</th>
<th>21140A Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frames sent, initially deferred</td>
<td>Driver must count the successfully transmitted frames when deferred (TDES0&lt;0&gt;) is set.</td>
</tr>
<tr>
<td>Frames sent, single collision</td>
<td>Driver must count the successfully transmitted frames when the collision count (TDES0&lt;6:3&gt;) is equal to 1.</td>
</tr>
<tr>
<td>Frames sent, multiple collisions</td>
<td>Driver must count the successfully transmitted frames when the collision count (TDES0&lt;6:3&gt;) is greater than 1.</td>
</tr>
<tr>
<td>Send failure, excessive collisions</td>
<td>Driver must count the transmit descriptors when the excessive collisions (TDES0&lt;8&gt;) bit is set.</td>
</tr>
<tr>
<td>Send failure, carrier check failed</td>
<td>Driver must count the transmit descriptors when both late collision (TDES0&lt;9&gt;) and loss of carrier (TDES0&lt;11&gt;) are set.</td>
</tr>
<tr>
<td>Send failure, short circuit</td>
<td>There were two successive transmit descriptors when the no_carrier flag (TDES0&lt;10&gt;) is set. This indicates a short circuit.</td>
</tr>
<tr>
<td>Send failure, open circuit</td>
<td>There were two successive transmit descriptors when the excessive_collisions flag (TDES0&lt;8&gt;) is set. This indicates an open circuit.</td>
</tr>
<tr>
<td>Send failure, remote failure to defer</td>
<td>Flagged as a late collision (TDES0&lt;9&gt;) in the transmit descriptors.</td>
</tr>
<tr>
<td>Receive failure, block check error</td>
<td>Driver must count the receive descriptors when CRC error (RDES0&lt;1&gt;) is set and dribbling bit (RDES0&lt;2&gt;) is cleared.</td>
</tr>
<tr>
<td>Receive failure, framing error</td>
<td>Driver must count the receive descriptors when both CRC error (RDES0&lt;1&gt;) and dribbling bit (RDES0&lt;2&gt;) are set.</td>
</tr>
<tr>
<td>Receive failure, frame too long</td>
<td>Driver must count the receive descriptors when frame too long (RDES0&lt;7&gt;) is set.</td>
</tr>
<tr>
<td>Unrecognized frame destination</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>Data overrun</td>
<td>Driver must count the receive descriptors when (RDES0&lt;0&gt;) is set.</td>
</tr>
<tr>
<td>System buffer unavailable</td>
<td>Reported in the missed frame counter CSR8&lt;15:0&gt; (Section 3.2.2.8).</td>
</tr>
<tr>
<td>User buffer unavailable</td>
<td>Maintained by the driver.</td>
</tr>
</tbody>
</table>

(continued on next page)
Table B–1 (Cont.)  CSMA/CD Counters

<table>
<thead>
<tr>
<th>Counter</th>
<th>21140A Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collision detect check failed</td>
<td>Driver must count the transmit descriptors when heartbeat fail (TDES0&lt;7&gt;) is set.</td>
</tr>
</tbody>
</table>
This appendix provides examples of a C routine that generates the hash index for a given Ethernet address. The bit position in the hash table is taken from the CRC32 checksum derived from the first 6 bytes.

There are two C routines that follow: the first is for the little endian architecture and the second is for big endian architecture.

1. Little endian architecture Hash C routine.

```c
#define CRC32_POLY 0xEDB88320UL /* CRC-32 Poly -- Little Endian*/
#define HASH_BITS 9 /* Number of bits in hash */

unsigned crc32_mchash(
    unsigned char *mca)
{
    u_int idx, bit, data, crc = 0xFFFFFFFFUL;
    for (idx = 0; idx < 6; idx++)
        for (data = *mca++, bit = 0; bit < 8; bit++, data >>=1)
            crc = (crc >> 1) ^ (((crc ^ data) & 1) ? CRC32_POLY : 0);
    return crc & ((1 << HASH_BITS) - 1) /* return low bits for hash */
}
```

2. Big endian architecture Hash C routine.

```c
#include <stdio>

unsigned HashIndex (char *Address);
```
main (int argc, char *argv[]) {
    int Index;
    char m[6];
    if (argc < 2) {
        printf("usage: hash xx-xx-xx-xx-xx-xx\n");
        return;
    }
    sscanf(argv[1], "%2X-%2X-%2X-%2X-%2X-%2X",
        &m[0], &m[1], &m[2],
        &m[3], &m[4], &m[5]);
    Index = HashIndex(&m[0]);
    printf("hash_index = %d byte: %d bit: %d\n",
        Index, Index/8, Index%8);
}

unsigned HashIndex (char *Address) {
    unsigned Crc = 0xffffffff;
    unsigned const POLY 0x04c11db6
    unsigned Msb;
    int BytesLength = 6;
    unsigned char CurrentByte;
    unsigned Index;
    int Bit;
    int Shift;
    for (BytesLength=0; BytesLength<6; BytesLength++) {
        CurrentByte = Address[BytesLength];
        for (Bit=0; Bit<8 ; Bit++) {
            Msb = Crc >> 31;
            Crc <<= 1;
            if ( Msb ^ (CurrentByte & 1)) {
                Crc ^= POLY;
                Crc |= 0x00000001;
            }
            CurrentByte >>= 1;
        }
    }
}
/* the hash index is given by the upper 9 bits of the CRC
* taken in decreasing order of significance
* index<0> = crc<31>
* index<1> = crc<30>
* ...
* index<9> = crc<23>
*/

for (Index=0, Bit=23, Shift=8;
     Shift >= 0;
     Bit++, Shift-- ) {
    Index |= ( (Crc>>Bit) & 1 ) << Shift ;
}

return Index;
Technical Support and Ordering Information

D.1 Technical Support

If you need technical support or help deciding which literature best meets your needs, call the Digital Semiconductor Information Line:

United States and Canada  1-800-332-2717
Outside North America    +1-508-628-4760

D.2 Ordering Digital Semiconductor Products

To order the Digital Semiconductor 21140A PCI Fast Ethernet LAN Controller and Evaluation Boards, contact your local distributor.

You can order the following semiconductor product from Digital:

<table>
<thead>
<tr>
<th>Product</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Semiconductor 21140A PCI Fast Ethernet LAN Controller</td>
<td>21140-AC</td>
</tr>
<tr>
<td>Digital Semiconductor 21140A TX Evaluation Board Kit</td>
<td>21A40-TX</td>
</tr>
<tr>
<td>Digital Semiconductor 21140A T4 Evaluation Board Kit</td>
<td>21A40-T4</td>
</tr>
</tbody>
</table>
### D.3 Ordering Associated Literature

The following table lists some of the available Digital Semiconductor literature. For a complete list, contact the Digital Semiconductor Information Line.

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Semiconductor 21140A PCI Fast Ethernet LAN</td>
<td>EC–QN7MB–TE</td>
</tr>
<tr>
<td>Controller Product Brief</td>
<td></td>
</tr>
<tr>
<td>Digital Semiconductor 21140A PCI Fast Ethernet LAN</td>
<td>EC–QN7PB–TE</td>
</tr>
<tr>
<td>Controller Data Sheet</td>
<td></td>
</tr>
</tbody>
</table>

### D.4 Ordering Third-Party Literature

You can order the following third-party literature directly from the vendor:

<table>
<thead>
<tr>
<th>Title</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI Local Bus Specification Revision 2.1</td>
<td>PCI Special Interest Group</td>
</tr>
<tr>
<td></td>
<td>1–800–433–5177 (U.S.)</td>
</tr>
<tr>
<td></td>
<td>1–503–797–4207 (International)</td>
</tr>
<tr>
<td></td>
<td>1–503–234–6762 (FAX)</td>
</tr>
<tr>
<td>PCI BIOS Specification Revision 2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Index

A
Alignment
  error description, 6-19
Alpha
documentation, D-2
Arbitration
  bus, 3-19
  scheme, 4-30
  timing, 5-7
Associated literature, D-2

B
100BASE-FX
  implementation, 6-2
100BASE-T
  definition, 6-1
100BASE-T4
  implementation, 6-1
100BASE-TX
  implementation, 6-1
100BASE-X
  implementation, 6-1
Block diagram, 1-5
Boot ROM
  byte read, 7-3
  byte write, 7-4
  Dword read, 7-5
  interface, 3-42
  overview, 7-1
  pointer, 3-45
  select, 3-44
  Boot ROM, serial ROM, and MII management register
    See CSR9
  Boot ROM programming address register
    See CSR10
  Bus
    commands, 5-2
    error bits, 3-27
    runt frames, 1-1
    unsupported transactions, 5-3
  Bus master
    enabling, 3-5
    operations, 5-6
  Bus mode register
    See CSR0
  Bus slave
    operations, 5-2
  Byte ordering
    big endian, 3-18
    little endian, 3-18

C
Capture effect
  2-0 backoff algorithm, 6-23
  definition, 6-20
  enable, 3-31
  example, 6-21
  resolution, 6-22
  sequence, 6-22
  special enable, 3-30
Carrier-sense multiple access with collision
detection
See CSMA/CD
CBER, 3-11 to 3-12
mapping, 3-12
ROM enable bit, 3-12
CBIO, 3-8 to 3-9
address, 3-9
CBMA, 3-9 to 3-10
address, 3-10
CFCS, 3-3 to 3-6
CFDD, 3-14 to 3-15
CFID, 3-2 to 3-3
CFIT, 3-12 to 3-14
CFLT, 3-7 to 3-8
CFRV, 3-6 to 3-7
Command and status configuration register
See CFCS
Configuration base I/O address register
See CBIO
Configuration base memory address register
See CBMA
Configuration driver area register
See CFDD
Configuration ID register
See CFID
Configuration interrupt register
See CFIT
Configuration latency timer register
See CFLT
Configuration registers
mapping, 3-1
purpose, 3-1
Configuration revision register
SeeCFRV
Control and status registers
See CSRs
CRC
See also Frame check sequence
error, 6-18
CSMA/CD
counters, B-1
CSR0, 3-17 to 3-20
CSR1, 3-20
CSR10, 3-45
CSR11, 3-45 to 3-47
expired, 3-25
value, 3-46
CSR12, 3-47 to 3-48
8-pin implementation, 3-47, 7-14
CSR13
reserved, 3-17
CSR14
reserved, 3-17
CSR15, 3-48 to 3-51
CSR2, 3-20 to 3-21
CSR3, 3-21 to 3-23
CSR4, 3-21 to 3-23
CSR5, 3-23 to 3-28
CSR6, 3-28 to 3-37
CSR7, 3-37 to 3-41
CSR8, 3-41 to 3-42
CSR9, 3-42 to 3-45
CSRs
access to, 3-15
mapping, 3-16
Cyclic redundancy check
See CRC

D
Data communications, 4-1
Descriptor
eroerror, 6-19
list addresses, 3-21
missed frame counter, 3-41
skip length, 3-18
Descriptor list address registers
See CSR3; CSR4
Destination address bit 1, 6-17
Device select
timing, 3-5
DMA
data transfers, 1-3
programmable burst length, 3-18

Index–2
Documentation, D-2
Dribbling bit, 4-7, 6-19

E
Encoder/decoder
See ENDEC
ENDEC
front-end decoder, 1-1
Error
system, 3-4
Ethernet
frame format, 6-8
imperfect filtering, 6-10
inverse filtering, 6-11
perfect filtering, 6-10
promiscuous reception, 6-11
receive addresses, 6-10
Expansion ROM base address register
See CBER
External register
operation, 7-13
select, 3-44

F
Fast transactions
back-to-back, 3-5
FIFO
purpose, 1-3
Frame check sequence
computation, 6-10
Frame format
description, 6-9
Frame too long
description, 6-19
Full-duplex mode, 3-32
Full-duplex operations, 6-20

G
General-purpose port register
See CSR12
General-purpose timer register
See CSR11

H
Hash C routine
examples, C-1
Heartbeat
collision pulse fail, 4-14
Host communication
data buffers, 4-1
descriptor lists, 4-1
descriptor ring and chain structures, 4-1
receive descriptor format, 4-3
transmit descriptor format, 4-11

I
I/O space
access to, 3-6
ID
device, 3-3
manufacturer, 3-3
IEEE 802.3
signals, 6-2
Interpacket gap
See IPG
Interrupt enable register
See CSR7
Interrupts
abnormal, 3-25
eyear receive, 3-25
enabling, 3-37
list, 4-31
multiple events, 4-31
normal, 3-25
pin definition, 3-13
IPG
IPS1
duration, 6-13
IPS2
duration, 6-13
J

Jabber timer
  interval, 3–50
  purpose of, 6–25
Joint test action group
  See JTAG
JTAG
  description, A–1
  instruction register, A–2
  registers, A–2
  boundary scan, A–3
  bypass, A–3
  instruction, A–2

L

Latency timer
  count, 3–8
LED connection, 7–14
Link fail
test fail, 4–14
Literature, D–2
Loopback modes, 6–19
  external, 6–19
  internal, 6–19
Loopback operation mode, 3–35

M

MAC, 6–5 to 6–15
  reception, 6–6
  transmission, 6–5
Media access control
  See MAC
Media-independent interface
  See MII
Memory controller
  terminations, 5–11
Memory space
  access to, 3–4, 3–6
MII
  characteristics, 6–2
  error, 4–7

MII (cont’d)
  location, 6–1
  management, 3–42
MII/SYM
  operating modes, 6–4
  port, 6–1
  port select, 3–31
  signals, 6–2
Missed frame counter register
  See CSR8
Modes
  descriptor byte ordering, 3–18
  filtering, 3–36
  force collision, 3–32
  imperfect address filtering, 3–33
  inverse filtering, 3–33
  pass bad frames, 3–33
  perfect address filtering, 3–34
  promiscuous, 3–33
  sleep, 3–15
  snooze, 3–15
  symbol, 3–30

N

Network events
  LEDs, 7–14

O

Operation mode register
  See CSR6
Ordering products, D–1

P

Packets
  IPG, 6–20
Parity
  disable checking, 3–5
  error, 5–16
  detection, 3–4
  generation, 5–16
  software reset, 3–27
  status, 3–5
Parking, 5–17
Parts
ordering, D-1
PCI
interface functions, 1–3
purpose, 5–1
PCS
implementation, 6–1
support features, 1–1
Peripheral component interconnect
See PCI
PHY layer
timing, 2–6
Physical coding sublayer
See PCS
Physical layer medium dependent
See PMD
Physical medium attachment
See PMA
Pinout diagram, 2–1
PMA, 6–1
PMD, 6–1
Port
data rate, 3–35
Power-saving modes, 6–23
sleep, 3–15
snooze, 3–15
Preamble recognition sequence, 6–17

Receive
data buffer 1 (cont’d)
  byte size, 4–9
data buffer 2
  byte size, 4–9
data type, 4–6
descriptor status validity, 4–11
DMA conditions, 4–30
end of ring, 4–9
error summary, 4–5
first descriptor, 4–6
frame length, 4–5
frame too long, 4–7
frame type, 4–7
last descriptor, 4–6
multicast frame, 4–6
OWN bit, 4–5
process state, 3–28
second address chained, 4–9
start of list, 3–22
Receive descriptor 0
See RDES0
Receive descriptor 1
See RDES1
Receive descriptor 2
See RDES2
Receive descriptor 3
See RDES3
Receive descriptors, 4–2 to 4–10
See also specific receive descriptor entries
Receive poll demand register
See CSR2
Receive process, 4–32 to 4–35
  buffer unavailable, 3–26
descriptor acquisition, 4–33
frame processing, 4–33
start, stop, 3–34
state transitions, 4–34
when suspended, 4–34
window, 4–31
Receiving operation, 6–15 to 6–19
  address matching, 6–17
frame conditions, 6–18
frame decapsulation, 6–17

R
RDES0, 4–3 to 4–8
RDES1, 4–8 to 4–9
RDES2, 4–9
RDES3, 4–10
Read cycle
  configuration, 5–5
  memory, 5–9
  slave, 5–3
Receive
  all packets, 3–30
  buffer 1 address, 4–9
  buffer 2 address, 4–10
  data buffer 1

Index–5
Receiving operation (cont’d)
initiation, 6–16
preamble processing, 6–16
terminating, 6–18
Registers, 3–1 to 3–51
See also specific register entries
Related documentation, D–2
Reset
hardware, 4–29
software, 4–29
Revision number, 3–7
Runt frame
size, 6–19
RxM
receive path, 1–4

S
Semiconductor Information Line, D–1
Serial port
See SRL port
Serial ROM
interface, 3–42
overview, 7–1
read operation, 7–7
select, 3–44
write operation, 7–10
Setup frame
imperfect filtering format, 4–22
perfect filtering format, 4–21
size, 4–19
Shielded twisted-pair
See STP
Signal pins, 2–3
Sleep mode, 3–15, 6–23
Snooze mode, 3–15, 6–24
SRL port, 3–31, 6–1
SSID, 3–10 to 3–11
subsystem ID, 3–11
subsystem vendor ID, 3–11
Startup procedure, 4–32
Status
CSR5, 3–23
Status register
See CSR5
STP, 1–1, 6–1
Subsystem ID register
See SSID
System error
enabling, 3–5

T
Target disconnect
termination, 5–14
Target retry
data transaction, 5–15
TDES0, 4–12 to 4–14
TDES1, 4–14 to 4–17
TDES2, 4–17 to 4–18
TDES3, 4–18
Technical support, D–1
Terminations, 5–11
master abort, 3–4
target abort, 3–4
Third-party documentation, D–2
Transmit
automatic polling, 3–19
buffer 1 address, 4–18
buffer 2 address, 4–18
collision counter, 4–14
CRC disable, 4–16
data buffer 1
byte size, 4–16
data buffer 2
byte size, 4–16
defer, 4–14
descriptor status validity, 4–19
DMA conditions, 4–30
end of ring, 4–16
error summary, 4–13
filtering types, 4–17
first segment, 4–15
interrupt on completion, 4–15
last segment, 4–15
OWN bit, 4–13
padding disable, 4–16
process state, 3–27

Index–6
Transmit (cont’d)
  second address chained, 4–16
  start of list, 3–23
  threshold, 3–34
Transmit descriptor 0
  See TDES0
Transmit descriptor 1
  See TDES1
Transmit descriptor 2
  See TDES2
Transmit descriptor 3
  See TDES3
Transmit descriptors, 4–11 to 4–19
  See also specific transmit descriptor entries
Transmit jabber timer
  timeout, 3–26
Transmit poll demand register
  See CSR1
Transmit process, 4–35 to 4–38
  buffer unavailable, 3–26
  frame processing, 4–36
  polling suspended, 4–36
  state transitions, 4–37
  window, 4–30
Transmit threshold
  10Mb/s or 100Mb/s, 3–30
Transmitting operation, 6–11 to 6–15
  collision, 6–13
  initial deferral, 6–13
  initiation, 6–12
  parameters, 6–15
  termination, 6–14
TxM
  transmit path, 1–3

Watchdog timer register
  See CSR15
Write cycle
  memory, 5–10
  slave, 5–4