Intel 8255x 10/100 Mbps Ethernet Controller Family

Open Source Software Developer Manual

January 2003

Revision 1.0

Information in this document is provided in connection with Intel® products. This specification, the Intel 8255x 10/100 Mbps Ethernet Controller Family Open Source Software Developer Manual, is provided "as is" with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification or sample. Intel products are not intended for use in medical, life saving, life sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® 82557, 82558, 82559, 82550, and 82551 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. It is intended to enable the maintenance of the open source Intel® PRO/100 drivers for the Intel® PRO/100 family of adapters. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

* Other product and corporate names may be trademarks of other companies and are used only for explanation and to the owners' benefit, without intent to infringe.

Contents

1 Introduction 1

1.1 Scope 1
1.2 Document Conventions 2
1.2.1 Device References 2
1.2.2 Numbering 2
1.2.3 Signal Name Representation 2
1.2.4 Memory Alignment Terminology 2

2 Adapter and Controller Overview 5

2.1 Adapter Block Diagram 5
2.2 Intel Fast Ethernet MAC Features 6
2.2.1 82557 Features 6
2.2.2 82558 Features 6
2.2.3 82559, 82550, 82551, and 82562 Features 7
2.3 Working with the Physical Layer 7

3 Power Management Interface 9

3.1 Low Power Mode Requirements 9
3.2 Device Power States 9
3.3 Power Management Registers 9
3.4 Link Operation 10

4 PCI Interface 11

4.1 PCI Configuration Space 11
4.1.1 Vendor ID (Offset 0) 12
4.1.2 Device ID (Offset 2) 12
4.1.3 Command Register (Offset 4) 12
4.1.4 Status Register (Offset 6) 12
4.1.5 Revision (Offset 8) 13
4.1.6 Class Code (Offset 9) 14
4.1.7 Cache Line Size (Offset C) 14
4.1.8 Latency Timer (Offset D) 14
4.1.9 Header Type (Offset E) 14
4.1.10 Built in Self Test (Offset F) 15
4.1.11 Subsystem ID (Offset 2C) 16
4.1.12 Subsystem Vendor ID (Offset 2E) 16
4.1.13 Expansion ROM Base Address Register (Offset 30) 16
4.1.14 The Capabilities Pointer (Offset 34) 17
4.1.15 Interrupt Line (Offset 3C) 17
4.1.16 Interrupt Pin (Offset 3D) 17
4.1.17 Max_Lat / Min_Gnt (Offset 3E) 18
4.1.18 Power Management PCI Configuration Registers 18
4.2 PCI Command Usage 21
4.2.1 Memory Write and Invalidate 22
4.2.2 Read Align 23
4.2.3 Odd Byte Alignment Support 23

5 EEPROM Interface....................25

6 Host Software Interface....................27

6.1 The Shared Memory Architecture....................27
6.2 Initializing the LAN Controller....................29
6.2.1 LAN Controller Addressing Format....................29
6.3 Controlling the Device....................31
6.3.1 Control / Status Registers (CSR)....................31
6.3.2 System Control Block (SCB)....................33
6.3.3 PORT Interface....................42
6.3.4 EEPROM Control Register....................45
6.3.5 Management Data Interface Control Register....................49
6.3.6 Receive Byte Count Register....................51
6.3.7 Early Receive Interrupt....................52
6.3.8 Flow Control Register....................53
6.3.9 Power Management Driver Register....................54
6.3.10 General Control Register....................55
6.3.11 General Status Register....................56
6.4 Shared Memory Structures....................57
6.4.1 Action Commands and Operating Modes....................57
6.4.2 Specific Action Commands....................59
6.4.3 Receive Operation....................97
6.5 Command Unit and Receive Unit Operation....................103
6.5.1 Starting and Completing Control Commands....................103
6.5.2 Generating and acknowledging interrupts....................103
6.5.3 Command Unit Control....................104
6.5.4 Receive Unit Control....................106
6.5.5 Updating SCB Status....................108
6.6 Flow Control....................108
6.6.1 PHY Based Flow Control....................109
6.6.2 Frame Based Flow Control....................109
6.6.3 Priority Aware Frame Based Flow Control....................113
6.6.4 Half Duplex Flow Control....................114
6.7 Collision Backoff Modification in Switched Environments....................114

7 Physical Layer Interface....................115

7.1 Management Data Interface (MDI)....................115
7.2 MDI Register Set....................116
7.2.1 Control Register: Register 0....................117
7.2.2 Status Register: Register 1....................118
7.2.3 Identification Registers: Registers 2 and 3....................118
7.2.4 Auto-Negotiation Advertisement Register: Register 4....................119
7.2.5 Auto-Negotiation Link Partner Ability Register: Register 5....................120
7.2.6 Auto-Negotiation Expansion Register: Register 6....................120
7.3 Intel 82555 Specific Registers....................122
7.3.1 Status and Control Register: Register 16....................122
7.3.2 Special Control Register: Register 17....................123
7.3.3 Clock Synthesis Test and Control Register: Register 18....................124
7.3.4 100BASE-TX Receive False Carrier Counter: Register 19....................124
7.3.5 100Base-TX Receive Disconnect Counter: Register 20....................124

7.3.6 100BASE-TX Receive Error Frame Counter: Register 21 125
7.3.7 Receive Symbol Error Counter: Register 22 125
7.3.8 100BASE-TX Receive EOF Error Counter: Register 23 125
7.3.9 10BASE-T Receive EOF Error Counter: Register 24 125
7.3.10 10BASE-T Transmit Jabber Detect Counter: Register 25 125
7.3.11 Equalizer Control and Status Register: Register 26 126
7.3.12 Special Control Register: Register 27 127
7.4 Auto-Negotiation Functionality 128
7.4.1 Description 128
7.4.2 Parallel Detection 129
7.5 Vendor-Specific PHY Programming 130
7.5.1 Intel 82555 TX PHY 130
7.5.2 82558 and 82559 Embedded PHY Unit 130

8 Programming Recommendations 133

8.1 Adapter Initialization 133
8.1.1 8255x Initialization 133
8.1.2 PHY Detection and Initialization 133
8.1.3 NOS Specific Initialization 134
8.2 Transmit Processing 134
8.3 Frame Reception 134
8.4 Interrupt Processing 135

Appendices

A Wake-up Functionality 137
B 82550 and 82551 Specific Information 153

Figures

1 82557 Network Interface Card Block Diagram 5
2 Command Register 12
3 Command Register 13
4 Cache Line Size 14
5 Base Address Register for Memory Mapping 15
6 Base Address Register for I/O Mapping 15
7 Expansion ROM Base Address Register 17
8 8255x Memory Architecture 28
9 SCB Status Word 34
10 SCB Command Word 36
11 Self-Test Results Format 43
12 EEPROM Control Register 45
13 EEPROM Read Timing Diagram 48
14 General Action Command Format 58
15 NOP Command Format 59
16 Individual Address Setup Command Format 60
17 Configure Command Format 61
18 Multicast Setup Command Format 80
19 Transmit Command Format 81

20 Transmit Buffer Descriptor 83
21 Load Microcode Command Format 88
22 Dump Command Format 89
23 Diagnose Command Format 95
24 Simplified Memory Structure 98
25 Receive Frame Descriptor Format 98
26 Management Frame Structure 116
27 Command Block Structure 144

Tables

1 PCI Configuration Space 11
2 Device and Revision ID 13
3 Base Address Register Summary 16
4 Power Management Capabilities 18
5 Power Management Control/Status Register 20
6 Power Consumption / Dissipation Reporting 21
7 Generated PCI Commands 22
8 Reset Commands 29
9 Device Addressing Formats 30
10 Alignment Requirements for 8255x Data Structures 31
11 Control / Status Register 32
12 System Control Block 34
13 SCB Status Word Bits Descriptions 35
14 SCB Command Word Bits Descriptions 37
15 SCB General Pointer for the CU Command 39
16 Statistical Counters 40
17 Port Register Location 42
18 Port Selection Function 43
19 Dump Wake-up Data Structure 44
20 EEPROM Control Register Locations 45
21 EEPROM Control Register Bits Definitions 45
22 EEPROM Opcode Summary (64-register EEPROM) 46
23 MDI Control Register Location 49
24 Management Data Pins 49
25 MDI Control Register Bits 50
26 Receive Byte Count Register Location 51
27 Early Receive Interrupt Register Location 52
28 Flow Control Registers Location 53
29 Flow Control Threshold Values 54
30 Power Management Driver Register Location 54
31 Power Management Driver Register 55
32 General Control Register Location 55
33 General Control Register 56
34 General Status Register Location 56
35 General Status Register 56
36 Operation Codes 57
37 82557 Configuration Byte Map 62
38 82558 Configuration Byte Map 63
39 82559 Configuration Byte Map 64

40 82557 Dual-Port FIFO Settings 65
41 82558 and 82559 Dual-Port FIFO Settings 66
42 Extended Statistics Functionality 70
43 Pre-amble Length 73
44 82558 B-step Configuration Block ARP Frame IP Address 74
45 82558 B-step ARP Frame IP Address Mapping 75
46 Full Duplex Functionality 77
47 Dump Data Bytes (0-79) 90
48 Dump Data Dwords (20-148) 93
49 RFD Status Bit Descriptions 99
50 Actual Count in Header RFD 100
51 CU Control Commands: Actions at Acceptance Time 105
52 CU Activities Performed at the End of Execution 105
53 RU Control Commands: Actions at Acceptance Time 106
54 Flow Control Frame Format 110
55 Flow Control Configuration Bits 113
56 MDI Register Set 116
57 82555 MDI Register Set 116
58 24-bit OUI Identification Number 119
59 MDI Identification Registers 2 and 3: PHY ID Encoding 119
60 LED Switch Control 128
61 Technology Ability Field Bit Assignments 129
62 Technology Priority 129
63 Fixed Wake-up Configuration Bits 141
64 82559 Port Commands 148
65 Dump Data Structure 149
66 IPCB Structure 153
67 IP Activation Bits (Byte 13) 153
68 IP Activation Bits (Byte 12) 153
69 IPCB Fields 154
70 IPCB Structure Checksum Offload 156
71 IPCB Structure Large Send 160

Revision History

Date	Revision	Description
January 2003	1.0	Initial release.

Introduction 1

This document is intended for use as a software technical reference manual for the Intel® 10/100 Mbps Fast Ethernet controller family, which includes the 82557, 82558, 82559, 82550, and 82551, as well as the 82562 Platform LAN Connect device. It also contains information for several PCI LAN adapters based on these devices: Intel® EtherExpress™ PRO/100+, Intel® EtherExpress™ PRO/100B Wake on LAN (WOL), Intel® EtherExpress™ PRO/100B, and Intel® EtherExpress™ PRO/10+.

1.1 Scope

This manual is intended to be used as a technical reference for software and test engineers developing device drivers or related software for adapters or systems using the Intel® 82557, 82558, 82559, 82550, or 82551 Fast Ethernet controllers or the Intel® 82562 Platform LAN Connect (PLC) device. It contains reference information about the controllers as well as other information that may be required by software developers (such as PHY information, EEPROM contents, PCI scanning, etc.). Since this document uses many examples and contains sample code fragments, it is assumed that the reader has a fundamental understanding of device driver programming and a working knowledge of both C programming language and x86 assembler programming language. Familiarity with at least one industry standard network operating system (NOS) device driver interface (for example, Network Driver Interface Specification [NDIS] or ODI) is also helpful.

The Intel® 10/100 Mbps Fast Ethernet Controller Family includes the following devices in successive order.

Device	Notes
82557	First generation Intel® 10/100 Mbps Fast Ethernet Controller (includes MAC unit only)
82558	Second generation Intel® 10/100 Mbps Fast Ethernet Controller (includes MAC and an integrated PHY unit)
82559	Third generation Intel® 10/100 Mbps Fast Ethernet Controller (includes both a MAC and PHY unit)
82550	Intel® 10/100 Mbps Fast Ethernet Controller (includes both MAC and PHY)
82551	Intel® 10/100 Mbps Fast Ethernet Controller (includes both MAC and PHY)

In general, the Intel family of Fast Ethernet controllers are similar. All family members share the same core hardware and software interface. The later generation components have a higher integration and include support for miscellaneous features (for example, manageability). Since the different generations of Fast Ethernet controllers are highly similar, this manual documents the functionality of all devices and details the differences between the devices. It is intended to be used as a tool to maintain and develop software for all devices in the Intel family of Fast Ethernet controllers.

1.2 Document Conventions

1.2.1 Device References

This document encompasses information for all members of the Intel Fast Ethernet controllers: 82551, 82550, 82559, 82558, 82557 and the 82562.

The document convention, “8255x,” will be used to refer to all devices. In addition, there are specific references to the 82557 throughout this manual that pertains to all 8255x devices. Device-specific differences and exceptions will be documented.

1.2.2 Numbering

Decimal, binary, and hexadecimal numbers are used through the manual. They will be designated as follows:

- **Decimal numbers:** Decimal numbers will not be followed by a suffix.
- **Binary numbers:** Binary numbers (base 2) will be followed by a “b” (for example, 01b).
- **Hexadecimal numbers:** Hexadecimal numbers (base 16) will be followed with the suffix “h” (for example, 1Ch). Hexadecimal numbers may also be noted with a prefix of “0x” (for example, 0x1c).

1.2.3 Signal Name Representation

Signals that are active in a low logic state when asserted are followed by the pound sign (#). For example, FRAME# is asserted low by the master during a transaction. It is asserted low at the start and duration of a transaction and de-asserted during the final data phase.

Signals that are not followed by a pound sign are active in a high logic state when asserted. For example, the IDSEL signal is asserted high when the 82559 during PCI read and write transactions.

1.2.4 Memory Alignment Terminology

The 8255x data structures have special memory alignment requirements. This implies that the starting physical address of a data structure must be aligned as specified. The following terms are used for this purpose:

- **Byte alignment:** Byte alignment implies that the physical addresses can be odd or even.

  Examples: 0FECBD9A1h or 02345ADC6h
- **Word alignment:** Word alignment implies that physical addresses must be aligned on even boundaries. In other words, the last nibble of the address may only end in 0h, 2h, 4h, 6h, 8h, Ah, Ch, or Eh.

  Example: 0FECBD9A2h
- **Dword alignment:** Dword alignment implies that the physical addresses may only be aligned on 4-byte boundaries. In other words, the last nibble of the address may only end in 0h, 4h, 8h, or Ch.

  Example: 0FECBD9A8h

- **Paragraph alignment:** Paragraph alignment implies that the physical addresses may only be aligned on 16-byte boundaries. In other words, the last nibble must be a 0.

  Example: 02345ADC0h

Adapter and Controller Overview 2

Adapters based on an Intel® 8255x device support the ANSI/IEEE 802.3u standard for 100BASE-TX (100 Mbps operation) and 10BASE-T (10 Mbps operation).

2.1 Adapter Block Diagram

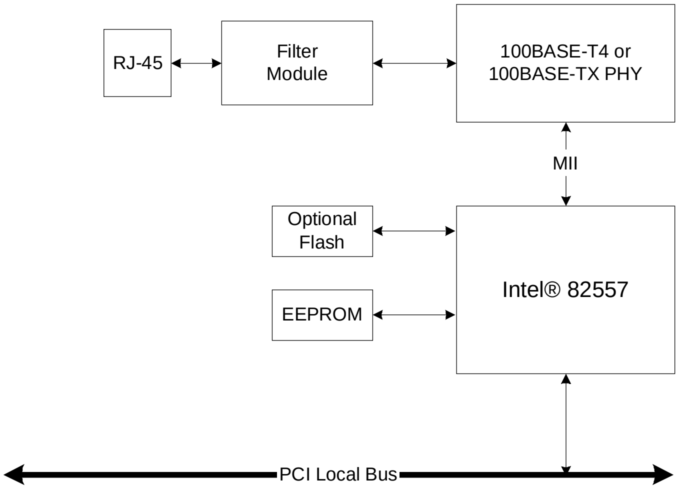
The main components of Intel Fast Ethernet adapters are:

- A Fast Ethernet Media Access Controller (MAC), such as the 8255x, is the core component. The MAC supports the Fast Ethernet ANSI/IEEE 802.3u standard.
- A Physical Layer (PHY) interface device is also required. The 82558, 82559, 82550, and 82551 components have an integrated PHY that supports 100BASE-TX and 10BASE-T. Adapters based on the 82557 must include an appropriate PHY component for their design.
- A serial EEPROM is required to hold the adapter's individual Ethernet node address and other configuration information including fixed PCI configuration parameters.

The adapters are based on 100BASE-TX specifications. 100BASE-TX is a specific scheme designed for use over 2 pairs of Category 5 unshielded twisted-pair cable. 100BASE-TX defines a signaling scheme for 100 Mbps and provides compatibility with the existing 10 Mbps IEEE 802.3 10BASE-T signaling standard. Since only 2-wire pairs are used, TX technology allows full duplex operation at 100 Mbps. The Intel 82555 is one possible TX solution.

The block diagram below illustrates an Intel® PRO/100B adapter configuration based on the 82557 MAC with a TX or T4 PHY.

Figure 1. 82557 Network Interface Card Block Diagram



2.2 Intel Fast Ethernet MAC Features

2.2.1 82557 Features

- Glueless 32-bit, zero wait state PCI bus master interface compliant with PCI Specification, Revision 2.1.
- 10 and 100 Mbps support in compliance with IEEE 802.3 10BASE-T and 802.3u 100BASE-TX.
- Fast back-to-back transmit interframe spacing (IFS) of 960 ns in 100 Mbps networks and 9.6 µs in 10 Mbps networks.
- On-chip Control/Status Register (CSR) incorporating the System Control Block (SCB).
- Simple and flexible packet support with Dynamic transmit chaining.
- Packed Transmit Buffer Descriptors (TBDs).
- Early transmit complete indication.
- Simple receive packet support allows early receive interrupt support for concurrent processing (in simplified mode).
- IEEE Media Independent Interface (MII) compliant PHY interface other MII compliant PHYs.
- Full and half duplex transmit and receive capability.
- Separate on-chip receive and transmit FIFOs.
- On-chip network management counters.
- EEPROM support.
- Optional Flash ROM support (256 Kbytes or 1 Mbyte).

2.2.2 82558 Features

For the most part, the 82558 is a superset of the 82557. In addition to incorporating the features of the 82557, it also includes the following:

- Backward compatible to 82557 software.
- Integrated 100BASE-TX PHY.
- IEEE 802.3u auto-negotiation support in 10BASE-T, 100BASE-TX, full duplex and full duplex flow control configurations.
- Auto-polarity correction for 10BASE-T.
- Optimized PCI interface with support for the memory write and invalidate PCI command.
- Automatic read of EEPROM (programmable ID).
- IEEE 802.3x flow control capable.
- PHY based flow control support when the internal 100BASE-TX PHY is used.
- Advanced Configuration and Power Interface (ACPI) Specification and PCI Power Management Specification compliant.
- Remote power up support (for Magic Packet*).

- Optional Flash support up to 64 Kbytes. (The 82557 is capable of larger Flash size support.)

2.2.3 82559, 82550, 82551, and 82562 Features

The 82559, 82550, and 82551 devices are supersets of the 82557 and 82558. However, the 82559 does not support PHY based flow control as the 82558 did. The new 82559 features are:

- Backward compatible to the 82557 and 82558 software.
- Low power 3.3 V device:
- Clockrun protocol support.
- System Management Bus (SMB) support.
- Wired for Management support (WfM).
- Expanded Wake on LAN capabilities.
- 128 Kbytes Flash size support. (The 82558 only supported a 64 Kbyte Flash.)
- Thin ball grid array (BGA) 15 x 15 mm package.

2.2.3.1 82559ER Features

The 82559ER is a member of the 82559 Fast Ethernet controllers. It is a subset of the 82559. However, the 82559ER does not support:

- SMB.
- Wake on Magic Packet*.

2.3 Working with the Physical Layer

The 82557 contains an IEEE MII compliant interface to a MII compliant PHY, allowing connections to 10/100 Mbps networks. Software communicates to a MII compliant device through the 82557 by using the its Management Data Interface (MDI) port.

The 82558, 82559, 82550 and 82551 contain an embedded PHY module. Although the PHY is internal for these devices, software still communicates to the PHY unit through the MDI port.

For 10/100 Mbps connections, the 82557 can be used in conjunction with the Intel® 82555. For 10 Mbps only connections, the 82557 can be interfaced to the Intel® 82503 serial interface, while maintaining software compatibility to 100 Mbps solutions. The 82558 and later devices do not have a 10 Mbps only interface as the 82557. However, it is possible to interface these devices with a 10 Mbps only MII device.

Power Management Interface 3

The 82557 has no power management support. The 82558 added support for the Advanced Configuration and Power Interface (ACPI) Specification and limited support for Wake on LAN (WOL). The 82558 B-step upgraded and expanded the WOL capability, while the 82559 expanded and simplified the WOL functionality even more.

3.1 Low Power Mode Requirements

The 82558, 82559, 82550, and 82551 adhere to the emerging power management standards as defined in:

- PCI Bus Power Management Interface Specification, Revision 1.0.
- Advanced Configuration and Power Interface Specification (ACPI), Rev 1.0; December 22, 1996.
- Device Class Power Management Reference Specification - Network Device Class, Revision 1.0.

These three specifications define how a PCI network device can be controlled in an OS Directed Power Management (OSPM) environment. These devices all adhere to these specifications. Additionally, they support bus isolation within the chip and Wake on LAN (WOL) capabilities.

3.2 Device Power States

Currently, operating systems only support the D0 and D3 power states. However, starting with the 82558, the Intel Fast Ethernet controller family supports all four power states as defined in the PCI Power Management Specification. These power states are named D0, D1, D2 and D3. D0 is the maximum powered state, and D3, the minimum powered state.

3.3 Power Management Registers

The 82558, 82559, 82550, and 82551 support power management registers:

- Power Management Capability Pointer (Cap_Ptr)
- Power Management Capabilities (PMC)
- Power Management Control/Status Register (PMCSR)
- Power Management Driver Register (PMDR)

The first three registers are located in PCI configuration space and are defined in the PCI Power Management Specification. It is part of the device CSR, which is mapped into system memory and I/O space.

3.4 Link Operation

In the D0 state, the device maintains an active link. The 82558 B-step (refer to Table 2, “Device and Revision ID” on page 13) and later devices also maintain an active link in the D3 state if PME is enabled and the device has power. This implies:

- **10BASE-T Mode:** The device expects a normal clock input on the X1 and X2 pins. It expects to receive normal reception on the Receive Differential Positive and Receive Differential Negative signals (RDP/RDN pair). The device will not transmit on the Transmit Differential Positive and Transmit Differential Negative signals (TDP/TDN pair).
- **100BASE-TX Mode:** The device expects a normal clock input on the X1 and X2 pins and to receive normal reception on the RDP/RDN pair. It transmits a continuous idle stream on the TDP/TDN pair, as required by the 100BASE-TX standard. The 82558 does not transmit frames on the link.
- **Auto-Negotiation:** If the link fails while the device is in the D1 state, it performs the normal auto-negotiation protocol in order to re-establish the link. For the 82558 B-step, if the link fails in the D3 state and PME is enabled and the device has power, the device will attempt to use the normal auto-negotiation protocol in order to re-establish the link. If the link fails on the 82559 in the D3 state and PME is enabled and the device has power, the 82559 will go into a deep power down state, rather than trying to re-establish the link with the auto-negotiation protocol.

During the D3 power state, the 82558 A-step does not maintain an active link. The 82558 B-step and later generation devices do not maintain a link in D3 if PME is disabled or if the device does not have power.

PCI Interface 4

4.1 PCI Configuration Space

One of the most important functions for enabling superior configurability and ease of use is the ability to relocate PCI devices in the address spaces. By default PCI devices support “Plug and Play.” When the system is powered on, device independent software (usually the system BIOS) determines present devices, builds an address map, and assigns non-conflicting resources to those devices. The device independent software accomplishes this configuration task by writing to the PCI configuration space of each individual PCI device.

The 8255x supports 16 Dwords of Type 0 Configuration Space Header, as defined in the PCI Specification, Revision 2.1. The 82259 and 82558 also support a small section in the device specific configuration space. The configuration space is depicted below. The registers that are not identical between the devices are shaded.

Table 1. PCI Configuration Space

Byte Offset (hexadecimal)	Byte 3	Byte 2	Byte 1	Byte 0
0	Device ID		Vendor ID	
4	Status Register		Command Register	
8	Class Code (200000h)			Revision ID
C	BIST	Header Type	Latency Timer	Cache Line Size
10	CSR Memory Mapped Base Address Register			
14	CSR I/O Mapped Base Address Register			
18	Flash Memory Mapped Base Address Register			
1C	Reserved			
20				
24				
28				
2C	Subsystem ID		Subsystem Vendor ID	
30	Expansion ROM Base Address Register			
34	Reserved			Cap_Ptr
38	Reserved			
3C	Max_Latency (FFh)	Min_Grant (FFh)	Interrupt Pin (01h)	Interrupt Line
DC	Power Management Capabilities		Next Item Pointer	Capability ID
E0	Reserved	Data	Power Management CSR	

4.1.1 Vendor ID (Offset 0)

This field identifies the device manufacturer. For the 82557 B-step this field equals 8086h. For the 82557 C-Step, 82558, and 82559, this field is automatically loaded from the EEPROM at power on or upon the assertion of PCI reset. If the EEPROM is not present or invalid, this value defaults to 8086h.

4.1.2 Device ID (Offset 2)

This field uniquely identifies the device. For the 82557 B-step this field is 1229h. For the 82557 C-Step, 82558, and 82559, this field is automatically loaded from the EEPROM at power on or upon the assertion of PCI reset. If the EEPROM is not present or invalid, this value defaults to 1229h for the 82558 and 82559. The 82559ER does not load the Device ID from the EEPROM and will always equal 1209h.

4.1.3 Command Register (Offset 4)

The Command Register provides control over the device's ability to generate and respond to CPU cycles. Its layout is shown below. The shaded bits are not used and are hard-wired to 0.

Figure 2. Command Register

15 10	9 0
Reserved	Command Bits

Bits	Initial Value	Description
15:10	0	Reserved.
9	0	Fast back-to-back enable.
8	x	SERR# enable.
7	0	Wait cycle enable.
6	x	Parity error response
5	0	Palette snoop enable.
4	x	Memory write and invalidate (MWI) enable. **NOTE:** More information regarding the MWI command is located in Section 4.2.1, "Memory Write and Invalidate".
3	0	Special cycle monitoring.
2	x	Mastering enable.
1	x	Memory access enable.
0	x	I/O access enable.

4.1.4 Status Register (Offset 6)

The Status Register is used to record status information for PCI bus related events. Its layout is shown below. The shaded bits are not used and are hard-wired to 0.

Figure 3. Command Register

15 4	3 0
Status Bits	Reserved

Bits	Initial Value	Description
15	x	Detected parity error.
14	x	Signaled system error.
13	x	Received master abort.
12	x	Received target abort.
11	0	Signaled target abort.
10:9	01	DEVSEL timing (indicates minimum timing).
8	x	Data parity reported.
7	1	Fast back-to-back capable.
6	0	UDF supported.
5	0	66 MHz capable.
4	1 (82559 and 82558) 0 (82557)	Capabilities list. This bit indicates whether the device implements a list of new capabilities such as PCI Power Management. If it is set, the Cap_Ptr register in the PCI Configuration Space points to the location of the first item in the Capabilities List. **NOTE:** This bit is set to 1 for the 82559 and 82558 if it is not disabled by the EEPROM. It is always equal to 0 for the 82557.
3:0	0	Reserved.

4.1.5 Revision (Offset 8)

This register specifies a device specific revision identifier. For the 82557 C-Step, 82558, and 82559, this field may be automatically loaded from the EEPROM at power on or upon the assertion of a PCI reset. The default revision register values for the various devices are:

Table 2. Device and Revision ID

Device	Revision ID	PCI Revision Supported	Intel Driver Supported
82557 A-Step	01h	2.0	Yes
82557 B-Step	02h	2.0	Yes
82557 C-Step	03h	2.1	No
82558 A-Step	04h	2.1	Yes
82558 B-Step	05h	2.1	Yes
82559 A-Step	06h	2.1	No
82559 B-Step	07h	2.1	No
82559 C-Step	08h	2.2	Yes

Table 2. Device and Revision ID

Device	Revision ID	PCI Revision Supported	Intel Driver Supported
82559ER A-Step	09h	2.2	Yes
82550	0Ch, 0Dh, 0Eh	2.2	Yes
82551	0Fh, 10h	2.2	Yes

4.1.6 Class Code (Offset 9)

The class code, 020000h, identifies the device as an Ethernet adapter.

4.1.7 Cache Line Size (Offset C)

This register specifies the system cache line size in units of 32-bit words and can be read or written to. The system BIOS or OS should initialize this register at power on or after a PCI reset.

The 82557 does not support Memory Write and Invalidate (MWI) and therefore returns 0 when this register is read. The 82258 and 82559 support the MW I command and must support this register. The 82558 and 82559 can only support cache line sizes of 8 and 16 Dwords. Any value other than 8 or 16 written to the register is ignored, and the device does not use the MWI command. If a value other than 8 or 16 is written into the Cache Line Size (CLS) register, the device returns all zeroes when the CLS register is read.

Figure 4. Cache Line Size

7	6	5	4	3	2	1	0
0	0	0	RW	RW	0	0	0

Bit 3 is set to 1 only if the value 00001000b (8) is written to this register. Bit 4 is set to 1 only if the value 00010000b (16) is written to this register. All other bits are read only and will return 0 on read.

4.1.8 Latency Timer (Offset D)

This register specifies, in units of PCI bus clocks, the minimum time that a bus master can retain ownership of the bus. This value is set by the PCI bus arbitrator based on the values in the maximum latency (Max_Lat) and Maximum Grant (Max_Gnt) registers.

4.1.9 Header Type (Offset E)

This byte field identifies the layout of the second part of the predefined configuration space header and if the device is a multi-function component. The 82557 and 82558 are both single function devices and have this register hard-coded to 00h. For the 82559, the value of this register is determined by a bit in the EEPROM. This register should read 00h for a standard Ethernet adapter, 00h.

4.1.10 Built in Self Test (Offset F)

This optional register is used for control and status of Built in Self Test (BIST). This register is hard-wired to 0 indicating that the devices do not support BIST.

Three base address registers are supported by the 8255x:

- CSR Memory Mapped Base Address Register (BAR 0 at offset 10)
- CSR I/O Mapped Base Address Register (BAR 1 at offset 14)
- Flash Memory Mapped Base Address Register (BAR 2 at offset 18)

Two request memory mapped resources, and the third, I/O mapping. Each register is 32 bits wide. The least significant bit in each base address register determines whether it represents an I/O or memory space. The figures below illustrate layouts for I/O and memory mapped base address registers. After determining which resources will be used, the power-up software maps the I/O and memory controllers into available locations and continues with the power up. To perform the mapping in a device independent manner, the base registers are placed in the predefined header portion of configuration space. Device drivers access this configuration space to determine the mapping of a particular device.

Figure 5. Base Address Register for Memory Mapping

32 4	3 0
Base Address	Configuration Bits

Bits	Initial Value	Description
31:4	x	Base Address.
3	x	Pre-fetchable.
2:1	x	00 = Locate address anywhere in 32-bit address space. 01 = Locate address below 1 MByte. 10 = Locate address anywhere in 64-bit address space. 11 = Reserved.
0	0	Memory space indicator.

NOTE: Bit 0 in all base registers is read-only and used to determine whether the register maps into memory or I/O space. Base registers mapping to memory space must return a 0 in bit 0, and base registers mapping to I/O space, a 1.

Figure 6. Base Address Register for I/O Mapping

32 2	1	0
Base Address	Reserved	1

NOTE: Base registers that map into I/O space are always 32 bits with bit 0 hard wired to a 1, bit 1 is reserved and must return 0 on reads, and the other bits are used to map the device into I/O space.

The number of upper bits that a device actually implements depends on how much of the address space the device responds to. A device that wants a 1 Mbyte memory address space would set the most significant 12 bits of the base address register to be configurable, setting the other bits to 0.

The 8255x requires one BAR for I/O mapping and one BAR for memory mapping of these registers anywhere within the 32-bit memory address space. The driver determines which BAR (I/O or Memory) is used to access the Control/Status Registers. However, both are always requested by the device.

One BAR is also required to map the accesses to an optional Flash memory. The 82557 implements this register regardless of the presence or absence of a Flash chip on the adapter. The 82558 and 82559 only implement this register if a bit is set in the EEPROM. The size of the space requested by this register is 1Mbyte, and it is always mapped anywhere in the 32-bit memory address space.

Note: Although the 82558 only supports up to 64 Kbytes of Flash memory and the 82559 only supports 128 Kbytes of Flash memory, 1 Mbyte of address space is still requested. Software should not access Flash addresses above 64 Kbytes for the 82558 or 128 Kbytes for the 82559 because Flash accesses above the limits are aliased to lower addresses. Table 3 describes the implementation of the base address registers.

Table 3. Base Address Register Summary

Register Location	Description
10h	Memory space for the device Control/Status Registers. The size of this space is 4 Kbytes and it is mapped anywhere in the 32-bit memory address space. It is marked as pre-fetchable. Software should not assume that this memory will be granted below 1 Mbyte.
14h	I/O space for the device Control/Status Registers. The size of this space is 32 bytes.
18h	Memory space for FLASH buffer accesses. The size of this space is 1Mbyte. It is mapped anywhere in the 32-bit address space and is not pre-fetchable.
1Ch - 27h	Reserved.

4.1.11 Subsystem ID (Offset 2C)

This register uniquely identifies the add-in adapter or subsystem where the PCI device resides. It provides a mechanism to distinguish different adapters that use the same PCI controller. For the 82557 B-step this field equals 0000h. For the 82557 C-Step and later devices, this field is loaded from the EEPROM at power on or upon the assertion of PCI reset. If the EEPROM is not present or invalid, this value defaults to 0000h.

4.1.12 Subsystem Vendor ID (Offset 2E)

This register uniquely identifies the add-in adapter or subsystem where the PCI device resides. It provides a mechanism to distinguish the vendor of a adapter from the vendor of the PCI controller used on the adapter. For the 82557 B-step this field is 0000h. For the 82557 C-Step and later devices, this field is automatically loaded from the EEPROM at power on or upon the assertion of PCI reset. If the EEPROM is not present or invalid, this value defaults to 0000h.

4.1.13 Expansion ROM Base Address Register (Offset 30)

The 8255x provides an interface to a local Flash device (or EEPROM) which may be used as an expansion ROM. A 32-bit Expansion ROM Base Address Register at offset 30h in the PCI Configuration Space is defined to handle the address and size information for boot-time access to the Flash. The 82557 implements this register regardless of the presence or absence of a Flash component on the adapter. For the 82558 and later Fast Ethernet controllers, this register is only implemented if a bit is set in the EEPROM.

The register functions exactly like a 32-bit base address register except that the encoding (and usage) of the bottom bits is different. The upper 21 bits correspond to the upper 21 bits of the expansion ROM base address. The 8255x only allow an expansion ROM to be mapped on a 1 Mbyte boundary. Therefore, only the most significant 12 bits are configurable to indicate the 1 Mbyte size requirement (as with the Flash Memory Mapped BAR, the 82558 and 82559 request a 1 Mbyte mapping even though the maximum Flash size allowed with those devices is 65 Kbytes). The format of the register is shown in the figure below.

Figure 7. Expansion ROM Base Address Register

32				20	19	1	0
Read / Write					Reserved (all bits set to 0)		En

Bit 0 in the register is used to control whether or not the device accepts accesses to its expansion ROM. When this bit is reset, the devices expansion ROM address space is disabled. This bit is programmed at initialization time by the system BIOS. The Memory Space bit in the Command register has precedence over the Expansion ROM Base Address Enable bit. A device responds to accesses to its expansion ROM only if both the Memory Space bit and the Expansion ROM Base Address Enable bit are set to 1 (it is reset to 0 upon PCI reset).

4.1.14 The Capabilities Pointer (Offset 34)

This an 8-bit field that provides an offset in the device PCI Configuration Space for the location of the first item in the Capabilities Linked List. The Power Management Interface documentation specifies this linked list to provide access to all appropriate device information in the implementation of the ACPI.

For the 82257, this register is hard-wired to 0 since it does not support power management.

For the 82558 this register is set to DCh if power management is enabled in the EEPROM. If power management is disabled, then this register is set to 0.

For the 82559 and later Intel Fast Ethernet controllers, this register is hard-wired to DCh.

4.1.15 Interrupt Line (Offset 3C)

The Interrupt Line register is an 8-bit register used to communicate interrupt line routing information. This register is configurable by the system BIOS or OS. POST software writes the routing information into this register as it initializes and configures the system. The value in this register specifies which system interrupt controller input the device interrupt pin is connected to. Device drivers and operating systems use this information to determine priority and vector information.

4.1.16 Interrupt Pin (Offset 3D)

The Interrupt Pin register specifies which interrupt pin the device (or device function) uses. This register is always set to a 1, indicating that INTA# is used.

4.1.17 Max_Lat / Min_Gnt (Offset 3E)

These registers specify the device settings for Latency Timer values. For both registers, the value specifies a period of time in units of ¼ microsecond. Min_Gnt is used to specify the burst length period the device needs assuming a clock rate of 33 MHz. Max_Lat is used to specify how often the device needs to gain access to the PCI bus. The values of these registers are 8h (2 µS) for Min_Gnt and 18h (6 µS) for Max_Lat.

4.1.18 Power Management PCI Configuration Registers

4.1.18.1 Capability Identifier (Offset DC)

The Capability Identifier signals this item in the capability linked list as the PCI Power Management registers. The PCI Power Management registers have been assigned the ID of 01h. Since power management is not implemented in the 82557, this register is hard-coded to 0 for that device. For the 82558 and later devices, this read only register returns 01h.

4.1.18.2 Next Item Pointer (Offset DD)

The Next Item Pointer register describes the location of the next item in the capability list. Since power management is the last item in the list, this register is set to 0.

4.1.18.3 Power Management Capabilities (Offset DE)

The Power Management Capabilities (PMC) register is a 16-bit read-only register, which provides information on the capabilities of the device related to power management. Since power management is not implemented in the 82557, this register is hard-coded to 0 for that device. For the 82558 and later devices, this register returns values according to the chart below.

Table 4. Power Management Capabilities

Bit	Default Value	R / W	Description
31:27	82558A: 00011 82558B, 82559: no auxiliary power - 01111 with auxiliary power - 11111	RO	**PME_Support.** This five bit field indicates the power states in which the device may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state. The 82558 A-step supports wake-up from D0 and D1. The 82558 B-step and 82559 support wake-up from D0, D1, D2 and $D3_{hot}$ if no auxiliary power is present and from all power states if auxiliary power exists.
26	1	RO	**D2_Support.** If this bit is set, this function supports the D2 Power Management State. All devices must support the D0 and D3 states. The 82559 and later devices support the D2 Power Management State.
25	1	RO	**D1_Support.** If this bit is set, this function supports the D1 Power Management State. The 82558 and later devices supports the D1 Power Management State.
24	82558A: 1 82558B: 0 82559: 0	RO	**FullClk.** If this bit is set, this function requires a full speed clock at all times when it is in the D0 state in order to perform its function. If this bit is cleared, the function only requires a full speed PCI clock while actually transferring data so dynamic clock control may be used. The 82558 A-step requires a full speed clock at all times when it is in the D0 state in order to perform its function.

Table 4. Power Management Capabilities

Bit	Default Value	R / W	Description
23	0	RO	**Reserved.** This field is not used by the 8255x.
22	82558: 0 82559: no auxiliary power - 0 with auxiliary power - 1	RO	**AUX_Current.** If the device is connected to an auxiliary power supply, the 82559 reports a “1” to indicate that it consume less than 250 mA from the 3.3 Vaux pin while in the D3$_{cold}$ state. This bit is a reflection of bit 31.
21	1	RO	**DSI.** The Device Specific Initialization bit indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. Device specific initialization is required for the 82558 and 82559 after a D3 to D0 transition.
20	82558A: 0 82558B, 82559: no auxiliary power - 0 with auxiliary power - 1	RO	**Auxiliary Power Source** This bit is only meaningful if PMCSR bit 31 (D3$_{cold}$ supporting PME) equals 1. When this bit also equals 1, it indicates that support for PME# in D3$_{cold}$ requires an auxiliary power supply. The 82558 B-step and 82559 require auxiliary power for wake up from the D3$_{cold}$ state. Therefore this bit is set to 1 if auxiliary power is present.
19	0	RO	**PME Clock.** When this bit is 1, it indicates that the PME# generation logic requires its host PCI bus to maintain a free-running PCI clock. When this bit is 0, it indicates that no host bus clock is required for the function to generate PME#. The 82558 and later generation devices do not require a clock to generate PME# and return 0.
81:16	001	RO	**Version.** This field specifies to software how to interpret the PMC and PMCSR registers. A value of 001b indicates that the device complies with the Revision 1.0 of the PCI Power Management Interface Specification.

4.1.18.4 Power Management Control/Status (Offset E0)

The Power Management Control/Status Register (PMCSR) is used to determine and change the current power state of the device. It also allows for the control of the power management interrupts in a standard way. Since power management is not implemented in the 82557, this register is hard-coded to 0 for that device. For the 82558 and later devices this register acts according to the chart below.

Table 5. Power Management Control/Status Register

Bit	Value at Reset	R / W	Description
15	82558A: 0 82558B, 82559: no auxiliary power - 0 Sticky bit	Read Clear	**PME Status.** This bit is set upon a wake-up event from the link. It is independent of the state of the PME_Enable bit. When software writes 1 to this bit it is cleared and the device stops asserting PME# (if enabled).
14:13	82558: 00 82559: 10 or 00	RO	**Data Scale.** The Data Scale is not supported on the 82558 and always returns 0. For the 82559, it is a 2-bit read-only field indicating the data register scaling factor. For the 82559, it equals 10b for registers 0 through 8 and 00b for registers 9 through 15.
12:9	0000	R/W	**Data Select.** This 4-bit field selects which data is reported through the Data Register and Data Scale field. This register is only supported on the 82559 and later generation devices.
8	82558A = 0 82558B & 82559 = 0 Unknown (0 if no auxiliary power available)	Read Clear	**PME Enable.** This bit enables the device to assert PME#.
7:5	000		**Reserved.**
4	0	RO	**Dynamic Data.** The 82558 does not implement this register and returns 0. The 82559 does not support the ability to monitor power consumption dynamically.
3:2	00	RO	**Reserved.**
01:00	00	R/W	**Power State.** This 2-bit field is used both to determine the current power state of the 82258 or 82559 and to set the 82558 or 82559 into a new power state. The definition of the field values is given below. 00b - D0 01b - D1 10b - D2 11b - D3

While wake-up events are not allowed in the D0 power state, hardware does not automatically preclude this functionality. To ensure that wake-up events are not generated when in D0, software must clear the PME Enable bit when putting the device into that state. To ensure that no spurious wake-up events are generated by the function, the PME Status bit (in the PMDR register or the PMCSR) must be specifically cleared (by writing a 1) when the PME Enable bit is set.

To support Wake on LAN mode (pre-boot wake), the PME Enable and PME Status bits are set with known values after power-up reset. The ALTRST# pin should be connected to the device auxiliary power good signal on the motherboard so that it will be active low on system power up. Assertion of ALTRST# clears the PME Status bit and sets the PME Enable bit if the clock is active on the CLK pin. Thus, if the Wake on LAN (WOL) bit in the EEPROM is set, the device will wake up the system upon receiving of Magic Packet*.

4.1.18.5 Ethernet Power Consumption Registers (Offset E2h)

The Data Register is an 8-bit read-only register providing a mechanism for the device to report state dependent maximum power consumption and heat dissipation. The value reported in this register depends on the value written to the Data Select field in the PMCSR register.

The power measurements defined in the register have a dynamic range of 0 to 2.55 W with 0.01 W resolution according to the data scale.

Note: The required accuracy should be in the range of +20% and -10%. The 82557 and 82558 do not implement this register. The 82559 and later Intel Fast Ethernet controllers do. The value reported in this register is hard-coded in the 82559 silicon. The structure of the data register is presented below:

Table 6. Power Consumption / Dissipation Reporting

Data Select	Data Scale	Data Reported
0	2	D0 Power Consumption = 58 (580 mW)
1	2	D1 Power Consumption = 40 (400 mW)
2	2	D2 Power Consumption = 40 (400 mW)
3	2	D3 Power Consumption = 40 (400 mW)
4	2	D0 Power Dissipated = 58 (580 mW)
5	2	D1 Power Dissipated = 40 (400 mW)
6	2	D2 Power Dissipated = 40 (400 mW)
7	2	D3 Power Dissipated = 40 (400 mW)
8	2	Common Function Power Dissipated = 00
9-15	0	Reserved 00 h

NOTE: The D1 and D2 power states are not currently supported by operating systems.

4.2 PCI Command Usage

The table below lists the PCI commands that the various Intel Fast Ethernet controllers can use.

Table 7. Generated PCI Commands

PCI Command	Name	Circumstance Used
0x6	MR	TxCB “S” bit read.
0x7	MW	CB and RFD. Writing statistics counters or dump data buffer to memory. Writing received packet data into receive buffers.
0xC	MRM	Reading transmit data buffers.
0xE	MRL	CB, TBD, and RFD.
0xF	MWI (82558 & 82559)	Writing received packet data into receive buffers.

The controllers do not generate I/O commands, Interrupt Acknowledge cycles, or Configuration cycles. The controllers also do not support Dual Address Cycle (DAC). Targets (typically the system bridge) must respond to all of the commands that the Ethernet controller generates.

4.2.1 Memory Write and Invalidate

The 82558, 82559, 82550, and 82551 have 4 internal DMA channels. Of these 4, the Receive DMA channel is used to deposit packet data received from the link into system memory. The Receive DMA channel uses both the Memory Write (MW) and the Memory Write and Invalidate (MWI) commands. In order to use MWI the device must guarantee:

- A minimum transfer of one cache line.
- All byte-enable bits are active during each MWI access.
- The device may cross a cache line boundary only if it intends to transfer the entire next cache line too.

In order to ensure the above conditions, the device may use the MWI command only if the following conditions hold:

1. The cache line size written in the CLS register during PCI configuration is 8 or 16 Dwords.
2. The accessed address is cache line aligned.
3. The 82558 or 82559 has at least a cache line size of data byte in its Receive FIFO.
4. There is at least a cache line size of space left in the system memory buffer. In addition, the device will use two configuration bits to enable and disable the use of MWI:
    a. MWI Enable bit in the PCI Configuration Command register (Section 4.1, “PCI Configuration Space”).
    b. MWI Enable bit in the device Configure command (Section 6.4.2.3, “Configure (010b)”).

If any one of these conditions does not hold, the device uses the MW command. If a MWI cycle is started and one of the conditions does not hold any more (for example, the data space in the memory buffer is less than the CLS), then the device terminates the MWI cycle at the end of the cache line. The next cycle is a MWI or MW cycle according to the conditions listed above.

If a MWI cycle is terminated by a Retry from the target, the device attempts to retry the access using the MWI command. If a MWI cycle is terminated in the middle of a cache line by a disconnect from the target (including Disconnect-C), the device issues a new cycle from the disconnected point using the MW command. If the disconnect occurs on a cache line boundary, the device may start the next cycle using either MW or MWI according to the conditions listed above. If the PCI latency timer or the 82558 (or later generation device) arbitration counter expires during a MWI cycle, the device continues the cycle until the end of the cache line.

If the device started a MW cycle and reaches a cache line boundary, it either terminates the cycle or continues according to the Term on CL configuration bit (Section 6.4.2.3, “Configure (010b)”). If the Term on CL bit is set, the device terminates the MW cycle and attempts to start a new cycle. The new cycle is a MWI cycle if all conditions are met. If the bit is not set, the device continues the MW cycle across the cache line boundary if required.

4.2.2 Read Align

The Read Align feature is aimed to enhance performance in cache line oriented systems. Starting a PCI transaction in these systems on a non-cache line aligned address may result in low performance.

To solve this performance problem, the controller can be configured to terminate Transmit DMA cycles on a cache line boundary, and start the next transaction on a cache line aligned address. This feature is enabled when the Read Align Enable bit is set in device Configure command (Section 6.4.2.3, “Configure (010b)”).

If this bit is set, the device operates as follows:

- When the device is close to running out of resources on the Transmit DMA (in other words, the Transmit FIFO is almost full), it attempts to terminate the read transaction on the nearest cache line boundary when possible.
- When the arbitration counters feature is enabled (maximum Transmit DMA byte count value is set in configuration space), the device switches to other pending DMAs on cache line boundary only.

4.2.3 Odd Byte Alignment Support

Various data structures have special memory alignment requirements. These alignment requirements are detailed in Section 6.2.1, “LAN Controller Addressing Format”.

EEPROM Interface 5

The 8255x has a local memory interface that provides access to a serial EEPROM and optional Flash device. All controllers implement these interfaces using multiplexed pins. Since the interface uses multiplexed pins, it is not simultaneously available to software. Thus, software cannot read the EEPROM at the same time as it is reading Flash memory. However, software can certainly read the EEPROM and then read Flash memory or vice versa.

The Serial EEPROM stores configuration data (such as the Ethernet MAC address) for the 8255x. The EEPROM is a serial in and serial out device. The 82557 and 82558 support a single size of EEPROM that contains 64 registers of 16 bits per register. The 82559 and later generation devices support either a 64 register EEPROM or a 256 register EEPROM.

Software may read or write to the EEPROM by accessing the EEPROM port in the 8255x.

All accesses, read or write, are preceded by a command instruction to the device. The command instructions, begin with a logical 1 as a start bit, two opcode bits (indicating read, write, erase, etc.), and n-bits of address. The address field is 6 bits for a 64-register EEPROM and 8 bits for a 256-register EEPROM. The end of the address field is indicated by a “dummy 0” bit from the EEPROM, which indicates the entire address field has been transferred to the device. A command is issued by asserting the chip select signal and clocking the data into the EEPROM on its data input pin relative to the serial clock input. The chip select signal is de-asserted after completion of the EEPROM cycle (Command, Address and Data).

The 8255x performs an automatic read of several registers in the EEPROM following the de-assertion of the PCI Reset signal. The controllers automatically read the EEPROM to properly set several power-on default configurations. Since the 82559 and later devices are capable of interfacing with different size EEPROMs (64 or 256 words), software determine the EEPROM size first using the “dummy zero mechanism” before it accesses the EEPROM after a reset.

Host Software Interface 6

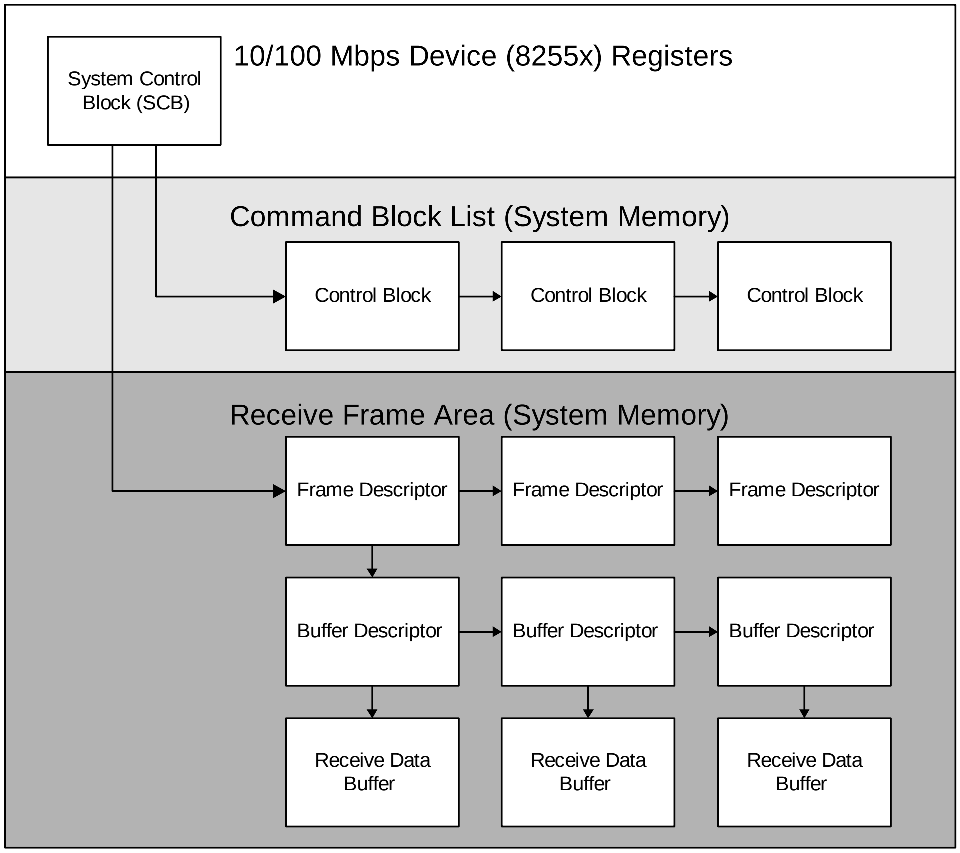
The 8255x LAN controllers establish a shared memory communication system with the host CPU. Software controls the device by writing and reading data to and from this shared memory space. All of the LAN controller functions (configuration, transmitting data, receiving data, etc.) that are software manageable are controlled through this shared memory space.

Note: Although references are made to both simplified and flexible memory modes for transmit and receive commands, only the simplified mode is supported. All bit settings and silicon configurations only refer to the simplified memory mode.

6.1 The Shared Memory Architecture

The shared memory structure is divided into three parts: the Control/Status Registers (CSR), the Command Block List (CBL), and the Receive Frame Area (RFA). The CSR physically resides on the LAN controller and can be accessed by either I/O or memory cycles, while the rest of the memory structures reside in system (host) memory. The first 8 bytes of the CSR is called the System Control Block (SCB). The SCB serves as a central communication point for exchanging control and status information between the host CPU and the 8255x. The host software controls the state of the Command Unit (CU) and Receive Unit (RU) (for example, active, suspended or idle) by writing commands to the SCB. The device posts the status of the CU and RU in the SCB Status word and indicates status changes with an interrupt. The SCB also holds pointers to a linked list of action commands called the CBL and a linked list of receive resources called the RFA. This type of structure is shown in the figure below.

Figure 8. 8255x Memory Architecture



The CBL consists of a linked list of individual action commands in structures called Command Blocks (CBs). The CBs contain command parameters and status of the action commands. Action commands are categorized as follows:

- Non-transmit (non-Tx) commands: This category includes commands such as no operation (NOP), Configure, IA Setup, Multicast Setup, Dump and Diagnose.
- Transmit (Tx) command: This includes Transmit Command Blocks (TxCB).

The Receive Frame Area (RFA) consists of a list of Receive Frame Descriptors (RFDs) and a list of user-prepared or NOS provided buffers. The receive architecture supports the simplified memory model similar to the way it is supported by the transmit command. In the simplified memory model, the data buffer immediately follows the RFD. The receive structures format and receive code flow is described in Section 6.4.3.1, “Receive Frame Area” and Section 6.4.3.4, “No Buffer Performance Improvements (82558 and 82559)”.

The LAN controller also provides read and write access to an external EEPROM and the Management Data Interface (MDI) registers. This is achieved through the EEPROM Control Register and the MDI Control Register, respectively. These registers occupy offsets 0Ch through 14h of the CSR.

6.2 Initializing the LAN Controller

A hardware or software reset prepares the 8255x for normal operation. Since the PCI Specification already provides automatic configuration of many critical parameters such as I/O, memory mapping and interrupt assignment, the device is set to an operational default state after reset. However, the device cannot transmit or receive frames until a Configure command is issued. Different reset commands affect the controller in different ways as detailed by the table below.

Table 8. Reset Commands

Reset Operation	Effect on LAN Controller
Hardware reset. This occurs when the Reset pin (RST#) is asserted. (This is caused by turning the system on or by pressing the system reset button.)	Resets all internal registers. A full initialization sequence is needed to make the device operational.
Software reset. (This is issued as Port Reset command.)	Resets all internal registers, except the PCI configuration registers. A full initialization sequence is needed to make the device operational.
Selective reset. (This is issued as Port Selective Reset command.)	Maintains PCI configuration, RU and CU base registers, HDS size, error counters, configure, IA setup and multicast setup command information. RU and CU are set to the idle state. All other setup and configuration information is lost.
Self test. (This is issued as a Port Self Test command.)	Resets all internal registers except for the PCI configuration registers. A full initialization sequence is needed to make the device operational. A selective reset is issued internally before the command is executed. A software reset is issued internally after the command is completed.

The phrase “Software Reset” will be used throughout this manual to indicate a complete reset using the Port Reset command, unless specified otherwise. Port commands are discussed in detail in Section 6.3.3, “PORT Interface”.

6.2.1 LAN Controller Addressing Format

The 8255x supports a 32-bit enhanced linear addressing mode and 32-bit segmented addressing mode. The 8255x accommodates both types of addressing schemes with the enhanced linear addressing mode. The controller always calculates a physical address by adding the appropriate 32-bit BAR (CU base or RU base) to a 32-bit offset. This allows a linear addressing scheme to be used by setting the base address to zero and using the full 32-bit offset registers to indicate the linear address. A 32-bit segmented scheme can be used as well by programming the appropriate 32-bit base address register and using the lower 16 bits of the 32-bit offset. This is illustrated in the table below.

Table 9. Device Addressing Formats

Points to	Base Register	32-bit Offset Pointer	Physical Address
Start of Command Block List (CBL)	CU Base (32-bit)	SCB General Pointer	Base (32) + Offset (32)
Start of Receive Frame Area (RFA)	RU Base (32-bit)	SCB General Pointer	Base (32) + Offset (32)
Next Command Block (CB)	CU Base (32-bit)	Link Address in CB	Base (32) + Offset (32)
Start of TBD Array	CU Base (32-bit)	TBD Array Address in TxCB	Base (32) + Offset (32)
Next Receive Frame Descriptor (RFD)	RU Base (32-bit)	Link Address in RFD	Base (32) + Offset (32)
TX Buffer	No Base Register	Transmit Buffer #n Address in TBD Array	Offset (32) (Physical address)
Dump Buffer (Dump CB)	CU Base (32-bit)	Buffer Address in CB	Base (32) + Offset (32)
Port Dump / Self-Test	No Base Register	Port Address	Offset (32) (Physical address)
Dump Counters	No Base Register	SCB General Pointer	Offset (32) (Physical address)

To support linear addressing, the device should be programmed as follows:

- Load a value of 00000000h into the CU base using the Load CU Base Address SCB command.
- Load a value of 00000000h into the RU base using the Load RU Base Address SCB command.
- Use the offset pointer values in the various data structures as absolute 32-bit linear addresses.

To support 32-bit segmented addressing, the device should be programmed as follows:

- Load the desired segment value into the CU base using the Load CU Base Address SCB command.
- Load the desired segment value into the RU Base using the Load RU Base Address SCB command.
- Use the offset pointer values in the various data structures as 16-bit offsets. Software must ensure that the upper 16 bits of this offset equal 0000h as the device will add all 32 bits of the base and offset values.

Note: The Load CU Base and the Load RU Base commands can only be executed when the CU and RU are in the idle state. Issuing these commands when the CU or RU is not idle is prohibited.

As mentioned earlier, the 8255x data structures have special memory alignment requirements. The table below lists these requirements. Most of the structures listed in the table will be discussed in much greater detail in subsequent sections.

Table 10. Alignment Requirements for 8255x Data Structures

Data Structure	Alignment Requirements
Port Self-Test	Paragraph aligned (16-byte)
Port Dump	Paragraph aligned (16-byte)
CSR and SCB	Address allocated by the BIOS. No other alignment requirements.
TxCB (buffer of TxCB in simplified mode)	Word (even address) aligned (2-byte aligned). However, Dword (4-byte aligned) structures are more efficient.
TBD	Word (even address) aligned (2-byte aligned). However, Dword (4-byte aligned) structures are more efficient.
Transmit Buffer (flexible mode only)	Byte aligned (address can be odd or even).
RFD (buffer of RFD in simplified mode)	Word (even address) aligned (2-byte aligned). However, Dword (4-byte aligned) structures are more efficient. **NOTE:** In an MWI aware system, for best performance RFDs should be allocated so that the RFD data area (if not zero) is cache line aligned.

As the table above indicates, the 8255x have the same alignment restrictions with one exception: The 82558, and 82559 have a limited capability to support odd byte aligned buffers.

6.3 Controlling the Device

Software issues control commands to the CU and RU through the SCB, which is part of the CSR. The CPU instructs the device to activate, suspend, resume or idle the CU or RU by placing the appropriate control command in the CU or RU control field. A CPU write access to the SCB causes the device to read the SCB, including the Status word, Command word, CU and RU Control fields, and the SCB General Pointer. Activating the CU causes the device to start executing the CBL. When execution is completed the device updates the SCB with the CU status then interrupts the CPU if it is configured. Activating the RU causes the device to access the RFA and go into the ready state for frame reception. When a frame is received the RU updates the SCB with the RU status and interrupts the CPU. It also automatically advances to the next free RFD in the RFA. This interaction between the CPU and the device can continue until a software or selective reset is issued to the device, at which point the initialization process must be executed again. The CPU can also perform certain controller functions directly through a CPU port interface.

6.3.1 Control / Status Registers (CSR)

The Control/Status Registers make up the CSR space. The basic registers are the SCB Command word, SCB Status word, SCB General Pointer, Port interface, EEPROM Control register, and MDI Control register. Additionally, the 82558 and later devices also contain registers for flow control, power management, etc. All of these registers are shown in the table below. Registers new to the 82558 are lightly shaded, and registers new to the 82559 (at offset 1Ch and beyond) are darkly shaded. Accessing these higher offset areas in older devices has an unpredictable effect and may cause errors.

Table 11. Control / Status Register

<table>
<tr><th colspan="2">Upper Word</th><th colspan="2">Lower Word</th><th rowspan="2">Offset</th></tr>
<tr><th>31</th><th>16</th><th>15</th><th>0</th></tr>
<tr><td colspan="2">SCB Command Word</td><td colspan="2">SCB Status Word</td><td>0h</td></tr>
<tr><td colspan="4">SCB General Pointer</td><td>4h</td></tr>
<tr><td colspan="4">PORT</td><td>8h</td></tr>
<tr><td colspan="2">EEPROM Control Register</td><td colspan="2">Reserved</td><td>Ch</td></tr>
<tr><td colspan="4">MDI Control Register</td><td>10h</td></tr>
<tr><td colspan="4">RX DMA Byte Count</td><td>14h</td></tr>
<tr><td>PMDR</td><td colspan="2">Flow Control Register</td><td>Reserved</td><td>18h</td></tr>
<tr><td colspan="2">Reserved</td><td>General Status</td><td>General Control</td><td>1Ch</td></tr>
<tr><td colspan="4">Reserved</td><td>20h-2Ch</td></tr>
<tr><td colspan="4">Function Event Register</td><td>30h</td></tr>
<tr><td colspan="4">Function Event Mask Register</td><td>34h</td></tr>
<tr><td colspan="4">Function Present State Register</td><td>38h</td></tr>
<tr><td colspan="4">Force Event Register</td><td>3Ch</td></tr>
</table>

- SCB Command Word. This register is where software writes commands for the CU and RU.
- SCB Status Word. The device places the CU and RU status for the CPU to read in this word.
- SCB General Pointer. The SCB General Pointer points to various data structures in main memory depending on the current SCB Command word.
- Port Interface. This special interface allows the CPU to reset the device and force it to dump information to main memory or perform an internal self test.
- EEPROM Control Register. The EEPROM Control Register allows the CPU to read and write to an external EEPROM.
- MDI Control Register. This register allows the CPU to read and write information from Physical Layer components through the Management Data Interface.
- Early Receive Interrupt Rx Byte Count (RXBC) Register. This register allows the CPU to read the current value in the Receive DMA byte count register. The Receive DMA byte count register tracks the number of receive data bytes that have been placed into host memory.
- Flow Control Threshold Register. This register allows the driver to set the flow control threshold value. (This register is not included in the 82557.)
- Flow Control Command Register. This register allows the driver to indicate flow control commands to the 82558 and later devices.
- Power Management Driver Register (PMDR). This register indicates power management events to the driver.

The CSR can be accessed as either an I/O mapped or memory mapped PCI slave.

Note: The PCI Configuration space Base Address Registers (BARs) automatically request memory space for the CSR and I/O space for the CSR. Software may use either memory mapped or I/O mapped mode or even use them interchangeably. In most environments, memory mapped mode is the preferred method of accessing the CSR. Some bridges may not properly transfer data in memory mapped mode and it may be necessary to have an I/O backup method if the memory method does not work.

Note: All fields in the CSR are byte, word, or Dword addressable. Accesses to the CSR, especially the SCB command and status word, should be limited to byte-wide operations to avoid and side effects. For example, issuing a new command through the CU, only the lower byte of the CSR command word should be accessed (byte 2 of the CSR). This will prevent any accidental modification of the interrupt mask or software interrupt bits that occupy the upper byte of the command word.

6.3.2 System Control Block (SCB)

The SCB plays a major role in communications between the CPU and the LAN controller. Commands issued by the software and status reported by the device are placed in the SCB. The SCB is part of the CSR and represents the first two Dwords of that structure.

Control commands are issued to the device by writing them into the SCB. This causes the device to examine the command, clear the lower byte of the SCB command word (indicating command acceptance), and perform the required action. Control commands perform the following types of tasks:

- Operate the Command Unit (CU). The SCB controls the CU by specifying the address of the Command Block List (CBL) and by starting or resuming execution of CBL commands.
- Operate the Receive Unit (RU). The SCB controls RU frame reception by specifying the address of the Receive Frame Area (RFA) and by starting, resuming, or aborting frame reception.
- Load the dump counters address.
- Command the device to dump or dump and reset its internal statistical counters.

  In a similar manner, the CPU can send Interrupt Acknowledgments to the device by writing them into the Interrupt Acknowledge byte (upper byte of the SCB Status word).

  The device also indicates status to the CPU through bits in the SCB Status word such as CU status and RU status.
- Indicate the cause of the current interrupt(s). Interrupts are caused by one or more of the following events:
  - — The CU will interrupt the CPU when it completes an action command that has its interrupt bit set (CX Interrupt).
  - — The CU will interrupt the CPU either when it leaves the active state (CNA Interrupt) or when it enters the idle state (CI Interrupt), depending on how the device is configured.
  - — The CU will interrupt the CPU when it completes a transmit command with a bad status (TNO Interrupt) if it is configured.
  - — The RU will interrupt the CPU when it receives either a complete frame or a predefined part of it (FR Interrupt or ER Interrupt for the 82558 and 82559 devices).
  - — The RU is not ready (RNR Interrupt).
  - — A previously initiated read or write cycle to the MDI has been completed (MDI Interrupt).
  - — Software has requested an interrupt (SWI Interrupt).

— A flow control pause frame was received (FCP Interrupt). This does not apply to the 82557.

Note: TNO interrupts should be avoided. Protocol stacks automatically retry failed transmits. This feature should only be enabled if software needs to know immediately about transmit failures.

Interrupt events can only be cleared by CPU acknowledgment. In other words, if the device has asserted its interrupt pin, the only way to clear it is with a CPU Acknowledgment of that particular interrupt bit in the SCB. Since multiple events could be active simultaneously, if some events are not acknowledged by the ACK field, the interrupt signal will remain asserted. However, if a new event occurs while an interrupt is set, it will not cause an additional interrupt.

The table below shows the SCB format. It is followed by a detailed description on the SCB bits and their functions.

Table 12. System Control Block

31 16	15 0	
Upper Word	**Lower Word**	**Offset**
SCB Command Word	SCB Status Word	Base + 00h
SCB General Pointer		Base + 04h

6.3.2.1 SCB Status Word

Figure 9. SCB Status Word

15 8	7 6	5 2	1	0
STAT / ACK	CUS	RUS	0	0

The SCB Status word is addressable as two bytes. The upper byte is called the STAT/ACK byte, and the lower, the SCB Status byte. The SCB Status byte indicates the status of the CU and RU. The STAT/ACK byte reports the device status as bits, which represent the causes of interrupts. Writing to the STAT/ACK bits will acknowledge pending interrupts. As described below, there are many different possible interrupt events. The LAN controller asserts the interrupt line to the CPU if any of these interrupt events need to be serviced. More than one STAT/ACK bits may be set at the same time. Writing 1 back to a STAT/ACK bit that was set will acknowledge that particular interrupt bit. The device will de-assert its interrupt line only when all pending interrupt STAT bits are acknowledged. All pending STAT bits do not need to be acknowledged in a single access, but it is recommended if the interrupt service routine is likely to process all pending interrupts.

Note: The LAN controller latches interrupts internally. Interrupts are PCI compliant and level-triggered. Setting a 1 in the interrupt acknowledge command for a non-pending interrupt does not cause any malfunctions. It is simply ignored by the device. Also, any 0 bits in the interrupt acknowledge command have no effect, whether the interrupt is pending or not.

Table 13. SCB Status Word Bits Descriptions

Bit	Symbol	Description
Bit 15	CX/TNO	This bit indicates that the CU finished executing a command with its interrupt bit set. The 82557 includes a TNO feature where the device can be configured to assert this interrupt when a transmit command is completed with a status of not okay. The TNO interrupt feature is not available in the 82558 or later devices.
Bit 14	FR	This bit indicates that the RU has finished receiving a frame or the header portion of a frame if the device is in header RFD mode.
Bit 13	CNA	This bit indicates when the CU has left the active state or has entered the idle state. There are 2 distinct states of the CU. When the device is configured to generate CNA interrupt, the interrupt is activated when the CU leaves the active state and enters either the idle or suspended state. When the device is configured to generate CI interrupt, an interrupt will be generated only when the CU enters the idle state.
Bit 12	RNR	This bit indicates when the RU leaves the ready state. The RU may leave the ready state due to an RU Abort command or because there are no available resources or if the RU filled an RFD with its suspend bit set.
Bit 11	MDI	This bit indicates when an MDI read or write cycle has completed. This interrupt only occurs if it is enabled through the interrupt enable bit (bit 29) in the MDI Control Register of the CSR.
Bit 10	SWI	This bit is used for software generated interrupts. In some cases, software may need to generate an interrupt to re-enter the ISR.
Bit 9	Reserved	This bit is reserved and should not be used.
Bit 8	FCP	This bit is used for flow control pause interrupt. It is present in the 82558 and later devices. This bit is not used on the 82557 and should be treated as a reserved bit.

Table 13. SCB Status Word Bits Descriptions

Bit	Symbol	Description
Bits 7:6	CUS	This field contains the CU status (2 bits). Valid values are for this field are: 00 Idle 01 Suspended 10 LPQ Active 11 HQP Active
Bits 5:2	RUS	This field contains the RU status (4 bits). Valid values are: 0000 Idle 0001 Suspended 0010 No resources 0011 Reserved 0100 Ready 0101 Reserved 0110 Reserved 0111 Reserved 1000 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved
Bits 1:0	Reserved	These bits are reserved and should not be used.

Note: The SCB Status word is not updated immediately in response to SCB commands. For example, the CU status will remain in the idle state for a period of time after the CU start command is issued. Software should not rely exclusively on the state of the SCB Status word to determine when it is appropriate to issue commands requiring the device to be in a specific state. Software may be required to keep an internal state engine on the commands recently issued to the device to insure that data read from the register is valid.

6.3.2.2 SCB Command Word

Figure 10. SCB Command Word

31 26	25	24	23 20	19	18 16
Specific Interrupt Mask Bits	SI	M	CU Command	0	RU Command

The SCB Command word is also addressable as two bytes. The upper byte is called the Interrupt Control byte. The least significant byte is called the Command byte.

The Interrupt Control byte allows software to either force the generation of an interrupt or mask device interrupts from occurring. The 82558 and later devices also allow individual interrupt sources from within the device to be masked (this feature is not available in the 82557).

When software wants to issue an action command, it should write to the Command byte. The CUC and RUC fields of the Command byte specify the actions to be performed by the 8255x. The command is ready for acceptance by the device as soon as it is written into the CUC or RUC field. The actual command execution may not start instantaneously and will depend on current receive and transmit DMA activity. The Command byte is set by the CPU and cleared by the 8255x indicating command acceptance.

Table 14. SCB Command Word Bits Descriptions

Bit	Symbol	Description
Bits 31:26	Specific Interrupt Mask Bits	The mask bits range from bit 31 to 26. Writing a 1 to a mask bit disables the 8255x (except the 82557) from generating an interrupt, or asserting the INTA# pin, due to that corresponding event. The device may still generate interrupts due to other interrupt events that are not masked. The corresponding bits and their masks are: 31 - CX Mask 30 - FR Mask 29 - CNA Mask 28 - RNR Mask 27 - ER Mask 26 - FCP Mask These bits are also described in Section 6.3.2, “System Control Block (SCB)”. These bits are not present in the 82557 and should be treated as reserved.
Bit 25	SI	This bit is used for the software generated interrupt. Writing a 1 to this bit causes the device to generate an interrupt, and writing a 0 has no effect. Reads from this bit always return a zero. The M bit (bit 24) has higher precedence than the SI bit. Thus, if a 1 is simultaneously written to both, no interrupts occur.
Bit 24	M	This bit is used as the interrupt mask bit. When this bit is set to 1, the device does not assert its INTA# pin (PCI interrupt pin). The M bit has higher precedence than bits 31 through 26 of this word and the SI bit (bit 25).

Table 14. SCB Command Word Bits Descriptions

Bit	Symbol	Description
Bits 23:20	CUC	This field contains the CU Command. Valid values for this field are: 0000 NOP. The no operation command does not affect the current state of the unit. 0001 CU Start. CU Start begins execution of the first command on the CBL. A pointer to the first CB of the CBL should be placed in the SCB General Pointer before issuing this command. **NOTE:** The CU Start command should only be issued when the CU is in the idle or suspended states (never when the CU is in the active state) and all of the previously issued CBs have been processed and completed by the CU. Sometimes, it is only possible to determine that all CBs are completed by checking the complete bit in all previously issued Command Blocks. 0010 CU Resume. The CU Resume command resumes CU operation by executing the next command. If the CU is Idle, it ignores the CU Resume command. 0100 Load Dump Counters Address. This command directs the device where to write dump data when the Dump Statistical Counters or Dump and Reset Statistical Counters command is used. It must be executed at least once before the Dump Statistical Counters or Dump and Reset Statistical Counters command is used. The address of the dump area must be placed in the general pointer register. 0101 Dump Statistical Counters. This command directs the device to dump its statistical counters to the area designated by the Load Dump Counters Address command. 0110 Load CU Base. The internal CU Base Register is loaded with the value in the SCB General Pointer. 0111 Dump and Reset Statistical Counters. This command directs the device to first dump its statistical counters to the area designated by the Load Dump Counters Address command and then to clear these counters. 1010 CU Static Resume. It resumes CU operation by executing the next command. If the CU is idle, it will ignore the CU Resume command. This command should be used only when the device CU is in the suspended state and has no pending CU Resume commands. This command is only valid for the 82558 and later devices. It is not valid for the 82557.
Bit 19	Reserved	This bit is reserved and should be set to 0.

Table 14. SCB Command Word Bits Descriptions

Bit	Symbol	Description
Bits 18:16	RUC	This field contains the RU Command. Valid values are: 000 NOP. NOP is a no operation command and does not alter current state of unit. 001 RU Start. RU Start enables the receive unit. The pointer to the RFA must be placed in the SCB General Pointer before using this command. The device prefetches the first RFD in preparation of receiving incoming frames that pass its address filtering. 010 RU Resume. The RU Resume command resumes frame reception (only when in suspended state). 011 Receive DMA Redirect. This command is only valid for the 82558 and later devices. The buffers are indicated by an RBD chain, which is pointed to by an offset stored in the general pointer register (in the RU base). 100 RU Abort. The RU Abort command immediately stops RU receive operation. 101 Load Header Data Size (HDS). After a load HDS command is issued, the device expects to only find header RFDs or to be used in Receive DMA mode until it is reset. This value defines the size of the header portion of the RFDs or receive buffers. The HDS value is defined by the lower 14 bits of the SCB General Pointer; thus, bits 15 through 31 should always be set to zeros when using this command. The value of HDS should be an even non-zero number. 110 Load RU Base. The internal RU Base Register is loaded with the value that was placed in the SCB General Pointer just before this command was issued.

6.3.2.3 SCB General Pointer

The SCB General Pointer is a 32-bit entity, which points to various data structures depending on the command in the CUC or RUC field. The two tables below indicate what the SCB pointer means for the different commands.

Table 15. SCB General Pointer for the CU Command

RUC Field	RU Command	SCB General Pointer	Added to
0	NOP	Don’t care	
1	RU Start	Pointer to first RFD in the Receive Frame Area	RU Base
2	RU Resume	Don’t care	
3	Reserved	Don’t care	
4	RU Abort	Don’t care	
5	Load HDS	Header Data Size (Upper 18 bits must be zero)	
6	Load RU Base	32-bit Base Register for RU data structures	

6.3.2.4 Statistical Counters

The 8255x provides information for network management by providing on-chip statistical counters that track a variety of events associated with both transmit and receive. The counters are updated by the device when it completes the processing of a frame. For example, after the completion of transmitting a frame on the link or when receiving a frame, the counter is updated. The Statistical Counters are reported to the software on demand by issuing the Dump Statistical Counters command or the Dump and Reset Statistical Counters command in the SCB CUC field. The counters are internal to the device and are listed in the table below.

Table 16. Statistical Counters

Byte Offset	Device Statistic
0	**Transmit good frames.** This counter contains the number of frames transmitted properly on the link. It is updated only after the actual transmission on the link is completed and not when the frame was read from memory as is done for the TxCB status.
4	**Transmit maximum collisions (MAXCOL) errors.** This counter contains the number of frames that were not transmitted because they encountered the configured maximum number of collisions. This counter should only increment when the network is heavily saturated with traffic.
8	**Transmit late collisions (LATECOL) errors.** This counter contains the number of frames that were not transmitted since they encountered a collision outside of the normal collision window.
12	**Transmit underrun errors.** This counter contains the number of frames that were either not transmitted or retransmitted due to a transmit DMA underrun. If the device is configured to retransmit on underrun, this counter may be updated multiple times for a single frame. Underruns occur due to a lack of PCI bandwidth resulting in the internal transmit FIFO running dry during the transmission of a frame.
16	**Transmit lost carrier sense (CRS).** This counter contains the number of frames transmitted by the device despite the fact that it detected the de-assertion of CRS during the transmission.
20	**Transmit deferred.** This counter contains the number of frames that were deferred before transmission due to activity on the link.
24	**Transmit single collision.** This counter contains the number of transmitted frames that encountered only one collision.
28	**Transmit multiple collisions.** This counter contains the number of transmitted frames that encountered more than one collision.
32	**Transmit total collisions.** This counter contains the total number of collisions that were encountered while attempting to transmit. This count includes late collisions and collisions from frames that encountered maximum collisions.
36	**Receive good frames.** This counter contains the number of frames that were received properly from the link. It is updated only after the actual reception from the link is completed and all data bytes are stored in memory.
40	**Receive CRC errors.** This counter contains the number of aligned frames discarded out to a CRC error. This counter is updated, if needed, regardless of the RU state. If the RX_ER pin is asserted during a receive frame, this counter is incremented (only once per receive frame). This counter is counter is mutually exclusive to the alignment errors and short frames counters.
44	**Receive alignment errors.** This counter contains the number of frames that are both misaligned (in other words, CRS de-asserts on a non-octet boundary) and contain a CRC error. The counter is updated, if needed, regardless of the RU state. This counter is mutually exclusive to the CRC errors and short frames counters.
48	**Receive resource errors.** This counter contains the number of good frames discarded due to unavailable resources. Frames intended for a host whose RU is in the no resources state fall into this category. If the device is configured to save bad frames and the status of the received frame indicates that it is a bad frame, this counter is not updated unless the RU is in a no resources state.

Table 16. Statistical Counters

Byte Offset	Device Statistic
52	**Receive overrun errors.** This counter contains the number of frames known to be lost because the internal receive FIFO overflowed (also known as receive overrun). This can occur if the device is unable to get the necessary bandwidth on the PCI (system) bus. If the overflow condition persists for more than one frame, the frames that follow the first can also be lost. However, since a lost frame indicator does not exist, these lost frames may not be counted. A frame that was counted as an overrun will not be counted in other error counters (short frames, CRC errors, or alignment errors).
56	**Receive collision detect (CDT) errors.** This counter contains the number of frames that encountered collisions during frame reception. This counter is always 0 on the 82559.
60	**Receive short frame errors.** This counter contains the number of received frames that are shorter than the minimum frame length. It is mutually exclusive to the CRC errors and alignment errors counters and has a higher priority (in other words, a short frame will always increment only the short frames counter).
64	**Flow control transmit pause.** This counter contains the number of flow control frames transmitted by the device. The count includes both the XOFF frames transmitted and XON frames (in other words, PAUSE(0)) transmitted.
68	**Flow control receive pause.** This counter contains the number of flow control frames received by the device. It includes both the XOFF frames received and XON frames (PAUSE(0)) received.
72	**Flow control receive unsupported.** This counter contains the number of MAC frames received by the device that are not flow control pause frames. These frames are valid MAC frames with the predefined MAC type value and a valid address; however, they contain an unsupported opcode. In multimedia mode this counter tracks the pause low frames received. This count includes both the XOFF_Low frames received and XON_Low frames (PAUSE_Low(0)) received.
76	**Transmit TCO frames.** This counter is incremented when the 82559 transmits a packet initiated by the TCO controller (or ICH device). It should be noted that any transmission of TCO packets also affects the normal transmit counters.
78	**Receive TCO frames.** This counter is incremented when the 82559 receives a TCO packet. It should be noted that any reception of TCO packets also affects the normal receive counters.

□	Applicable to all controllers.
░	Applicable only to 82558 and later generation controllers.
▒	Applicable only to 82559 and later generation controllers.

As the above table indicates, the 8255x track of 16 different statistics. However, the 82558 also maintains three additional statistics (lightly shaded in the above table) for a total of 19 counters. In addition to the 19 statistics maintained by the 82558, the 82559 tracks two additional statistics and six reserved statistics (indicated by darker shading in the above table).

The counters are initially set to zero by the device after reset. They cannot be preset to anything other than zero. The device increments the counters by internally reading them, incrementing them, and writing them back. This process is invisible to the CPU and PCI bus. In addition, the counters adhere to the following rules:

- The counters are wrap around counters. After reaching 0FFFFFFFFh, the counters wrap around to 0. There is no indication when the counters wrap around to 0. Software must track this.
- The device updates the required counters for each frame. It is possible for more than one counter to be updated as multiple errors can occur in a single frame.
- The counters are 32 bits wide and their behavior is fully compatible with the IEEE 802.1 standard. The device supports all mandatory and recommended statistics functions through the status of the receive header and directly through these statistics counters.

Software can access the counters by issuing a Dump Statistical Counters SCB command. This provides a snapshot, in main memory, of the internal statistical counters. For the 82557, this dump always consists of 16 statistics. For the 82558 and 82559, this dump may contain more statistics depending on how the device is configured. It is recommended for software to use the following sequence for maintaining its own statistics:

1. Allocate an array in host memory large enough to hold all of the statistics dumped plus one additional Dword for status information (for example, 68 bytes for the 82557). This memory space must be Dword aligned.
2. Load the absolute address of this location into the device using the Load Dump Counters Address command.
3. Write zeros to the last Dword in this area. This can be done before or after step 2.
4. Write the Dump Statistical Counters or Dump and Reset Statistical Counters command into the CUC field in the SCB.
5. Wait for the device to dump the content of the statistical counters into the allocated memory area. The dump is followed by the device writing a completion status into the last Dword in this area. Software should check this Dword before processing the counters. A value of A005h indicates the Dump Statistical Counters command has completed. A value of A007h indicates the Dump and Reset Statistical Counters command has completed.

There should be no interrupts from the device after the completion of this operation. Also, no changes in the CU status or RU status fields should result after operation completion.

6.3.3 PORT Interface

Table 17. Port Register Location

Bits 31:16 (Upper Word)	Bits 15:0 (Lower Word)	Offset
SCB Command Word	SCB Status Word	Base + 0h
SCB General Pointer		Base + 4h
PORT		Base + 8h

The Port interface allows software to perform certain control functions on the device. Unlike action commands, port commands do not require access to the SCB. To initiate a port command, software should write the appropriate Dword (described below) to the Port register (offset 08h) in the CSR. Port commands automatically generate an internal selective or complete software reset, depending on the command. The Dword written as part of a Port command should include:

- 16-byte aligned address value on the AD31:AD4 data bus pins.
- Port function selection code on AD3:AD0

The port Dword may be written as a 32-bit entity, two 16-bit entities, or 4 8-bit entities. In the latter case, the device accepts only the port command after the high byte (offset Bh) is written; therefore, the high byte should be written last. Four different port commands are supported in the 82557 and 82558 devices. The 82559 and later generation controllers support an additional command, Dump Wake-up.

Table 18. Port Selection Function

Function	Pointer Field (Bits 31:4)	Opcode (Bits 3:0)
Software Reset	Don't care	0000
Self-test	Self-test results pointer (16 byte alignment)	0001
Selective Reset	Don't care	0010
Dump	Dump area pointer (16 byte alignment)	0011
Dump Wake-up	Dump area pointer (16 byte alignment)	0111

6.3.3.1 PORT Software Reset

The Port Software Reset is synonymous with the software reset and is used to issue a complete reset to the device. Software must wait for ten system clocks and five transmit clocks before accessing the SCB registers again. (This may be a conservative 10 µs delay loop in software.) A software reset clears the device CSR and the PCI master block internal registers. It also requires the device to be completely re-initialized.

6.3.3.2 PORT Self-test

The controller self-test begins by issuing an internal selective reset and running a general internal self-test of the device. The self-test function can be used to test the device micromachine functionality, internal registers and internal ROM. After the self-test is completed, the results are written to memory. The device provides the results of the self-test at the address specified by the self-test port command. The format of the self-test results is shown in Figure 11. The self-test command checks the following blocks:

- **ROM.** The contents of the entire ROM are sequentially read into a Linear Feedback Shift Register (LFSR). The LFSR compresses the data and produces a signature unique to one set of data. The results of the LFSR are then compared to a known good ROM signature. The pass or fail result and the LFSR contents are written into the address specified by the self-test port command.
- **Parallel Registers:** The micromachine performs write and read operations to all internal parallel registers and checks the contents for proper values. The pass or fail result is then written into the address specified by the self-test port command.
- **Diagnose:** The micromachine issues an internal diagnose command to the serial subsystem. The pass or fail result of the diagnose command is written into the address specified by the self-test port command.

Figure 11. Self-Test Results Format

Odd Word	Even Word
31 **16**	**15** **0**
CROM Content Signature	
00 00 00 00 00 00 00 00	00 0S 00 00 00 D0 MR 00

where

S (bit 12) General Self Test result: 0 = pass, 1 = fail

D (bit 5) Diagnose result: 0 = pass, 1 = fail

M (bit 3) Register result: 0 = pass, 1 = fail

R (bit 2) ROM Content result: 0 = pass, 1 = fail

After completing the self-test and writing the results to memory, the device executes a full internal reset and re-initializes to the default configuration.

Note: The self-test does not generate an interrupt or similar indicator to the host CPU upon completion.

6.3.3.3 Port Selective Reset

The Port Selective Reset is useful when only the device needs to be reset and all configuration parameters need to be maintained. The selective reset puts the CU and RU to the idle state but maintains the current configuration parameters. The selective reset maintains RU and CU base, HDS, error counters, configuration information, and individual and multicast addresses. As in a Port software reset, software must wait for ten system clocks and five transmit clocks before accessing the device (approximately 10 µs in software).

6.3.3.4 Port Dump

The Dump function writes dumped data to the specified location by the Dump Area Pointer. It is useful for troubleshooting “No Response” problems. If the device is in a no response state, the Port Dump operation can be executed to obtain internal device information without disturbing the rest of the system. When the Port Dump command is completed, it writes a DWORD with the value A006h at the end of the Dump space (Dword 149). The Dump command results format is discussed in Section 6.4.2.7, “Dump (110b)”.

6.3.3.5 PORT Dump Wake-up

The Port Dump Wake-Up command is only available on the 82559 and later generation controllers. It is not available on the 82558 or 82557.

After a Port Dump Wake-up command, the 82559 writes the stored data of the wake-up packet to the host memory starting at the address specified in the pointer field. The Dump Wake-up data structure is shown below:

Table 19. Dump Wake-up Data Structure

Dword Offset	D31		D0
0	Reserved	Status Word (A000h)	
1	Reserved		Byte Count
2:n	Wake-up Packet		

The 82559 executes the following sequence after it receives a Port Dump Wake-up command:

1. Writes the byte count field at Dword 1. This field contains the actual number of bytes posted in the host memory. A value of FFh indicates that the wake-up packet length exceeded the 120 bytes. In this case, only the first 120 bytes are posted.
2. Writes the Wake-up packet data starting at Dword 2.

3. Writes a status word composed of the Complete OK bits (equals A000h at Dword 0). Prior to the Dump Wake up packet command, the driver should initialize the status word to 0. After the Dump Wake-up packet command, it should poll the status word for a completion status.

6.3.4 EEPROM Control Register

The EEPROM control register is a 32-bit entity at offset 0Ch of the CSR space. They are used to read from and enable writes to an external EEPROM component.

Table 20. EEPROM Control Register Locations

Upper Word (D31:D16)	Lower Word (D15:D0)	Offset
SCB Command Word	SCB Status Word	Base + 0h
SCB General Pointer		Base + 4h
PORT		Base + 8h
EEPROM Control Register	Reserved	Base + Ch

The serial EEPROM or equivalent integrated circuit (IC) stores configuration data for the controller and the adapter. The EEPROM is a serial in and serial out device. Serial EEPROMs range in size from 16 to 256 registers of 16 bits per register. All accesses, read or write, are preceded by a command instruction to the device. The command instructions begin with a logical 1 as the start bit, two opcode bits (indicating read, write, erase, etc.), and n-bits of address. The address field varies with the size of the EEPROM and is 6 bits for a 64 register EEPROM and 8 bits for a 256 register device. The end of the address field is indicated by a dummy 0 bit from the EEPROM, which indicates the entire address field has been transferred to the device. A command is issued by asserting the chip select signal and clocking the data into the EEPROM on its data input pin relative to the serial clock input. The chip select signal is de-asserted after the completion of the EEPROM cycle (command, address and data).

6.3.4.1 CPU Accesses to the EEPROM

The EEPROM access port is shown below. This register is located at offset 0Eh in the device Control register block. The CPU directly manipulates these bits to read to or write from the EEPROM. There should be no other local bus activity at this time.

Figure 12. EEPROM Control Register

23	22	21	20	19	18	17	16
X	X	X	X	EEDO	EEDI	EECS	EESK

Table 21. EEPROM Control Register Bits Definitions

Bit	Symbol	Description
23:20		**Reserved.**
19	EEDO	**Serial Data Out.** This bit contains the value read from the EEPROM when performing a read operation on the EEPROM.

Table 21. EEPROM Control Register Bits Definitions

18	EEDI	**Serial Data In.** The value of this bit is written to the EEPROM when performing write operations.
17	EECS	**Chip Select.** Setting this bit to 1 enables the EEPROM. Setting the bit to 0 disables the EEPROM. This bit must be set to 0 for a minimum of 1 µs between consecutive instruction cycles.
16	EESK	**Serial Clock.** Setting this bit to 1 drives the serial clock line to the EEPROM high. Setting this bit to 0 drives the serial clock line low. Toggling this bit high and then low clocks data in or out of the EEPROM. The serial EEPROM specifies a minimum clock period of 4 µs. The minimum period that the clock can be high or low is 1 µs. If the clock is driven high for only 1 µs, then it must followed by a low period of 3 µs to meet the minimum clock frequency specification.

Table 22. EEPROM Opcode Summary (64-register EEPROM)

Instruction	Start Bit	Opcode	Address	Data	Comments
Read	1	10	$A_5A_4A_3A_2A_1A_0$		Read register $A_5A_4A_3A_2A_1A_0$
Write	1	01	$A_5A_4A_3A_2A_1A_0$	D1:D0	Write register $A_5A_4A_3A_2A_1A_0$
Erase	1	11	$A_5A_4A_3A_2A_1A_0$		Erase register $A_5A_4A_3A_2A_1A_0$
EWEN	1	00	11xxxx		Erase/write enable
EWDS	1	00	00xxxx		Erase/write disable
ERAL	1	00	10xxxx		Erase all registers
WRAL	1	00	01xxxx	D15:D0	Write all registers

6.3.4.2 Software Determination of EEPROM Size

To determine the size of the EEPROM, software may use the following steps.

Note: This algorithm will only work if the EEPROM drives a dummy zero to EEDO after receiving the complete address field.

1. Activate the EEPROM by writing a 1 to the EECS bit.
2. Write the read opcode, including the start bit (110b), one bit at a time starting with the most significant bit (1):
   a. Write the opcode bit to the EEDI bit.
   b. Write a 1 to EESK bit and wait the minimum SK high time.
   c. Write a 0 to EESK bit and wait the minimum SK low time.
   d. Repeat steps 2.a through 2.c for the next two opcode bits.
3. Write the address field, one bit at a time, keeping track of the number of bits shifted in, starting with the most significant bit.
   a. Write the address bit to the EEDI bit.
   b. Write a 1 to the EESK bit and wait the minimum SK high time.
   c. Write a 0 to the EESK bit and wait the minimum SK low time.
   d. Read the EEDO bit, looking for the dummy 0 bit.

   e. Repeat steps 3.a through 3.d until the EEDO bit equals 0. The number of loop iterations performed is the number of bits in the address field.

4. Read a 16-bit word from the EEPROM one bit at a time, starting with the most significant bit, to complete the transaction (but discard the output).
   a. Write a 1 to the EESK bit then wait the minimum SK high time.
   b. Read a data bit from the EEDO bit.
   c. Write a 0 to the EESK bit then wait the minimum SK low time.
   d. Repeat steps 4.a through 4.c an additional 15 times.
   e. De-activate the EEPROM by writing a 0 to the EECS bit.

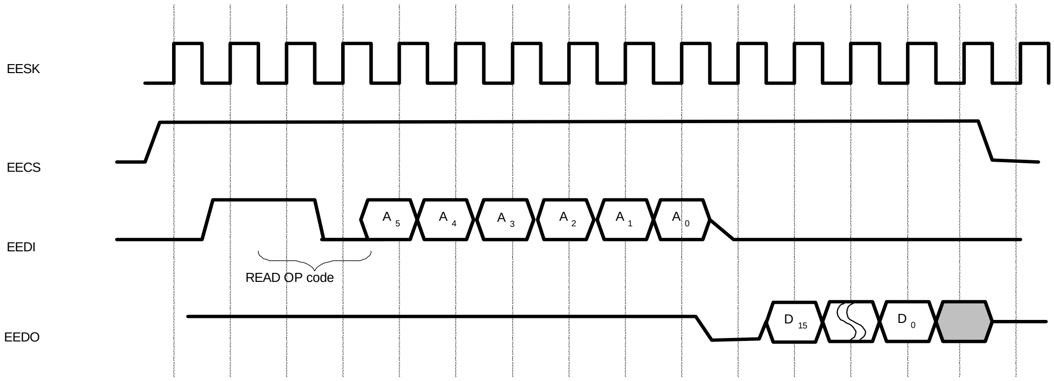
6.3.4.3 Software Read Access from the EEPROM

To read from the EEPROM, software is required to perform the following steps. The example is a read from address 02h (0000 0010b).

Note: Since the address field is written most significant bit first, software must know the address field size prior to starting a read or write access.

1. Activate the EEPROM by writing a 1 to the EECS bit.
2. Write the read opcode, including the start bit (110b), one bit at a time, starting with the most significant bit (1):
   a. Write the opcode bit to the EEDI bit.
   b. Write a 1 to EESK bit then wait the minimum SK high time.
   c. Write a 0 to EESK bit then wait the minimum SK low time.
   d. Repeat steps 2.a through 2.c for the next two opcode bits.
3. Write the address field, one bit at a time, starting with the most significant bit.
   a. Write the address bit to the EEDI bit.
   b. Write a 1 to EESK bit then wait the minimum SK high time.
   c. Write a 0 to EESK bit then wait the minimum SK low time.
   d. Read the EEDO bit (looking for the dummy 0 bit).
   e. Repeat steps 3.a through 3.d until the EEDO bit equals 0, indicating that the address field has been completely written.
4. Read a 16-bit word from the EEPROM, one bit at a time, starting with the most significant bit.
   a. Write a 1 to the EESK bit then wait the minimum SK high time.
   b. Read a data bit from the EEDO bit.
   c. Write a 0 to the EESK bit then wait the minimum SK low time.
   d. Repeat steps 4.a through 4.d an additional 15 times.
5. De-activate the EEPROM by writing a 0 to the EECS bit.

Figure 13. EEPROM Read Timing Diagram



6.3.4.4 Software Write Access to the EEPROM

Write access to the EEPROM is similar to the read access outlined above, with the differences of a write opcode and step 4:

1. Activate the EEPROM by writing a 1 to the EECS bit.
2. Write the read opcode, including the start bit (110b), one bit at a time, starting with the most significant bit (1):
   a. Write the opcode bit to the EEDI bit.
   b. Write a 1 to EESK bit then wait the minimum SK high time.
   c. Write a 0 to EESK bit then wait the minimum SK low time.
   d. Repeat steps 2.a through 2.c for the next two opcode bits.
3. Write the address field, one bit at a time, starting with the most significant bit.
   a. Write the address bit to the EEDI bit.
   b. Write a 1 to EESK bit then wait the minimum SK high time.
   c. Write a 0 to EESK bit then wait the minimum SK low time.
   d. Read the EEDO bit (looking for the dummy 0 bit).
   e. Repeat steps 3.a through 3.d until the EEDO bit equals 0, indicating that the address field has been completely written.
4. Write a 16-bit word to the EEPROM, one bit at a time, starting with the most significant bit (write a data bit to the EEDI bit):
   a. Write a 1 to the EESK bit then wait the minimum SK high time.
   b. Write a 0 to the EESK bit then wait the minimum SK low time.
   c. Repeat steps 4.a through 4.c an additional 15 times.
5. De-activate the EEPROM by writing a 0 to the EECS bit.

6.3.5 Management Data Interface Control Register

The Management Data Interface (MDI) Control register is a 32-bit entity at offset 10h of the CSR. The MDI Control register is used to read and write bits from the management data Interface. More details regarding the MDI can be found in *Section 7.1, “Management Data Interface (MDI)”* and *Section 8.1.2, “PHY Detection and Initialization”*.

Table 23. MDI Control Register Location

Upper Word (D31:D16)	Lower Word (D15:D0)	Offset
SCB Command Word	SCB Status Word	Base + 0h
SCB General Pointer		Base + 4h
PORT		Base + 8h
EEPROM Control Register	Reserved	Base + Ch
MDI Control Register		Base + 10h

The MII Management Interface allows software to have direct control over a MII compatible PHY through a control register in the device. This allows the driver software to place the PHY in specific modes such as full duplex, loopback, power down, etc., without the need for specific hardware pins to select the desired mode. This register, called the MDI Control register, resides at offset 10h in the Control register block. The CPU writes commands to this register and the Ethernet controller reads or writes control and status parameters to the PHY device through a serial, bi-directional data pin called Management Data Input/Output (MDIO). These serial data transfers are clocked by the management data clock output from the LAN controller.

Table 24. Management Data Pins

Symbol	Type	Name and Function
MDIO	In/Out	**Management Data Input/Output.** MDIO is a bi-directional signal between the device and an MII compatible PHY. It is used to transfer control information and status between the device and the PHY. Control information is driven by the Ethernet controller on the MDIO pin synchronously to MDC and sampled synchronously by the PHY. Status information is driven synchronously by the PHY and sampled synchronously by the LAN controller.
MDC	Out	**Management Data Clock.** MDC provides the timing reference for transfer of control information and status on the MDIO signal. The frequency of this clock is up to 2.5 MHz.

6.3.5.1 MDI Control Register

The MDI register may be written as a 32-bit entity, two 16-bit entities, or four 8-bit entities. When writing to the MDI register using word or byte access, the data is latched only on the write to the most significant byte of the register, which is located at offset 13h. Thus, the high byte should be written last.

Table 25. MDI Control Register Bits

Bits	Field	Description
31:30	Reserved	**Reserved.** This field is reserved and returns 0.
29	IE	**Interrupt Enable.** When this bit is set to 1 by software, it causes the device to assert an interrupt indicating the end of an MDI cycle.
28	R	**Ready.** set to 1 by the device at the end of MDI transaction (i.e., indicates a Read or Write has been completed. It should be reset to 0 by software at the same time the command is written.
27:26	Opcode	**Opcode.** For an MDI write, the opcode equals 01b, and for MDI read, 10b. 00b and 11b are reserved and should not be used.
25:21	PHYAdd	**PHY Address.**
20:16	RegAdd	**PHY Register Address.** **NOTE:** This value equals 1 for Intel PRO/100B TX and T4 adapters.
15:0	Data	**Data.** In a write command, software places the data bits here and the device shifts them out to the PHY. In a read command the device reads these bits serially from the PHY and software can read them from this location.

6.3.5.2 MDI Write cycle

The sequence of events for a MDI write cycle is:

1. The CPU performs a PCI write cycle to the MDI register with:
   a. Ready (bit 28) = 0
   b. Interrupt Enable (bit 29) = 1 or 0
   c. Opcode (bits 27:26) = 01b (write)
   d. PHYAdd = the PHY address from the MDI register
   e. RegAdd = the register address of the specific register to be accessed (0 through 31)
   f. Data = data to be written to the specified PHY register
2. The LAN controller shifts the following sequence out of the MDIO pin:
   <PREAMBLE><01><01><PHYADD><REGADD><10><DATA><IDLE>
3. The LAN controller asserts an interrupt indicating MDI is finished if the Interrupt Enable bit was set.
4. The LAN Controller sets the Ready bit in the MDI register to indicate step 2 has been completed.
5. The CPU may issue a new MDI command.

6.3.5.3 MDI Read cycle

The sequence of events for a MDI read cycle is:

1. The CPU performs a PCI write cycle to the MDI register with:
   a. Ready (bit 28) = 0
   b. Interrupt Enable (bit 29) = 1 or 0
   c. Opcode (bits 27:26) = 10b (read)

d. PHYAdd = the PHY address from the MDI register

e. RegAdd = the register address of the specific register to be accessed (0 through 31)

2. The LAN controller shifts the following sequence out of the MDIO pin:
   <PREAMBLE><01><10><PHYADD><REGADD><Z>
   where Z = the LAN controller tri-stating the MDIO pin
3. The PHY shifts the following sequence out of the MDIO pin:
   <0><DATA><IDLE>
4. The LAN controller discards the leading bit and places the following 16 data bits in the MDI register.
5. The LAN Controller asserts an interrupt indicating MDI has completed if the Interrupt Enable bit was set.
6. The LAN controller sets the Ready bit in the MDI register indicating the read is complete.
7. The CPU may read the data from the MDI register and issue a new MDI command.

6.3.6 Receive Byte Count Register

The early receive interrupt Receive Byte Count (RXBC) register is the 32-bit entity at offset 14h of the CSR. This read only register reflects the value of the internal receive DMA byte count register.

Note: Unless the software uses a very complicated early receive interrupt scheme, which requires the use of header RDFs, this register is of no value to software. Such a scheme could be used by software to increase performance by decreasing NOS receive latencies. However, most software early interrupt schemes would increase CPU utilization and software complexity. Thus, use of this register is not recommended.

Table 26. Receive Byte Count Register Location

Upper Word (D31:D16)	Lower Word (D15:D0)	Offset
SCB Command Word	SCB Status Word	Base + 0h
SCB General Pointer		Base + 4h
PORT		Base + 8h
EEPROM Control Register	Reserved	Base + Ch
MDI Control Register		Base + 10h
Early Receive Interrupt Receive Byte Count Register		Base + 14h

Bits 13:11 of this register are reserved and should equal 0. Bits 10:3 contain the receive DMA byte count. Bits 2:0 are hard wired to 0, giving an 8-byte granularity.

The RXBC register is first initialized to the size of the next receive data buffer. This data buffer size could equal the HDS size (if header RFDs are used) or the RFD buffer size. When a frame is received over the wire and passed to memory by the receive DMA, the register decrements until it reaches zero. At this point the register is set to the size of the next receive data buffer (HDS or RFD), and the receive DMA is restarted.

6.3.7 Early Receive Interrupt

Note: For operating systems with an increased interrupt latencies, the Early Receive Interrupt feature can be used to mask some of the latency. However, for Linux or Unix operating systems, the Early Receive Interrupt does not provide any benefit since these operating systems have little interrupt latencies. Thus, there is essentially no use for this feature in Linux or Unix operating systems.

The Early Receive Interrupt register is an 8-bit field at offset 18h of the CSR. This register is not present on the 82557. It is used for configuring the device to assert an additional receive interrupt before the entire packet has been received and deposited into host memory.

Table 27. Early Receive Interrupt Register Location

<table>
<tr><th colspan="2">Upper Word (D31:D16)</th><th colspan="2">Lower Word (D15:D0)</th><th>Offset</th></tr>
<tr><td colspan="2">SCB Command Word</td><td colspan="2">SCB Status Word</td><td>Base + 0h</td></tr>
<tr><td colspan="4">SCB General Pointer</td><td>Base + 4h</td></tr>
<tr><td colspan="4">PORT</td><td>Base + 8h</td></tr>
<tr><td colspan="2">EEPROM Control Register</td><td colspan="2">Reserved</td><td>Base + Ch</td></tr>
<tr><td colspan="4">MDI Control Register</td><td>Base + 10h</td></tr>
<tr><td colspan="4">Early Receive Interrupt Receive Byte Count Register</td><td>Base + 14h</td></tr>
<tr><td>PMDR</td><td>FC Xon/Xoff</td><td>FC Threshold</td><td>Early Rx Int</td><td>Base + 18h</td></tr>
</table>

When operating with the Early Receive Interrupt scheme, the device generates an early interrupt depending on the length of the frame. When a frame is received, the controller looks at the Type/ Length field (byte 13 and 14) of the received frame. If the Type/Length field contains a valid length value (0 < Type/Length ≤ 1500), the device generates an early interrupt approximately X quadwords before the end of the frame. If the Type/Length field contains a Type value, the device does not generate an early interrupt, except in the case where the Type value is 8137h (IPX) or 0800h (IP) and the device is configured to generate early interrupts on IPX or IP frames. In these two cases, it is known that the actual frame length is defined in bytes 17 and 18.

The early receive interrupt value X, in 8 bytes resolution, is programmed into the Early Rx Int register at address 18h in the device’s CSR. When this value is all zeros no early interrupt is generated. The Early interrupt is indicated by the ER bit in the SCB. and the assertion of INTA#. X should be determined by the driver as a function of the interrupt latency, PCI speed, etc.

The device also generates an interrupt at the end of the frame that will assure that no frame is missed (in case of a race condition), but is in most cases ignored by software (the interrupt is either already asserted or masked since the driver is in the Interrupt Handler).

The following list describes special cases for early receive interrupt assertion:

- If the programmed value is larger than the frame length, the device asserts the interrupt when it is ready to post the length field into memory.
- Short and overrun frames that contain less than the length minus the programmed value do not generate an early interrupt.
- The device does reclaim the RFD used by a frame that caused an early interrupt if this frame is an error frame and the device is configured to discard bad frames. This implies that the assertion of an ER interrupt does not guarantee that this frame will also generate an FR interrupt (in other words, the driver should not poll for the end of frame if it is not set). If the

device is configured to SBF no RFD is reclaimed and the driver may safely assume that an FR interrupt and RFD status will follow the ER interrupt.

- The ER interrupt mechanism operates only if the device does not discard the incoming frames. Therefore, the device does not generate ER interrupts before the RU is started. The device also may not assert the ER interrupt for frames that exceed the allocated buffer space and are being discarded.
- When the ER interrupt mechanism is first activated, it may not generate an ER interrupt for the first frame. An FR interrupt is generated if the RU is ready.

6.3.8 Flow Control Register

The flow control register is a 16-bit field at offset 18h (bits 23:8) of the CSR. This register does not exits on the 82557. It reflects flow control status information and contains some control bits that allow software to alter the flow control configuration parameters of the device.

Table 28. Flow Control Registers Location

Upper Word (D31:D16)		Lower Word (D15:D0)		Offset
SCB Command Word		SCB Status Word		Base + 0h
SCB General Pointer				Base + 4h
PORT				Base + 8h
EEPROM Control Register		Reserved		Base + Ch
MDI Control Register				Base + 10h
Early Receive Interrupt Receive Byte Count Register				Base + 14h
PMDR	FC Xon/Xoff	FC Threshold	Early Rx Int	Base + 18h

- **Bits 23:21 - Reserved.** These bits are reserved.
- **Bit 20 - FC Paused Low.** This read only bit is an indication of the device flow control state. It is set by the device when it receives a pause low command with a value greater than zero and cleared when the flow control timer reaches zero or a pause frame is received.
- **Bit 19 - FC Paused.** This read only bit is an indication of the device flow control state. It is set by the device when it receives a pause command with a value greater than zero and cleared when the flow control timer reaches zero.
- **Bit 18 - FC Full.** This read only bit indicates device flow control state. It is set by the device when it sends a pause command regardless of its cause (either due to the FIFO reaching the flow control threshold or due to the device fetching an RFD with its FCP bit set or due to writing into the Xoff bit). The bit is cleared by the device when it exits the above mentioned state.
- **Bit 17 - Xoff.** Writing 1 to this bit forces the Xoff request to 1. This causes the device to behave as if the FIFO extender is full. The Xoff request is cleared by writing 1 to the Xon bit (bit 16). Reading this bit returns 1 after it was set and 0 after the Xon bit was set. This bit returns 1 after an Xoff request was generated through the RFD Xoff bit until the Xon bit is set.
- **Bit 16 - Xon.** Writing 1 to this bit resets the Xoff request to the device. The Xoff request can become active through the RFD Xoff bit or if the driver writes 1 to the Xoff bit (bit 17). Reading this bit returns 0.
- **Bits 15:11 - Reserved.** These bits are reserved.

- **Bits 10:8 - FC Threshold.** The 82558 or later generation controller is capable of generating a flow control pause frame when its receive FIFO is almost full. This three-bit field determines the number of bytes left in the receive FIFO when the pause frame is generated. The trade-off occurs between a higher degree of data integrity (high flow control threshold value) or high performance (low flow control threshold value).

Table 29. Flow Control Threshold Values

FC TH Value	FC TH (free bytes in receive FIFO)	Comment
000	0.5 Kbyte	Fast system (recommended default).
001	1 Kbyte	
010	1.25 Kbyte	
011	1.5 Kbyte	
100	1.75 Kbyte	
101	2 Kbyte	
110	2.25 Kbyte	
111	2.5 Kbyte	Slow system.

6.3.9 Power Management Driver Register

The Power Management Driver Register (PMDR) provides an indication of power management events. It is an 8-bit field located at offset 18h of the CSR. This register is only present in the 82558 and later generation controllers and is not valid on the 82557.

Table 30. Power Management Driver Register Location

Upper Word (D31:D16)		Lower Word (D15:D0)		Offset
SCB Command Word		SCB Status Word		Base + 0h
SCB General Pointer				Base + 4h
PORT				Base + 8h
EEPROM Control Register		Reserved		Base + Ch
MDI Control Register				Base + 10h
Early Receive Interrupt Receive Byte Count Register				Base + 14h
PMDR	FC Xon/Xoff	FC Threshold	Early Rx Int	Base + 18h

The PMDR has evolved over time in the various Intel Fast Ethernet controllers. The PMDR bits for the 82558 and 82559 are described below.

Note: Not all bits are meaningful in the different generations of devices.

Table 31. Power Management Driver Register

Bits	Operation	Default	Description
31	Read/ Clear	0	Valid for 82559 only. **Link Status Change Indication.** The link status change bit indicates change in the link status. Writing a 1 to this bit will clear it.
30	Read/ Clear	0	Valid for 82559 (not 82559ER) only. **Magic Packet.** This bit is set when a Magic Packet is received regardless of the Magic Packet Wake-up disable bit in the configuration command and the PME enable bit in the PMCSR. Writing a 1 to this bit will clear it.
29	Read/ Clear	0	Valid for 82559 only. **Interesting Packet.** This bit is set when an interesting packet is received. The interesting packet is defined by the 82559 packet filters. Writing a 1 to this bit will clear it.
28	Read Only	0	**Reserved.**
27	Read Write	0	Valid for 82558 B-step only. **TCO Ready.** When this bit is set (by the driver), the TCO ready signal on the TCO interface is active signaling the TCO controller that the 82558 is idle and ready for a TCO cycle.
26	Read Only	0	Valid for 82558 B-step and 82559 only. **Force TCO Indication.**
25	Read Only	0	Valid for 82558 B-step and 82559 only. **TCO Request.** This bit is set to 1 when the 82559 is busy receiving packets for or transmitting packets from the TCO controller.
24	Read/ Clear (No clear on 82559)	0	Valid for the 82558 and 82559. **PME Status.** This bit is reflects the PME status bit in the PMCSR. It is set upon a wake-up event, independent of the PME enable bit. Writing a 1 to this bit clears it. It also clears the PME status bit in the PMCSR and the PME# signal. Writing a 0 has no effect on the 82558.

For the 82559, PMDR is initialized at alternate reset only and not at PCI reset (unless a PCI reset occurs with an alternate reset).

6.3.10 General Control Register

The General Control register provides control over some general purpose features in the 82559. It is an 8-bit field at offset 1Ch of the CSR. This register is only present in the 82559 and later generation controllers and is not valid for the 82558 or 82557.

Table 32. General Control Register Location

Upper Word (D31:D16)	Lower Word (D15:D0)		Offset
SCB Command Word	SCB Status Word		Base + 0h
SCB General Pointer			Base + 4h
PORT			Base + 8h
EEPROM Control Register	Reserved		Base + Ch
MDI Control Register			Base + 10h

Table 32. General Control Register Location

Upper Word (D31:D16)		Lower Word (D15:D0)		Offset
Early Receive Interrupt Receive Byte Count Register				Base + 14h
PMDR	FC Xon/Xoff	FC Threshold	Early Rx Int	Base + 18h
Reserved		General Status	General Control	Base + 1Ch

Table 33. General Control Register

Bits	Operation	Default PCI Reset	Description
7:2	R	0	**Reserved.**
1	R/W	0	**Deep Power Down on Link Down.** When this bit is 1, the 82559 may enter a deep power down state (sub 7 mA) in the D2 and D3 power states while the link is down. At this state, the 82559 does not maintain link integrity. It is not supported for point to point connection of two end stations.
0	R/W	0	**Clockrun Disable.** When this bit is 1, the 82559 always requests the PCI CLK. This mode can be used to overcome potential receive overruns caused by a very long system CLKRUN latency.

6.3.11 General Status Register

The General Status register provides some basic status information in the 82559. It is an 8-bit entity at offset 1Dh of the CSR. This register is only present in the 82559 and is not valid for the 82558 or 82557.

Table 34. General Status Register Location

Upper Word (D31:D16)		Lower Word (D15:D0)		Offset
SCB Command Word		SCB Status Word		Base + 0h
SCB General Pointer				Base + 4h
PORT				Base + 8h
EEPROM Control Register		Reserved		Base + Ch
MDI Control Register				Base + 10h
Early Receive Interrupt Receive Byte Count Register				Base + 14h
PMDR	FC Xon/Xoff	FC Threshold	Early Rx Int	Base + 18h
Reserved		General Status	General Control	Base + 1Ch

Table 35. General Status Register

Bits	Operation	Default	Description
7:3	R	0	**Reserved.**

Table 35. General Status Register

Bits	Operation	Default	Description
2	R		**HDX / FDX.** This bit indicates duplex mode: 0 = half duplex (HDX) and 1 = full duplex (FDX).
1	R		**10 / 100 Mbps.** This bit indicates the wire speed: 0 = 10 Mbps and 1 = 100 Mbps.
0	R		**Link Status Indication.** This bit indicates the status of the link: 0 = link down and 1 = link up.

6.4 Shared Memory Structures

The 8255x shared memory structures consist of the Command Block List (CBL) and the Receive Frame Area (RFA) and are controlled by the SCB portion of the CSR. The SCB is internal to the device while the CBL and RFA reside in main system memory.

6.4.1 Action Commands and Operating Modes

In addition to SCB control commands, the device can be controlled with action commands. This section lists all the action commands that can be a part of the CBL. Each command contains a command field, status and control fields, a link to the next action command, and command specific parameters. There are three basic types of action commands: device configuration and setup, transmission, and diagnostics. Alignment requirements are detailed in Table 10, “Alignment Requirements for 8255x Data Structures”.

Table 36. Operation Codes

Opcode	Name	Description
000	NOP	This command results in no action by the device other than the normal command processing such as fetching the command and decoding the command field.
001	Individual Address Setup	This command is used to load the device unique address. The unique address is contained in the parameter field of the command.
010	Configure	The configure command is used to load the device with its operating parameters. Upon reset, the device initializes to the IEEE 802.3 based parameters, with the exception of choosing either the PHY interface mode (for example, MII). If the user wishes to use any other values, the configure command is used.
011	Multicast Address Setup	This command allows the programmer to setup one or more multicast or multiple individual addresses in the device. These addresses are located in the parameter field of the command.
100	Transmit	One transmit command is used to send a single frame. If more than one frame needs to be sent, the host CPU can link multiple transmit commands together.
101	Load Microcode	This command downloads microcode to the device. **Note**: Documentation for microcode is beyond the scope of this manual.
110	Dump	This command causes the device to dump its internal registers into memory. The registers included are those loaded by the configure and address setup commands, plus status and other internal working registers.
111	Diagnose	The diagnose command puts the device CSMA/CD subsystem through a self-test procedure and reports the result of the internal test.

6.4.1.1 General Action Command Format

The format common to all action commands and the algorithms for beginning and completing the execution (also common to all action commands) is described below.

The general format of the Command Block (CB) includes the following fields:

Figure 14. General Action Command Format

Offset	Command Word Bits 31:16					Status Word Bits 15:0			
00h	EL	S	I	0000000000	CMD	C	X	OK	XXXXXXXXXXXXX
04h	Link Offset								
08h	Optional Address and Data Fields								

6.4.1.1.1 Beginning Execution

An action command can be started by either the CU start or CU resume control command. Otherwise, it may follow a previous action command. However, the actual command start may be delayed by RU activity.

The following sequence is performed by the CU at the beginning of execution of each action command:

1. The device reads 4 Dwords of the CB in one continuous PCI burst (if possible).
2. The device analyzes the contents of the command word to determine the necessary action.
3. The device reads and analyzes the link offset of the next CB and saves it.
4. The device performs specific actions according to the action command specified in the current command field.

If the commands are chained, the CU prefetches the next command block by accessing the address specified in the link offset field of the current CB. The device reads, analyzes, and saves the command word of the next CB (the following fields are saved for later use: EL, S, I and CMD).

6.4.1.1.2 Completing Execution

Command completion time is asynchronous to the beginning of the command. It is determined by the command type, RU activity, CU control commands, bus latency, etc. The CU is always in the active state at this time. The EL, S, and I bits determine the next actions.

The following sequence is performed by the CU at the completion of execution of an action command:

1. The devices writes command specific status to the status word of the current CB (usually the C and OK bits are written to). If the command is a transmit command, the C and OK bits are updated when the last transmit buffer DMA has completed, not after the actual transfer of the frame on the serial link. This allows the transmit buffer resources to be returned as soon as the data is copied into the device internal transmit FIFO instead of waiting for actual transmission on the wire. Transmit status is kept in the transmit statistical counters of the device.
2. If the I bit is set, the device sets a request for the CX interrupt.
3. If the EL bit is set, after completion of the command the CU becomes idle. If the S bit is set, the CU becomes suspended. Otherwise, the CU requests the beginning of the next action

command. A transition from an active to suspended state also generates a CNA interrupt if the device is configured to do so.

4. The device updates the status word in the SCB. (In step 1, the transmit command status is actually set at the end of transmit DMA, not at the completion of the actual transmit command.)

6.4.2 Specific Action Commands

6.4.2.1 NOP (000b)

This command results in no action by the device except for those performed in the normal command processing. It is used to manipulate the CBL. The NOP command format is shown below.

Figure 15. NOP Command Format

Offset	Command Word Bits 31:16					Status Word Bits 15:0			
00h	EL	S	I	0000000000	000	C	X	OK	XXXXXXXXXXXXX
04h	Link Address (A31:A0)								

Link Address This is the 32-bit address of the next command block. It is added to the CU base to obtain the actual address.

EL (Bit 31) If this bit is set to one, it indicates that this command block is the last one on the CBL. The CU will go from the active to the idle state after the execution of the CB is finished. This transition will always cause an interrupt with the CNA/CI bit set in the SCB.

S (Bit 30) If this bit is set to one, the CU will be suspended after the completion of this CB. A CNA interrupt will be generated if the device is configured for this. The CU transitions from the active to the suspended state after the execution of the CB.

I (Bit 29) If the I bit is set to one, the device generates an interrupt after the execution of the CB is finished. If I is not set to one, the CX interrupt will not be generated.

Bits 28:19 These bits are reserved and should all be set to 0.

CMD (Bits 18:16) This is the NOP command, which has a value of 000b.

C (Bit 15) This bit indicates the execution status of the command. Software should reset this bit before issuing the command to the device. Following a command completion, the device sets it to one.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

OK (Bit 13) The OK bit indicates that the command was executed without error. If it equals one, no error occurred (command executed OK). If the OK bit is zero and the C bit is set, then an error occurred.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

After reading the command and determining it is a NOP, the device CU performs the following sequence:

1. Begins execution of the NOP action command.
2. Prepares the status word with C equal to 1 and OK equal to 1.
3. Completes the NOP action command.

6.4.2.2 Individual Address Setup (001b)

This command is used to load the device with the individual address. This address is used by the device for inserting the source address during transmission and recognizing the destination address during reception. After a full reset and prior to individual address setup command execution, the device assumes the broadcast address (FF FF FF FF FF FFh) is the individual address in all respects.

This address loaded into the device is used as the individual address match reference. It will also be used as the source address of a transmitted frame (if the no source address insertion bit equals 0).

Figure 16. Individual Address Setup Command Format

Offset	Command Word Bits 31:16					Status Word Bits 15:0			
00h	EL	S	I	0000000000	001	C	X	OK	XXXXXXXXXXXXX
04h	Link Address (A31:A0)								
08h	IA 4th Byte, IA 3rd Byte					IA 2nd Byte, IA 1st Byte			
0Ch						IA 6th Byte, IA 5th Byte			

Link Address This is the 32-bit address of the next command block. It is added to the CU base to obtain the actual address.

EL (Bit 31) If this bit is set to one, it indicates that this command block is the last one on the CBL. The CU will go from the active to the idle state after the execution of the CB is finished. This transition will always cause an interrupt with the CNA/CI bit set in the SCB.

S (Bit 30) If this bit is set to one, the CU will be suspended after the completion of this CB. A CNA interrupt will be generated if the device is configured for this. The CU transitions from the active to the suspended state after the execution of the CB.

I (Bit 29) If the I bit is set to one, the device generates an interrupt after the execution of the CB is finished. If I is not set to one, the CX interrupt will not be generated.

Bits 28:19 These bits are reserved and should all be set to 0.

CMD (Bits 18:16) This is the Individual Address Setup command, which has a value of 001b.

C (Bit 15) This bit indicates the execution status of the command. Software should reset this bit before issuing the command to the device. Following a command completion, the device sets it to one.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

OK (Bit 13) The OK bit indicates that the command was executed without error. If it equals one, no error occurred (command executed OK). If the OK bit is zero and the C bit is set, then an error occurred.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

Individual Address The individual address of the node is 6 bytes long. IA byte 1 corresponds to the first byte of the address that is transmitted over the wire. For example, if the node address is 00 AA 00 01 02 03h, the bytes will be programmed as follows:

IA Byte 1 00h
IA Byte 2 AAh
IA Byte 3 00h
IA Byte 4 01h
IA Byte 5 02h
IA Byte 6 03h

The individual address is transferred by the transmit DMA through the transmit FIFO to the execution machine in the CSMA/CD module. Therefore, it may take some time to execute. The execution unit maintains a 48-bit individual address register used for source address insertion during transmission and for destination address recognition during reception. A reset causes the individual address register to be set to FF FF FF FF FF FFh.

After reading the command and determining whether it is an IA setup command, the device CU performs the following sequence:

1. Begins execution of the IA setup action command.
2. Initiates the transmit DMA with the address of the first byte of the individual address and a byte count of 6.
3. Waits for the transmit byte machine to complete the internal update of the individual address register.
4. Completes the IA setup action command.

6.4.2.3 Configure (010b)

The configure command loads the device with its operating parameters. A maximum of 22 configuration bytes are supported. The first eight bytes are used by the CU, and the remaining bytes are passed to the CSMA/CD unit through the transmit DMA. The minimum number of configuration bytes is 8.

Parameters not configured automatically use default values. The only exception is the PHY interface configuration bit. For 82557 based adapters, this bit must be set to either a zero or one before the 82557 will transmit or receive frames. For 82558 and later generation controllers, this bit must be set to 1 before the device will send and receive.

Figure 17. Configure Command Format

<table>
<tr><th>Offset</th><th colspan="5">Command Word Bits 31:16</th><th colspan="4">Status Word Bits 15:0</th></tr>
<tr><td>00h</td><td>EL</td><td>S</td><td>I</td><td>0000000000</td><td>010</td><td>C</td><td>X</td><td>OK</td><td>XXXXXXXXXXXXX</td></tr>
<tr><td>04h</td><td colspan="9">Link Address (A31:A0)</td></tr>
<tr><td>08h</td><td colspan="3">Byte 3</td><td colspan="2">Byte 2</td><td colspan="2">Byte 1</td><td colspan="2">Byte 0</td></tr>
<tr><td>0Ch</td><td colspan="3">Byte 7</td><td colspan="2">Byte 6</td><td colspan="2">Byte 5</td><td colspan="2">Byte 4</td></tr>
<tr><td>10h</td><td colspan="3">Byte 11</td><td colspan="2">Byte 10</td><td colspan="2">Byte 9</td><td colspan="2">Byte 8</td></tr>
<tr><td>14h</td><td colspan="3">Byte 15</td><td colspan="2">Byte 14</td><td colspan="2">Byte 13</td><td colspan="2">Byte 12</td></tr>
<tr><td>18h</td><td colspan="3">Byte 19</td><td colspan="2">Byte 18</td><td colspan="2">Byte 17</td><td colspan="2">Byte 16</td></tr>
<tr><td>1Ch</td><td colspan="3">00 00 00 00</td><td colspan="2">00 00 00 00</td><td colspan="2">Byte 21</td><td colspan="2">Byte 20</td></tr>
</table>

The individual bit fields of the configure command is another area where there are numerous differences between the controllers (82557, 82558, 82559, etc.). Therefore, a complete configuration map for each device will be presented below. Bit descriptions for the configuration bits follow the configuration map.

Table 37. 82557 Configuration Byte Map

Byte	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	Byte Count					
1	0	Transmit FIFO Limit			Receive FIFO Limit			
2	Adaptive Interframe Spacing							
3	Reserved (must be set to 0)							
4	0	Receive DMA Maximum Byte Count						
5	DMBC Enable	Transmit DMA Maximum Byte Count						
6	Save Bad Frames	Discard Overrun Receive	1	1	CI Interrupt	TNO Interrupt	1	Late SCB
7	0	0	0	0	0	Underrun Retry		Discard Short Receive
8	0	0	0	0	0	0	0	503/MII
9	0	0	0	0	0	0	0	0
10	Loopback		Preamble Length		NSAI	1	1	0
11	0	0	0	0	0	Linear Priority		
12	Interframe Spacing				0	0	0	L PRI MODE
13	0	0	0	0	0	0	0	0
14	1	1	1	1	0	0	1	0
15	CRS and CDT	1	0	0	1	0	Broadcast Disable	Promis-cuous
16	0	0	0	0	0	0	0	0
17	0	1	0	0	0	0	0	0
18	1	1	1	1	0	Receive CRC Transfer	Padding	Stripping
19	FDX Pin Enable	Force FDX	0	0	0	0	0	0
20	0	Multiple IA	1	1	1	1	1	1
21	0	0	0	0	Multicast All	1	0	1

Table 38. 82558 Configuration Byte Map

Byte	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	Byte Count					
1	0	Transmit FIFO Limit			Receive FIFO Limit			
2	Adaptive Interframe Spacing							
3	0	0	0	0	Term Write on CL	Read Al Enable	Type Enable	MWI Enable
4	0	Receive DMA Minimum Byte Count						
5	DMBC Enable	Transmit DMA Maximum Byte Count						
6	Save Bad Frames	Discard Overruns	Ext. Stat. Count	Extended Transmit CB	CI Interrupt	0	1	0
7	Dynamic TBD	2 Frames in FIFO	0	0	0	Underrun Retry		Discard Short Receive
8	CSMA Disable	0	0	0	0	0	0	1
9	MC Match Wake-up Enable	ARP Wake-up Enable	Link Wake-up Enable	VLAN ARP (0)	0	0	0	0
10	Loopback		Preamble Length		NSAI	1	1	0
11	0	0	0	0	0	0	0	0
12	Interframe Spacing				0	0	0	1
13	00000000 IP_address_Low							
14	11110010 IP_address_High							
15	CRS and CDT	1	0	Ignore U/L	1	Wait After Win	Broadcast Disable	Promis-cuous
16	FC Delay Least Significant Byte							
17	FC Delay Most Significant Byte							
18	1	Priority FC Threshold			Long Receive OK	Receive CRC Transfer	Padding	Stripping
19	Automatic FDX	Force FDX	Reject FC	Receive FC Restart	Receive FC Restop	Transmit FC	Magic Packet Wake-up	IA Address Match Wake-up Enable (0)
20	0	Multiple IA	Priority FC Location	1	1	1	1	1
21	0	0	0	0	Multicast All	1	0	1

Note: The shaded bits in the table above have different meaning for the 82558 B-step.

Table 39. 82559 Configuration Byte Map

Byte	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	Byte Count					
1	0	Transmit FIFO Limit			Receive FIFO Limit			
2	Adaptive Interframe Spacing							
3	0	0	0	0	Term Write on CL	Read Al Enable	Type Enable	MWI Enable
4	0	Receive DMA Minimum Byte Count						
5	DMBC Enable	Transmit DMA Maximum Byte Count						
6	Save Bad Frames	Discard Overruns	Ext. Stat. Count	Extended TxCB	CI Interrupt	TCO Statistics	1	0
7	Dynamic TBD	2 Frames in FIFO	0	0	0	Underrun Retry		Discard Short Receive
8	CSMA Disable	0	0	0	0	0	0	1
9	0	0	Link Wake-up Enable	VLAN TCO	0	0	0	TCP/UDP Check-sum
10	Loopback		Pre-amble Length		NSAI	1	1	0
11	0	0	0	0	0	0	0	0
12	Interframe Spacing				0	0	0	1
13	(00000000)							
14	(11110010)							
15	CRS and CDT	1	CRC16 (0)	Ignore U/L	1	Wait After Win	Broadcast Disable	Promis-cuous
16	FC Delay Least Significant Byte							
17	FC Delay Most Significant Byte							
18	1	Priority FC Threshold			Long Receive OK	Receive CRC Transfer	Padding	Stripping
19	Automatic FDX	Force FDX	Reject FC	Receive FC Restart	Receive FC Restop	Transmit FC	Magic Packet Wake-up	Reserved
20	0	Multiple IA	Priority FC Location	1	1	1	1	1
21	0	0	0	0	Multicast All	1	0	1

6.4.2.3.1 Configuration Parameters

The interpretation of the fields from the configuration byte maps are:

- **BYTE 0.**

Bits 5:0 - Byte Count. The byte count indicates the number of Command Block bytes to be configured (and is always included in the count). It allows changing some of the parameters by specifying a byte count less than the maximum number of configuration bytes (22 bytes). The first eight bytes are used by the CU, and the remaining bytes are passed to the CSMA/CD unit through the transmit DMA. The value permitted is 8 bytes.

Default - none.

Recommended -16h.

Note: If a runtime algorithm for Adaptive IFS is implemented, it is recommended that software issue an 8 byte configure command. If any of the first 8 bytes needs to be re-configured and the last 14 bytes do not need to be changed, then it is more appropriate to use an 8 byte configure command. This is a more efficient way of re-configuring the device.

- **BYTE 1.**

— Bits 6:4 - Transmit FIFO Limit. The transmit FIFO limit specifies the number of bytes located in the 64 byte dual-ported transmit FIFO at which the device requests the bus in order to transfer data from system memory to its internal transmit FIFO. The transmit FIFO is organized in 32-bit wide Dwords. The FIFO limit programming is showed in the table below.

Default - 0.

Recommended - 0.

— Bits 3:0 - Receive FIFO Limit. The receive FIFO limit specifies the number of bytes located in the dual-ported receive FIFO at which the device requests the bus in order to transfer data from its internal receive FIFO to system memory. The dual-ported receive FIFO is organized into 32-bit wide Dwords. For the 82557, the FIFO size is 64 bytes. For the 82558 and 82559 the FIFO is 128 bytes. The FIFO limit programming is showed in the table below.

Default - 8.

Recommended - The default value is fine. However, lower values will result in better PCI efficiency, whereas higher values will result in lower latencies.

Table 40. 82557 Dual-Port FIFO Settings

Configuration Value (Nibble Wide)				Transmit FIFO Limit		Receive FIFO Limit	
Binary (Transmit Bits 6:4 & Receive Bits 3:0)				Dwords	Bytes	Dwords	Bytes
0	0	0	0	0	0	16	64
0	0	0	1	1	4	15	60
0	0	1	0	2	8	14	56
0	0	1	1	3	12	13	52
0	1	0	0	4	16	12	48
0	1	0	1	5	20	11	44
0	1	1	0	6	24	10	40
0	1	1	1	7	28	9	36
1	0	0	0	0	0	8	32[a]
1	0	0	1	1	4	7	28
1	0	1	0	2	8	6	24
1	0	1	1	3	12	5	20

Table 40. 82557 Dual-Port FIFO Settings

1	1	0	0	4	16	4	16
1	1	0	1	5	20	3	12
1	1	1	0	6	24	2	8
1	1	1	1	7	28	1	4

[a] This line represents the default values.

NOTE: The configuration values are from 0 through Fh. The table shows the values in binary (4 bits wide).

Table 41. 82558 and 82559 Dual-Port FIFO Settings

Configuration Value				Transmit FIFO Limit		Receive FIFO Limit	
Binary (Transmit Bits 6:4 & Receive Bits 3:0)				Dwords	Bytes	Dwords	Bytes
0	0	0	0	0	0	32	128
0	0	0	1	1	4	30	120
0	0	1	0	2	8	28	112
0	0	1	1	3	12	26	104
0	1	0	0	4	16	24	96
0	1	0	1	5	20	22	88
0	1	1	0	6	24	20	80
0	1	1	1	7	28	18	72
1	0	0	0	0	0	16	64[a]
1	0	0	1	1	4	14	56
1	0	1	0	2	8	12	48
1	0	1	1	3	12	10	40
1	1	0	0	4	16	8	32
1	1	0	1	5	20	6	24
1	1	1	0	6	24	4	16
1	1	1	1	7	28	2	8

[a] This line represents the default values.

- **BYTE 2: Adaptive IFS.** This byte indicates the minimum number of PCI clocks counted between sending two transmit frames on the wire. The resolution of this counter is 8 PCI clocks making the range from 0 to 2040 PCI clocks.

  Default - 0.

  Recommended - 0.

- **BYTE 3.**

  — Bit 3 - Terminate Write on Cache Line. This bit is reserved on the 82557 and should be set to 0.

  However, when this bit is set on the 82558 or a later generation controller, the device attempts to terminate its write accesses on cache lines. This may yield lower PCI throughput in systems which are not extremely cache line oriented. This bit should therefore be set only in systems that are extremely cache line oriented.

  0 = Terminate Write on Cache Line disabled.

1 = Terminate Write on Cache Line enabled.

Default - 0 (Terminate Write on Cache Line disable).

Recommended - 0.

— Bit 2 - Read Alignment Enable. This bit is reserved on the 82557 and should be set to 0.

However, when this bit is set on the 82558 and later generation controllers, the device attempts to align its read accesses to cache lines. This may yield lower PCI throughput in systems that are not extremely cache line oriented. Thus, this bit should be set only in systems that are extremely cache line oriented. More information of the read alignment capability is detailed in Section 4.2.2, "Read Align".

0 = Read Alignment disabled.

1 = Read Alignment enabled.

Default - 0 (Read Alignment disabled).

Recommended - 0.

— Bit 0 - MWI Enable. This bit is reserved on the 82557 and should be set to 0.

However, for the 82558 and later generation controllers, it enables the device to perform Memory Write and Invalidate (MWI) cycles on the PCI bus. If both this bit and the MWI enable bit in the PCI command register are both set, then the device attempts to perform MWI cycles when writing data to system memory. If either this bit or the MWI enable bit in the PCI command register are clear, the device will not perform MWI cycles. A more detailed description of MWI can be found in Section 4.2.1, "Memory Write and Invalidate".

0 = MWI disabled. The device will not perform MWI cycles even if it is permitted by the PCI command register.

1 = MWI enabled. The device will perform MWI cycles if it is permitted by the PCI command register.

Default - 0 (MWI disabled).

Recommended - 1.

- **BYTE 4.**

Bits 6:0 - Receive DMA Maximum Byte Count. This byte indicates the maximum number of receive DMA PCI transfers that will be completed before internal arbitration. The counter has a 4 cycle resolution. This counter is useful in throttling back the receive DMA in order to let other device DMA channels, such as the transmit DMA, CU DMA, or RU DMA, complete PCI cycles. For instance, if the counter is set to 4, the receive DMA will only do a 16-cycle PCI transfer if one of the other internal DMA channels also wants to initiate a transfer. If no other internal DMA channels are requesting a transfer, the receive DMA may run an extended PCI burst. In order for this counter to be enabled, the DMA maximum byte count enable bit (byte 5, bit 7) must be set. If the enable bit is not set, the receive DMA will continue until it is finished (no other DMA unit can pre-empt it).

Note: If this counter is enabled and set to zero, then the receive DMA may be pre-empted almost immediately.

Default - 0.

Recommended - 0.

- **BYTE 5.**

— Bit 7 - DMA Maximum Byte Count Enable. Bit 7 enables the receive and transmit DMA maximum byte count enable counters. These counters are only valid when this bit is set to 1. This bit enables both the receive and transmit DMA maximum byte counters.

Default - 0.

Recommended - 0.

- Bit 6:0 - Transmit DMA Maximum Byte Count. This byte indicates the maximum number of transmit DMA PCI cycles that will be completed after internal arbitration. The counter has a 4 cycle resolution. It is useful in throttling back the transmit DMA in order to let other DMA channels, such as the receive DMA, CU DMA, or RU DMA complete PCI cycles. For instance, if the counter is set to 4, the transmit DMA will only do a 16-cycle PCI transfer if one of the other internal DMA channels also wants to initiate a transfer. If no other internal DMA channels are requesting a transfer, the transmit DMA may perform an extended PCI burst. In order for this counter to be enabled, the DMA maximum byte count enable bit (byte 5, bit 7) must be set. If the enable bit is not set, the transmit DMA will continue until it is done (no other DMA unit can pre-empt it).

Note: If this counter is enabled and set to zero, then the transmit DMA can be pre-empted almost immediately.

Default - 0.

Recommended - 0.

- **BYTE 6.**

- Bit 7 - Save Bad Frames.

This bit determines whether erroneous frames (CRC error, alignment error, etc.) are to be discarded or saved. Erroneous frames are those where the OK bit equals 0 in the frame descriptor status field. All frames are saved regardless of their status.

When the device is configured to save bad frames, the Receive Frame Descriptor (RFD) is not re-used for the next frame. When bad frames are not saved, these structures are re-used and no information is left in memory.

Note: The statistical counters are still updated upon receiving bad frames regardless of the state of this bit.

0 = Received bad frames are not saved in memory.

1 = Received bad frames are saved in memory.

Default - 0 (do not save bad frames).

Recommended - 0 (1 for promiscuous mode).

- Bit 6 - Discard Overrun Receive Frames. This bit determines whether Receive Overrun frames are to be discarded or saved. When activated (set to 0) the device may internally discard frames that were Overrun. When not activated (set to 1) the device will pass these frames to memory and only then reclaim the memory space or not according to the SBF configuration. If this bit is cleared (set to 0), Overrun frames will be discarded regardless of the setting of SBF. Note that Overrun frames will not always be discarded even if this bit is activated. If a frame has started to be transferred to memory before the overrun is detected the frame will be passed to memory regardless of the configuration.

0 = Discard overrun frames.

1 = Pass overrun frames to memory.

Default - 0 (do not pass overrun frames to memory).

Recommended - 0.

- Bit 5 - Extended Statistical Counter.

This bit is reserved on the 82557, and should be set to 1. For the 82558 or 82559, it determines the number of statistical counters that are dumped by the device when the Dump Statistical Counters or Dump and Reset Statistical Counters command is issued. If it is set to 1, the device dumps the 82557 compatible 16 counters into 68 bytes of memory. If the bit is 0, the device dumps the full 19 counters into 80 bytes of memory.

0 = Extended Statistical Counters.

1 = Standard Statistical Counters.

Default - 1 (standard statistical counters).

Recommended - 1.

— Bit 4 - Extended Transmit CB (TxCB). This bit is reserved on the 82557 and should be set to 1. However, for the 82558 or 82559, it determines the type of TxCB that is to be used by the device.

If this bit is 1, the device reads the standard 4 Dword TxCB. When this bit equals 0, the device reads 8 Dwords for all CBs and processes the TxCBs as Extended TxCBs as described in Section 6.4.2.5, “Transmit (100b)”.

0 = Extended TxCB.

1 = Standard TxCB.

Default - 1 (Standard TxCB).

Recommended - 1 for compatibility reasons. If performance is the main criteria, it is recommended that this bit equal 0.

— Bit 3 - CI Interrupt = CU Idle Interrupt. This bit determines whether the device generates an interrupt when the CU leaves the Active state (CNA interrupt) or when the CU enters the Idle state (CI interrupt). If CNA interrupt is enabled, the device will generate an interrupt when the CU goes from the Active to a non-active state (Idle or Suspended). Interrupts are generated whenever the device sees an EL or S bit in a CB that causes it to go into the Idle or Suspended state respectively on completion of the command. The CI interrupt will generate interrupts only on a transition from an Active to the Idle state. If the CI mode is enabled, interrupts can be generated in dynamic chaining (suspend/resume) by setting the I-bit on individual CBs.

0 = CNA Interrupt. An interrupt is generated when the CU goes from active to idle or suspended state.

1 = CI Interrupt. An interrupt is generated when the CU goes from the active to the idle state.

Default - 0 (CNA interrupt).

Recommended - 0, depending on the implementation of the transmit code.

— Bit 2. This bit is only used on the 82557 and 82559. However, it has a completely different meaning for both devices. For the 82557, it is the TNO Interrupt = Transmit Not OK Interrupt (82557 only), and for the 82559, the TCO Statistical Counter.

For the 82557, this bit determines whether or not the device generates an interrupt when a transmission ends with a bad status. If it is configured to TNO Interrupt, the device generates an interrupt by setting the CX interrupt bit in the SCB register and asserting the INTA# signal. This interrupt is related to the completion of actual transmission on the link and cannot be correlated to a specific transmit CB status. The status of the bad transmission is reflected only in a statistical manner through the statistical counters.

Note: When it is configured to TNO Interrupt, the 82557 still generates a CX interrupt if it encounters a transmit CB with its I bit set.

0 = CX Interrupt only.

1 = TNO Interrupt enabled.

For the 82559, setting this bit to 1 causes the device to provide TCO statistical counters. In this case, the statistical counters are 24-Dword long structures with the last 4 Dwords.

The effect of the TCO statistics bit together with the extended statistical counters bit is shown in the table below:

Table 42. Extended Statistics Functionality

TCO Statistical Counters	Extended Statistical Counters	Statistics Counters Functionality
0	1	82557 compatible (16 counters / 16 Dwords)
0	0	82558 compatible (19 counters / 19 Dwords)
1	1	82559 mode (21 counters / 24 Dwords)
1	0	Reserved

Default - 0.

Recommended - 0.

— Bit 0 - Late SCB = Late SCB Update. This bit is reserved on the 82558 and 82559 and should be set to 0 on those devices.

This bit only has meaning on the 82557 and determines when the device updates the SCB in relation to the completion of an action command. When it equals zero, the device updates the SCB status if there is an interrupt to report after completing the action command and before the next action command is started. If it is set to one, the device delays the updating of the SCB until after the next command on the CBL is started.

0 = Inactive (update SCB after command completion).

1 = Active (update SCB after next command is started).

Default - 0.

Recommended - 0.

- **BYTE 7.**

— Bit 6 - Two Frames in FIFO. This bit is reserved on the 82557 and should be set to 0. It is a valid bit for the 82558 and 82559 devices.

When this bit is set on the 82558 or 82559, the device limits the number of transmit frames in its FIFO to no more than two. This bit is expected to be used only when the device is used in multimedia mode.

0 = Two frames in FIFO disabled.

1 = Two frames in FIFO enabled.

Default - 0 (disabled).

Recommended - 0.

Bit 7 - Dynamic TBD. This bit is reserved on the 82557 and should be set to 0. However, when this bit is set on the 82558 or 82559, the device checks the validity of the transmit buffer pointer in the TBD and the EL bit in the TBD. When it is clear (0), the device assumes that the transmit buffer pointer in the TBD is always valid and the last TBD is indicated by the TBD count field in the transmit CB. When this configuration bit is set, the driver should set the TBD count field in the transmit CB to FFh.

0 = Dynamic TBD disabled.

1 = Dynamic TBD enabled.

Default - 0 (disabled).

Recommended - 0 (unless reducing transmit latency is large concern).

— Bits 2:1 - Underrun Retry. This field specifies the number of transmission retries after an underrun has occurred.

0 (00) = No re-transmission. If a transmitted frame encounters an underrun it will not be re-transmitted and the status indicating that the transmission failed will be returned and counted in the transmit underrun counter.

1 (01) = One re-transmission. If a transmitted frame encounters an underrun, it will be re-transmitted after the whole frame is stored in the FIFO.

2 (10) = Two re-transmissions. If a transmitted frame encounters an underrun it will be re-transmitted when there are 512 bytes in the FIFO. If the transmission encounters another underrun, the frame will be transmitted once again when the whole frame is stored in the FIFO.

3 (11) = Three re-transmissions. If a transmitted frame encounters an underrun, it will be re-transmitted when there are 512 bytes in the FIFO. If the transmission encounters another underrun, the frame will be transmitted once again when there are 1024 bytes in the FIFO. If the third attempt also encounters an underrun, the device will transmit it again when the whole frame is stored in the FIFO.

Default - 0 (no retransmission).

Recommended - 1.

— Bit 0 - Discard Short Receive Frames.

This bit determines whether short frames (shorter than 64 bytes) are to be discarded or saved. When it is set to 1, the device internally discards frames shorter than 64 bytes on the link regardless of the SBF setting. When it equals 0, the device passes these frames to memory and reclaims the memory space according to the SBF configuration. (Depending on how the device is configured, it may not reclaim memory space.)

0 = Pass short frames to memory.

1 = Discard short frames.

Default - 0 (pass short frames to memory).

Recommended - 1 (0 in promiscuous mode).

The discard short frames feature should we used with caution when it is combined with header receive interrupts. If the discard short frames feature is used, no data is passed to memory before 64 bytes are received. Therefore, even if the HDS is set to less than 64, the device will not pass a bad receive status if a short frame is encountered. However, a problem may occur if the discard short frames feature is not used and HDS is set to less than 64 bytes. In this case, the device may report a bad receive status if a short frame is encountered since it does not reclaim an RFD that has had its HDS field filled.

- **BYTE 8.**

— 82557: Bit 0 - 503/MII. This bit is reserved on the 82558 and 82559 and should be set to 1 on those devices. It is valid on the 82557. It is used to select the link interface mode of the 82557. If set to 503 mode (0), the 82557 transfers data to and from the link assuming 10 Mbps operation as done when operating with the 82503 or an equivalent serial interface. If set to MII mode (1), the 82557 transfers data to and from the link nibble-wide, assuming MII compatible operation.

0 = 503 mode.

1 = MII mode.

Default - none.

Recommended - For the 82557, the recommended value depends on the PHY detection. For the 82558 and 82559, the recommended value is 1.

— 82558/82559: Bit 7 - CSMA Disable. This bit is reserved on the 82557 and should be set to 0. It is valid on the 82558 or 82559 and used to disable the link operation of the device. If it is set to 1, the device will not receive data to or from the link. If it is set to 0, the device transfers data to and from the link. Software should always set this bit to 0 when it is issuing a configure command with more than 8 bytes.

0 = Enable.

1 = Disable.

Default - 0.

Recommended - 0.

— 82558/82559: Bit 0 - TCP/UDP Checksum. This bit is reserved on the 82557 and 82558, and should be set to 0 on those devices. This bit was added for the 82559. When this bit is set to ‘1, the 82559 provides a checksum word of incoming packets, excluding MAC header and CRC. A detailed description of the checksum calculation and memory structure can be found in Section 6.4.3.4, “No Buffer Performance Improvements (82558 and 82559)”.

0 = Disabled.

1 = Enabled.

Default - 0.

Recommended - 0 (unless the NOS supports TCP/UDP checksum offload).

- **BYTE 9.**

— Bit 7 - Multicast Match Wake Enable.

This bit is available only in the 82558 B-step. It should be set to 0 on 82557, 82558 A-step, and 82559 devices.

This bit enables assertion of the power management event signal (PME#) upon reception of packets that pass the multicast address filtering. The PME# signal is further gated by the PME enable bit in the PMCSR. Although this bit is not present in the 82559, this functionality is present through the extended wake-up packet command.

0 = Disabled.

1 = Enabled.

Default - 0 (disabled).

Recommended - 0.

— Bit 6 - ARP Wake-up Enable.

This bit is available only in the 82558 B-step. It should be set to 0 on the 82557, 82558 A-step, and 82559 devices.

This bit enables assertion of the power management event signal (PME#) upon reception of ARP frames (as defined above). The PME# signal is further gated by the PME enable bit in the PMCSR. Although this bit is not present in the 82559, this functionality is present through the extended wake-up packet command.

0 = Disabled.

1 = Enabled.

Default - 0 (disabled).

Recommended - 0.

— Bit 5 - Link Status Change Wake Enable. This bit is available only in the 82558 B-step and the 82559. It should be set to 0 on the 82557 and 82558 A-step devices.

This bit enables assertion of PME# upon a link status change event. The PME# signal is further gated by the PME enable bit in the PMCSR.

0 = Disabled.

1 = Enabled.

Default - 0 (disabled).

Recommended - 0.

- Bit 4 - VLAN ARP (82558 B-step) or VLAN TCO (82559). This bit is available only in the 82558 B-step and 82559. It should be set to 0 on 82557 and 82558 A-step devices.

  VLAN ARP: For the 82558 B-step, this bit enables wake-up upon reception of ARP frames with a dynamic presence of a VLAN header. This bit takes affect only if the ARP wake-up enable bit is also set. This same functionality has been moved on the 82559 to the extended wake-up packet command.

  0 = VLAN header not supported.

  1 = Dynamic VLAN header supported.

  Default - 0 (off).

  Recommended - 0.

  VLAN TCO: On the 82559, this bit activates VLAN capability filtering of received TCO packets at nominal D0 state. When this bit is clear, the 82559 implements receive TCO in D0 for non-tagged TCO packets only. If this bit is set, the 82559 looks for both tagged and non-tagged TCO packets. When the 82559 is in the power down state or the Force TCO state, the 82559 looks for the VLAN type for recognition of tagged versus non-tagged packets. In all other states, the 82559 does not look for the VLAN type for qualification.

  0 = Only TCO packets without VLAN headers are supported.

  1 = TCO packets with or without VLAN headers are supported.

  Default - 0 (off).

  Recommended - 0.

- **BYTE 10.**

  - Bits 7:6 - Loopback. This bit defines the type of loopback.

    00 = Normal operation (no loopback).

    01 - Internal loopback.

    10 - Reserved.

    11 - External loopback (loopback pin active).

    Default - 00.

    Recommended - 00.

  - Bits 5:4 - Pre-amble Length. This bit selects the length of the pre-amble, not including the SFD, according to table below.

Table 43. Pre-amble Length

D5	D4	Preamble Length
0	0	1 byte
0	1	3 bytes
1	0	7 bytes
1	1	15 bytes

Default setting - 10b (7 bytes).

Recommended - 10b.

- Bit 3 - No Source Address Insertion. This bit determines the source of the source address.

  0 = SA insertion (SA comes from internal device IA).

1 = No SA insertion (SA comes from memory).

Default - 1.

Recommended - Depends on the NOS and driver environment.

- Bits 2:0 - Linear Priority. These bits are reserved on the 82558 and 82559 and should be set to 000b on those devices.

  For the 82557, these bits correspond to he number of slot times that the device will wait after the IFS or after backoff before enabling transmission. A higher number reduces the priority. Stations with this value set to 0, the highest priority, conform to the IEEE 802.3 backoff algorithm.

  Default - 000 (normal CSMA/CD operation).

  Recommended - 000.

- **BYTE 12.**
  - Bits 7:4 - Interframe Spacing. This field specifies the period (in multiples of 16 bit times) that the device must defer after the later of the following two events:
    - The last bit has been transmitted.
    - Carrier sense becomes inactive.

    Default - 96 (6 in the register).

    Recommended - 6.
  - Bit 0 - Linear Priority Mode. This bit is reserved on the 82558 and 82559 and should be set to 1 for those devices. For the 82557, it determines the way the linear priority mechanism works.

    0 = Wait after transmit only. The device defers for IFS + N * slot time after the transmission of the frame only.

    1 = Wait transmit or receive. The device defers for IFS + N * slot time after the transmission or reception of a frame.

    N = Linear priority number.

    Default - 0.

    Recommended - 0.
- **BYTE 13 and BYTE 14.**

  Bits 7:0 (byte 13 and byte14) - IP Address. This field is available only in the 82558 B-step. Byte 13 should be set to 0h, and byte 14 should be set to F2h for the 82557, 82558 A-step, and 82559 devices.

  For the 82558 B-step, this field holds the 16 least significant bits of the IP address used for ARP frame filtering. For example, to configure the filter for ARP frames with an IP address of 012h 034h 056h 078h, the following values are written to the configuration block:

Table 44. 82558 B-step Configuration Block ARP Frame IP Address

Configuration Block Offset	Configuration Parameter Name	Example Value
13	IP Address Low	078h
14	IP Address High	056h

The ARP filter compares the value stored in offset 13 of the configuration block to the byte at offset 41 in an ARP frame without a VLAN header and to byte 45 in ARP frames with a VLAN header.

Similarly, the value at offset 14 of the configuration block is compared to the byte at offset 40 in ARP frames without a VLAN header and to byte 44 in ARP frames with a VLAN header.

The 16-bit value of the IP address in the configuration block is in non-canonical format, while the IP address of an ARP frame is stored in canonical format. Using the same IP address in the example above (012h 034h 056h 078h), the ARP filter performs the following comparison:

Table 45. 82558 B-step ARP Frame IP Address Mapping

Configuration Block		
Block Offset	Offset 13 IP Address Low	Offset 14 IP Address High
Example Value	078h	056h
Frame Offset	40	41
Incoming Frame		

Although this field is not present in the 82559, its functionality is present in the extended wake-up packet command.

Default - 00h, F2h (for backward compatibility).

- **BYTE 15.**

  — Bit 7 - CRS or CDT. When this bit is set, the device will interpret an active CDT during transmission as an active carrier.

  0 = CRS only.

  1 = CRS or CDT.

  Default - 1.

  Recommended - 1 for 82557/82503 based designs, 0 for 82557/MII based designs, 0 for 82558 or 82559 based designs.

  — Bit 5 - CRC16.

  This bit selects the 16-bit or 32-bit CRC engine. When it is set to 1, the 82559 operates with a 16-bit CRC generator. Clearing this bit selects the 32-bit CRC engine. (Ethernet operates with a 32-bit CRC.)

  0 = 32-bit CRC.

  1 = 16-bit CRC.

  Default - 0.

  Recommended - 0 (must be 0).

  — Bit 4 - Ignore U/L. This bit is reserved on the 82557 and should be set to 0. When this bit is set on the 82558 or 82559, the device ignores the U/L bit when checking for IA match on received frames.

  0 = Consider U/L bit.

  1 = Ignore U/L bit.

  Default - 0 (Consider U/L bit).

  Recommended - 0.

  — Bit 2 - Wait After Win. This bit is reserved on the 82557 and should be set to 0. For the 82558 or 82559, it activates the modified backoff algorithm, Wait After Win (Section 6.7, “Collision Backoff Modification in Switched Environments”).

  0 = Wait After Win disabled.

  1 = Wait After Win enabled.

Default - 0 (disabled).

Recommended - 0.

- — Bit 1 - Broadcast Disable. When this bit is set, it disables the device from receiving any frames with a broadcast address (address of all 1s). Promiscuous mode setting overrides broadcast disable.

  Default - 0 (off).

  Recommended - 0.

- — Bit 0 - Promiscuous Mode. When this bit is set, it causes the device to receive all frames regardless of their destination address.

  Default - 0 (off).

  Recommended - 0 (1 if promiscuous mode will be enabled).

- **BYTE 16.**

  Bits 7:0 - FC Delay Least Significant Byte. This byte is reserved on the 82557 and should be set to 00h.

  For the 82558 or 82559, this byte corresponds to the least significant byte of the flow control delay field. This delay is used as the time parameter for the assembly of transmitted flow control frames. The value is defined in slot time (512 bit time) resolution.

  Default - 0 (82557 compatible).

  Recommended - 0.

- **BYTE 17.**

  Bit 7:0 - FC Delay Most Significant Byte. This byte is reserved on the 82557 and should be set to 40h.

  For the 82558 or 82559, this byte corresponds to the most significant byte of the flow control delay field. This delay is used as the time parameter for the assembly of transmitted flow control frames.

  Default: 01000000 (82557 compatible).

  Recommended: 0.

- **BYTE 18.**

  - — Bit 3 - Long Receive OK. This bit is reserved on the 82557 and should be set to 0.

    When this bit is set on the 82558 or 82559, the device considers received frames that have a data field longer than 1500 bytes as good frames. The frames are still flagged as long in the RFD status word but the OK bit is set. Software can pass the frame to the NOS if long frames are supported.

    Default - 0 (disabled).

    Recommended - 0 (unless the device is used in a VLAN environment).

  - — Bit 2 - Receive CRC Transfer. When this feature is enabled, the CSMA/CD block transfers the CRC to host memory. If the CRC is not transferred to memory it is stripped. The report of CRC and alignment error is reported immediately. Setting this bit disables the stripping enable bit. Thus, if the frame is padded (the length is less than the byte count), the frame will be transferred to memory as a whole, without stripping, even if stripping is enabled.

    Default - 0 (disabled).

    Recommended - 0.

  - — Bit 1 - Padding Enable. If this bit is set to 1, the device enables the padding mechanism. If the byte count of a transmitted frame is less than the minimum frame length, a padding byte (7Eh) will be transmitted to pad (in other words, fill) the minimum frame length. The CRC will include the padded bytes. If padding is disabled, no padding bytes will be added even if the frame is a short frame.

Default - 1 (enabled).

Recommended - 1.

— Bit 0 - Stripping Enable. If this bit is set to 1, the device enables the stripping mechanism. If the byte count of a received frame is lower than the actual length received, every byte beyond the specified length will be stripped from the frame except the CRC. If it is set to 0, no stripping will be performed.

Stripping is performed only on frames that have the Length/Type field set to Length ($0 < \text{value} \le 1500$).

Default - 0 (disabled).

Recommended - 1. However, it should be avoided if the minimum packet length cannot be safely assumed.

- **BYTE 18.**

Bits 6:4 - Priority Flow Control Threshold. These bits are reserved on the 82557 and should be set to 111.

For the 82558 or 82559, this three-bit field defines the threshold at which the device differentiates between Pause and Pause Low FC frames (Section 6.6.3.1, “Priority Flow Control Operation”). Every FC frame with “priority field” greater than “Priority FC Threshold” is considered Pause_Low. Setting this configuration field to any value other than the default 111 activates the Priority FC mode.

Default - 111 (disabled).

Recommended - 111 (unless the priority flow control threshold mechanism is implemented).

- **BYTE 19.**

— Bit 7 - Full Duplex Pin Enable. When this bit is set, the device examines the FDX# pin to determine if it should operate in full duplex or half duplex mode. If the force full duplex bit (bit 6) is set to one, then this bit has no meaning and the device will not examine the level of the FDX# pin. This is described in the table below.

Default - 0 (off) for the 82557 and 82558 A -step; 1 (on) for the 82558 B-step and 82559.

Recommended - 1.

Table 46. Full Duplex Functionality

FDX PIN ENABLE (bit 7)	FORCE FDX (bit 6)	State of FDX#	Device Operating Mode
0	0	0	Half Duplex
1	0	0	Full Duplex
0	1	0	Full Duplex
1	1	0	Full Duplex
0	0	1	Half Duplex
1	0	1	Half Duplex
0	1	1	Full Duplex
1	1	1	Full Duplex

— Bit 6 - Force Full Duplex. This bit forces the device to operate in full duplex mode. Transmit and receive execution can be active simultaneously. CRS is only a receive activity indicator. Minimum reception spacing between back to back frames is two bytes.

Default - 0 - off.

Recommended - 0 (1 if the user specifies a valid override).

— Bit 5 - Reject FC (address filtering of full duplex transmit flow control frames). This bit is reserved on the 82557, and should be set to 0.

When this bit is set on the 82558 or 82559, received flow control frames will not be passed to memory, regardless of any address mechanism they might pass. This bit has no effect on the action taken upon reception of such a frame.

Default - 0 (82557 compatible).

Recommended - 0.

— Bit 4 - Full Duplex Restart Flow Control. This bit is reserved on the 82557 and should be set to 0.

When this bit is set on the 82558 or 82559, it enables transmissions of flow control frames to the peer station in order to stop its transmissions. The sending of such a frame is triggered by the high threshold parameter, as set in the flow control threshold register (Section 6.3.8, “Flow Control Register”). The flow control frame transmitted will carry the configured flow control delay value in the time field. When the receive FIFO is empty, another flow control frame is sent with the value 0 in the time field.

Default - 0 (82557 compatible).

Recommended - 0.

— Bit 3 - Full Duplex Restop Flow Control. This bit is reserved on the 82557 and should be set to 0.

When this bit is set on the 82558 or 82559, it enables transmissions of flow control frames to the peer station in order to stop its transmissions. The sending of such a frame is triggered by the high threshold parameter, as set in the flow control threshold register (Section 6.3.8, “Flow Control Register”). The flow control frame transmitted will carry the configured flow control delay value in the time field. When this delay expires, the device checks the receive FIFO state. If the FIFO is not empty, another flow control frame is sent.

Default - 0 (82557 compatible).

Recommended - 0.

— Bit 2 - Full Duplex Transmit Flow Control Disable. This bit is reserved on the 82557 and should be set to 0.

When this bit is 0 on the 82558 or 82559, it enables the transmit flow to be paused by incoming flow control commands. Flow control commands come from the link as special flow control frames with a time parameter. In this mode, upon reception of such a frame, the device pauses transmissions according to the time parameter.

Default - 0 (82557 compatible).

Recommended - 0.

— Bit 1 - Magic Packet Wake-up disable. This bit is reserved on the 82557 and 8259ER and should be set to 0 for those devices.

When this bit is set on the 82558 or 82559, it disables the assertion of a special wake-up signal upon reception of a Magic Packet* frame (which is a frame with certain predefined fields). This bit takes effect only if the wake enable bit is set in the PMCSR.

Default: 0 - on (82557 compatible).

Recommended - 0.

— Bit 0 - Address Wake-up (82558 A-step); IA Match Wake Enable (82558 B-step). This bit is reserved on the 82557 and 82559 and should be set to 0 on those devices.

When this bit is set on the 82558 A-step, it enables assertion of the INTA# signal as a special wake-up signal upon reception of a frame that passes any of device address filtering mechanisms (according to the configuration of broadcast, promiscuous, IA, multicast all or multiple IA). This bit takes effect only if the wake enable bit is set in the PMCSR.

For the 82558 B-Step, this bit has a similar but slightly different function. On the 82558 B-step, it enables the assertion of the PME# signal upon reception of packets that pass the individual address filtering. The PME# signal is further gated by the PME enable bit in the PMCSR.

Default - 0 (off; 82557 compatible).

Recommended - 0.

- **BYTE 20.**

— Bit 6 - Multiple IA. When this bit is set, it enables the device to receive multiple IA frames using the HASH mechanism. If it is disabled, HASH will only be used for multicast frames (odd address number).

Default - 0 (disabled).

Recommended - 0.

— Bit 5: Priority FC Location. This bit is reserved on the 82557 and should be set to 01.

For the 82558 and 82559, this bit determines the location of the priority field in the flow control frame. When it equals 0, the priority field is in byte #19 (after the time filed). When it is 1, the priority field is in byte #31 (12 bytes later).

0 = Priority field in byte #19.

1 = Priority field in byte #31.

Default - 1.

- **BYTE 21.**

— Bit 3 - Multicast All. This bit enables the device to receive all frames with a multicast address (1 in the least significant byte - odd address).

Default - 0 (disabled).

Recommended - 0.

The first 8 bytes of the configuration are kept by the CU, and the remainder are transferred by the transmit DMA to the execution machine. When a configuration command is received, the CU performs the following sequence:

1. Begins execution of the configuration action command.
2. Reads the first eight configure bytes and saves their content.
3. Writes the configure command to the transmit FIFO.
4. Initiates the transmit DMA to transfer the remainder of the configure bytes, up to the specified byte count, to the execution machine.
5. Waits for the execution machine to complete its internal update of configuration registers.
6. Prepares the status word with C = 1 and OK = 1.
7. Completes the configuration action command.

In the case of a port selective reset, the execution machine maintains configuration registers for the device. In the case of a port software reset or a hardware reset, the device reverts to the default values.

6.4.2.4 Multicast Setup (011b)

The multicast setup command is used for loading multicast IDs into the device for filtering purposes. As previously noted, the filtering done on the multicast IDs is not perfect and some unwanted frames may be accepted. This command resets the current filter and reloads it with the specified multicast IDs. The format of the multicast addresses setup command is shown below.

Figure 18. Multicast Setup Command Format

<table>
<tr><th>Offset</th><th colspan="6">Command Word Bits 31:16</th><th colspan="5">Status Word Bits 15:0</th></tr>
<tr><td>00h</td><td>EL</td><td>S</td><td>I</td><td colspan="2">0000000000</td><td>011</td><td>C</td><td>X</td><td>OK</td><td colspan="2">XXXXXXXXXXXXX</td></tr>
<tr><td>04h</td><td colspan="11">Link Address (A31:A0)</td></tr>
<tr><td>08h</td><td colspan="4">2nd Byte</td><td colspan="2">1st Byte</td><td>X</td><td>X</td><td colspan="3">Multicast Count</td></tr>
<tr><td>0Ch</td><td colspan="11">Multicast Address List</td></tr>
<tr><td></td><td colspan="4">Nth Byte</td><td colspan="2"></td><td colspan="4"></td><td></td></tr>
</table>

Link Address This is the 32-bit address of the next command block. It is added to the CU base to obtain the actual address.

EL (Bit 31) If this bit is set to one, it indicates that this command block is the last one on the CBL. The CU will go from the active to the idle state after the execution of the CB is finished. This transition will always cause an interrupt with the CNA/CI bit set in the SCB.

S (Bit 30) If this bit is set to one, the CU will be suspended after the completion of this CB. A CNA interrupt will be generated if the device is configured for this. The CU transitions from the active to the suspended state after the execution of the CB.

I (Bit 29) If the I bit is set to one, the device generates an interrupt after the execution of the CB is finished. If I is not set to one, the CX interrupt will not be generated.

Bits 28:19 These bits are reserved and should all be set to 0.

CMD (Bits 18:16) This is the multicast setup command, which has a value of 011b.

C (Bit 15) This bit indicates the execution status of the command. Software should reset this bit before issuing the command to the device. Following a command completion, the device sets it to one.
NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

OK (Bit 13) The OK bit indicates that the command was executed without error. If it equals one, no error occurred (command executed OK). If the OK bit is zero and the C bit is set, then an error occurred.
NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

Multicast Count This 14-bit field indicates the number of bytes in the multicast list field. The multicast count must be a multiple of 6 bytes; otherwise, the device reduces the multicast count to the nearest multiple of 6. If the multicast count equals 0, it resets the hash table, which is equivalent to disabling the multicast filtering mechanism.

Multicast List This field contains a list of multicast addresses or multiple IAs to be accepted by the device. The least significant bit of the most significant byte of each multicast address must equal 1.

The transmit DMA transfers the list of multicast addresses from memory to the execution machine through the transmit FIFO. The CU performs the following sequence:

1. Begins execution of the multicast setup action command.
2. Reads the multicast count field and saves it internally.
3. Initiates the transmit DMA with the multicast list address and byte count according to the multicast count field.
4. Waits for the transmit byte machine to complete the internal hash table update.
5. Completes the multicast setup action command.

The receive byte machine maintains a 64-bit hash table used for checking multicast addresses during reception. After the execution machine reads a multicast setup command, it clears the hash table and reads the bytes in groups of 6. Each group is hashed using CRC logic, and the bit in the hash table that bits 2 through 7 of the CRC register point to is set to one. A group that is not complete has no effect on the hash table. The execution machine notifies the CU after completion.

An incoming frame is accepted if it has a destination address with the significant bit in the most significant byte equal to 1 and after hashing points to a bit in the hash table whose value is one. The hash function is selecting bits 2 through 7 of the transmit CRC register. A software reset causes the hash table to become all zeros.

6.4.2.5 Transmit (100b)

Transmit commands can use either the simplified or flexible memory structure. The simplified structure expects the transmit data to reside entirely in the memory space immediately after the transmit command block (TCB). The flexible transmit structure allows multiple data buffers to be accessed through a transmit buffer descriptor (TBD) array. Both models require the use of one transmit command block per frame transmitted.

The 82558 introduced several new enhancements to the design of the software and hardware interface for transmits. Both the 82558 and the 82559 allow software to use either the original 82557 compatible TCB format or the new extended TCB format. For the 82558 and 82559 devices, the TCB type used is determined by a configuration bit (Section 6.4.2.3, "Configure (010b)").

The format of the 82557 TCB (original TCB format) is illustrated in the figure below. There were a few additional capabilities added in the 82558 and 82559 that can be utilized through this command block interface. These new capabilities are highlighted.

Figure 19. Transmit Command Format

Offset	Command Word Bits 31:16								Status Word Bits 15:0				
00h	EL	S	I	CID	000	NC	SF	100	C	X	OK	U	XXXXXXXXXXXX
04h	Link Address (A31:A0)												
08h	Transmit Buffer Descriptor Array Address												
	TBD Number				Transmit Threshold				EOF	0	Transmit Command Block Byte Count		

Link Address This is the 32-bit address of the next command block. It is added to the CU base to obtain the actual address.

EL (Bit 31) If this bit is set to one, it indicates that this command block is the last one on the CBL. The CU will go from the active to the idle state after the execution of the CB is finished. This transition will always cause an interrupt with the CNA/CI bit set in the SCB.

S (Bit 30) If this bit is set to one, the CU will be suspended after the completion of this CB. A CNA interrupt will be generated if the device is configured for this. The CU transitions from the active to the suspended state after the execution of the CB.

I (Bit 29) If the I bit is set to one, the device generates an interrupt after the execution of the CB is finished. If I is not set to one, the CX interrupt will not be generated.

CID (Bits 28:24) The CNA Interrupt Delay field is only present on 82558 and later generation controllers. (It is not a valid field for the 82557, unless special microcode is downloaded to this device.) The CID indicates the length of time CNA interrupts are delayed by the device.

Bits 23:21 These bits are reserved and should all be set to 0.

NC 0: CRC and Source Address are inserted by the controller. If the “No Source Address Insertion” (NSAI) bit is set by the configure command, then only the CRC is inserted by the controller. Normally, this bit should be set because it is desirable to have the device compute and insert the CRC automatically.

1: CRC and Source Address are not inserted by the controller and are assumed to come from memory.

SF This bit indicates whether the device is operating in simplified or flexible mode.

0 = Simplified Mode. All transmit data is in the TCB, and the TBD array address field must equal all 1s.

1 = Flexible Mode. Data is in the TCB (optional) and in a linked list of the TBDs.

CMD (Bits 18:16) This is the transmit command, which has a value of 100b.

C (Bit 15) The C bit indicates that the transmit DMA has completed processing the last byte of data associated with the TCB. This is not the actual completion of the transmit command as the C bit indicates in other action commands. The actual completion of a transmit command occurs when the frame is actually sent out on the wire. At the end of actual transmission, no further status is posted in the TCB, but the transmit statistical counters are updated.

OK (Bit 13) The OK bit indicates that the command was executed without error. If it equals 1, no error occurred (command executed OK). If the OK bit is zero and the C bit is set, then an error occurred.

NOTE: For the transmit command, the OK bit is always set when the C bit is set.

U (Bit 12) The U bit indicates that one or more underruns were encountered by this or previously transmitted frames since the last TCB status update. Since there is no mechanism for indicating underruns during or at the end of frame transmission, this bit is set in addition to the transmit underruns statistical counter for software management purposes.

Bits 11:0 These bits must be set to all zeros.

TBD Array Address In flexible mode, this is a 32-bit address pointing to the first TBD in a contiguous list of TBDs called the TBD array. A TBD is two Dwords, a transmit buffer pointer and buffer size data. In simplified mode this field should be set by software to a null pointer (0FFFFFFFFh).

TBD Number In flexible mode, this represents the number of transmit buffers in the contiguous TBD array. It should have a one to one correspondence of TBDs and buffers in the array. If the device finds the TBD number equal to 0, it assumes the TBD array address is a null pointer and the EOF bit is set. The 82558 and 82559 have a special dynamic TBD mode that the 82557 does not have. If the dynamic TBD mode is enabled (in the configure command), software should write a value of FFh into this field. Software should also mark each TBD as valid or invalid. In the 82557, the TBD number is the only indication that the TBD is the last associated with a particular transmit frame.

Transmit Threshold The transmit threshold defines the number of bytes that should be present in the controller's transmit FIFO before it starts transmitting the frame. The value is internally multiplied by 8 to give a granularity of 8 bytes. For example, a value of 1 means the 82557 will start transmitting only when it has 8 bytes in its transmit FIFO. The transmit threshold should be within a range of 1 to 0E0h. (The value 0FFh should not be used.)

EOF The EOF bit indicates if the whole frame is in the transmit command block. For consistency, it should be set by software, although it is not checked in simplified or flexible mode.

TCB Byte Count For either simplified or flexible mode, the controller is able to transmit data from memory immediately contiguous to the TCB itself. The amount of data to be read from this space is determined by the 14-bit TCB byte count. This counter indicates the number of bytes that will be transmitted from the transmit command block, starting with the third byte after the TCB count field (address N + 10h). The TCB count field can be any number of bytes up to a maximum of 2600, which allows the user to transmit a frame with a header having an odd number of bytes. In simplified mode, the TCB byte count indicates the total number of bytes to be transmitted and should not equal zero. In flexible mode, if the TCB byte count equals 0, then all data is taken from the buffers pointed to by the TBD array.

The 82558 and 82559 also offer a more advance transmit command block. When they are configured to use extended TCBs, the device reads an 8-Dword TCB from host memory into its internal registers instead of the standard 4-Dword TCB. The new TCB structure is composed of the 4 standard TCB Dwords followed by 2 TBDs or 2 Dwords each.

The fields in the first 4 Dwords are identical to the first 4 Dwords of the standard TCB except for the TBD array address, which points to the third TBD rather than the first one. In other words, if the frame consists of more than two transmit buffers, the rest of the TBDs, from the third one onwards, are placed in a standard TBD array, which is pointed to by the TBD array address field. The TBD number field indicates the total number of TBDs including the two TBDs located in the latter 4 Dwords of the extended TCB. If a transmitted frame consists of less than two TBDs, the driver can set the size field of the second (or both) TBD to zero or set the EL bit on the first TBD.

The advantage of the extended TCB is that it enables the device to read the TCB and the first two TBDs in one 8-Dword PCI burst. This eliminates one PCI read and its associated latency and enables both the TCB and its immediate data field to be cache line aligned.

An extended TCB is assumed to be flexible. The two TBDs that are part of the extended TCB may use the EL bit, but it is required that the transmit buffer pointers in the two TBDs are always valid (in other words, not equal to 0).

The transmit buffer descriptor (TBD) array is a contiguous structure of TBDs. A TBD is defined as a transmit buffer address and a transmit buffer size. The format of the TBD array is shown below.

Figure 20. Transmit Buffer Descriptor

Odd Word (Bits 31:16)		Even Word (Bits 15:0)		
Transmit Buffer #0 Address				0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	EL	0	Size (Actual Count)	4
Transmit Buffer #1 Address				8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	EL	0	Size (Actual Count)	C
Transmit Buffer #N Address				N*8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	EL	0	Size (Actual Count)	N*8+4

Transmit Buffer #N	This is the starting address of the memory area that contains the data to be sent. It is an absolute 32-bit address. It does not add the CU base value to determine the physical address.
EL (End of List)	The EL bit is not used by the 82557 and is only valid for 82558 and later generation devices. When it is set, the TBD is the last TBD associated with this transmit frame.
Size (Actual Count)	This 14-bit quantity specifies the number of bytes that hold information for the current buffer. It is set by the CPU before transmission.

6.4.2.5.1 Dynamic TBD Mode

Note: Dynamic TBD mode only exists in the 82558 and 82559 devices. It is not a valid mode for the 82557.

The 82557 requires all TBDs to be setup by the driver before the device is issued the CU start or CU resume command. However, in environments where virtual addresses must be translated to physical addresses, TBD setup is a very time consuming process. The 82558 and 82559 support a new configuration mode called “dynamic TBD” mode, which activates two new features in the TBD structure. (Details regarding configuration of this mode are in Section 6.4.2.3, “Configure (010b)”.) Each TBD, which still has two 32-bit Dwords as defined in the 82557, has the following two features defined.

- **NV - Not valid pointer.** When the device is configured to dynamic TBD, it checks the transmit buffer pointer in the TBD. If it equals all zeros, it is considered to be an invalid pointer. The device discards the TBD and attempts to read it again as soon as possible. When this pointer is valid, the TBD is valid and the device can use the transmit buffer.
- **EL - End of list bit.** When this bit is set, the current TBD is the last TBD associated with this transmit frame. The EL bit does not have to be set in the last TBD as indicated by the TBD number field in the TCB. If the device reaches the last TBD in the array as indicated in the TBD number field, it terminates the transmission regardless of the EL bit status. However, if the device detects a TBD with a valid pointer and its EL bit set, it terminates the frame even if it did not reach the number of TBDs indicated in the TBD number field. If dynamic TBD configuration is currently in use, the driver should set the value of the TBD number field in the TCB to FFh.

These two features enable the driver to spontaneously add TBDs after issuing the CU resume command. The goal of automatically adding transmit buffer descriptors is to reduce overall latency by achieving more parallelism between the driver and the device. This scheme allows the driver to issue the CU resume command after filling in the first TBD (or even before that) and, while the device is processing the transmit command block, the first TBD, and first transmit buffer, to continue setting up the TBD array.

The driver programs FFh in the TBD number field of the transmit command block. The driver prepares the first TBD in the TBD array with a valid pointer, and it is considered valid. Afterwards, the driver issues the command to hardware.

The device flow is:

1. Fetch any immediate data from TCB.
2. Fetch the first and second TBDs.
3. Fetch the first transmit buffer since the pointer is valid in the first TBD.
4. Begin transmission (depending on the transmit threshold value).

5. Fetch data if the transmit buffer pointer is zero (invalid) in the second TBD or poll the TBD.
6. Finish the transmission if the EL bit is set.

6.4.2.5.2 Transmit Command Operation

The execution of a transmit command causes frame transmission. If the frame experiences collisions, the device automatically attempts to re-transmit the frame up to 15 times. If it still experiences collisions after 16 tries, the device increments the maximum collisions counter. The following sequence outlines a general transmit command operation for the flexible memory structure (TCBs and TBDs).

1. Place the transmit command opcode (100b) in the command word.
2. Place the destination address and length field in the appropriate transmit structure.

   The TBD array address should point to the first TBD in the array. When the simplified memory structure is used, the TBD array address is not used.
3. Configure the transmit buffer address and size (actual count) for each buffer. The last buffer in the TBD array is determined by the TBD number field in the TCB.

The flow of events for transmitting a single frame using a flexible TCB is:

1. The CPU creates a TCB and TBD array in system memory. The transmit buffer address pointers in the TBDs point to valid data buffers in host memory.
2. The CPU writes a CU start command (or CU resume if the CU is suspended) into the SCB. The write event causes the device to read the CUC field, and the device notices that it should start the CU.
3. The device processes the SCB, reads the SCB general pointer, and clears the SCB command word.
4. The device reads the first TCB in the CBL and the first TBD from the TBD array.
5. If the TCB size field does not equal zero, the TCB holds data to be transmitted and the device reads this data first.
6. The controller reads the first transmit data buffer from host memory at the address provided in the transmit buffer #0 address field of the transmit buffer array.
7. After the transmit threshold bytes are read (either from one or multiple transmit buffers), the controller begins frame transmission to the PHY interface.
8. If there are multiple TBDs, the controller reads the next TBD from the TBD array.
9. After the first buffer has been completely read, the device starts reading the transmit data from the next buffer.
10. After the last buffer is completely read, the device sets the C bit in the TCB, enabling the driver to re-use reuse the TCB, TBDs, and transmit buffers. The controller posts the underrun bit in the TCB if an underrun occurred since the last TCB status was reported.
11. The device completes the frame transmission to the serial interface (for the 82557, either MII or 82503).
12. The controller updates its internal transmit status counters.

The transmit command differs from other action commands. Generally, the action commands have parameters in one memory block. However, the transmit command may have parts of the parameters scattered in a linked list of buffers. The CU spontaneously pre-fetches the buffers in the list.

While the CU pre-fetches the address and byte count of one buffer, the transmit DMA is transferring the previous buffer to the transmit byte machine. Completion of a buffer transfer by the transmit DMA triggers the CU to initiate the transmit DMA for the next buffer (if it is already pre-fetched) and to start the pre-fetch of the next buffer. The buffer pre-fetch cycle is terminated when the transmit DMA reads the last buffer (indicated by the TBD number) and transfers it to the transmit FIFO.

Internally, the controller CU performs the following sequence during transmission:

1. Begins execution of the transmit action command.
2. Reads and saves the TBD array address.
3. If the TCB size field is greater than zero, the device performs as follows:
    a. If the TBD array address is not equal to all ones, the CU performs a pre-fetch and transfer cycle, initiates the transmit DMA to the address of the first byte of the destination address field in the CB and to the byte count of the last specified data byte in the command block.
    b. If the TBD array address equals all ones, after completing DMA of the command block the CU writes the end of command byte to the transmit FIFO.
4. If the TCB size field in the command block is zero, it runs a buffer pre-fetch and transfer cycle and forces one dummy DMA completion.
5. The CU waits for completion of the transmit command. This includes only the transfer of the whole frame to the transmit FIFO subsystem, not the frame transmission by the CSMA/CD unit. At this point, the device posts the C bit (to 1) in the TCB. The CPU can reclaim the TCB and associated data structures.
6. If transmission completed with a collision (but did not exceed the maximum collisions), regardless of errors, the subsystem generates a re-transmit command and sends the data bytes again from the FIFO. This causes re-transmission of the frame without any additional PCI bus access.
7. If the transmit DMA encountered an underrun due to a lack of PCI bus bandwidth, it appends a jam pattern to the end of the partially transmitted frame. Frames that are aborted during transmission are jammed. Such an interruption of transmission can be caused by several different events. Jamming will not start before completion of pre-amble transmission (before the first byte of the destination address is sent). Collisions detected during transmission of the last 11 bits of the frame will not result in jamming.
8. The device CU completes the transmit action command.

The device may report completion of a transmit command before the actual transmission on the link has completed. Software can reuse the resources to prepare a new transmit command. When the frame is eventually transmitted on the link, the CSMA/CD sub-system will return the status of the transmission to the 82557 micro-machine, but the TxCB Status **WILL NOT** be updated in host memory. The CU will update the internal Tx counters according to the Tx status

6.4.2.5.3 Framing Operation

The transmit byte machine maintains the following registers for construction of frames: pre-amble pattern, SFD field, source address, CRC generator, and jam patterns.

After the transmit byte machine reads the transmit command from the transmit FIFO, a frame is constructed and transferred to the transmit bit machine for bit and nibble transmission. The transmit byte machine performs the following sequence:

1. Pre-amble bytes are transferred according to pre-amble length configuration parameter.

2. The SFD field is transferred.
3. Start CRC calculation.
4. Read and transfer the 6 destination address bytes from the transmit FIFO.
5. If the no source address insertion configuration parameter is zero, the individual address should be transferred as the source address. Otherwise, the source address should be read and transferred from the transmit FIFO. If the no source address insertion is 1 and there are less than address length bytes in the transmit FIFO, a DMA underrun is forced.
6. All remaining bytes from the transmit FIFO are read and transferred. These are the length and data fields.
7. The CRC is transferred.
8. If the device is configured to enable padding, the flag bytes (07Eh) are transferred automatically so that a valid frame (64 bytes including CRC) is transferred onto the link.

If a collision or underrun occurred during transmission, the transmit byte machine completes the transfer of the pre-amble and transfers 4 bytes of the jam pattern. If a collision occurred, the retry counter is incremented. Jamming will not start before completing pre-amble transmission.

If a collision is detected during transmission of the last 11 bits in the frame, it does not result in jamming. If the collision is detected during transmission of the last bit or later, the collision is not reported and re-transmission does not occur. This can happen for an invalid frame shorter in length than the slot time.

Note: A DMA underrun cannot logically occur during the pre-amble because the serial subsystem generates its own pre-amble.

6.4.2.5.4 Delayed CNA Interrupts

The 82558 and later generation controllers have the ability to delay the CNA interrupt for a predefined length of time, called the CNA interrupt delay (CID). If the CID is set to a non-zero value, the device does not assert the interrupt immediately when entering a non-active state. Instead, it initializes an internal counter with the CID parameter. The interrupt is asserted only when the counter expires. If a CU resume or CU start command is issued while the counter is counting, the interrupt will not be asserted. This opens a window for the device driver to set a new command without the overhead of an additional interrupt service routine (ISR).

The device delays the interrupt, regardless of whether it is configured for CI interrupts or CNA interrupts. However, the controller does not delay the updating of the CU status field. Therefore, if the CID is greater than zero, it posts the CU status field (without the CNA bit) before it posts the CNA bit and asserts the INTA# signal. (This feature is primarily targeted to NDIS systems but can be beneficial for other systems as well.)

The CID parameter is set on a frame by frame basis, and its value is read by the device from the TCB. Since the internal counter is automatically initialized to the CID value from the current TCB, the existing value of the counter (set by the previous TCB) is overwritten, causing the counter to reset even if it has not yet reached zero. This allows a rolling delay, where a number of back to back TCBs can be given to the controller while only generating one interrupt at the end of the chain.

The purpose of the delay is to avoid issuing this interrupt if it is not required. It is assumed that the interrupt is not required in the following cases:

- The device was issued another action command and the CU returns to the active state.

- The device received a frame and generated a receive interrupt.

If neither of these events occurred, the controller generates a CNA interrupt when the CID time interval has elapsed. The actual delay experienced may be longer than the CID value that was loaded. The CID is given in a granularity of approximately 256 PCI clocks and the maximum value is 8192 clocks (which corresponds to 8 to 256 µs in a 33 MHz system).

The delayed CNA interrupt flow is outlined below.

1. The delayed CNA interrupt is issued in the suspend or idle state. In other words, if the device is in the suspend or idle state, raising the interrupt would be delayed by specified time in the CID field of each command header.
2. The end of receive processing cancels the pending delayed CNA interrupt. It also causes the CNA interrupt to be set simultaneously with the frame interrupt, regardless of the internal counter value. This is based on the theory that any pending transmit cleanup would be done in the context of a receive interrupt.
3. Resume and start commands cancel pending delayed CNA interrupts. This allows only the last TCB of a chain to be interrupted (the rolling delay).
4. **The CX interrupt (caused by the I bit) is not affected in any way by this mode or delay parameter.** It may be that regardless of anything else, we may want to interrupt on, say, every third TX in a chain to return resources to the protocol. This would be accomplished by setting the I bit in the TxCB. There would be no delay associated with an I-bit interrupt. Note that if I and S bits are set in a TxCB and the CID field is set to a non-zero value, the CX & CNA interrupts *will not occur together*.
5. The delay specification is a 5-bit field and ranges between 8 and 256 µs, in 8 µs resolution. The actual delay will only be within a certain percentage of the value specified (but never less than the specified delay). The inaccuracy percentage is typically in the range of 10 to 20%. However, in a few extreme conditions (for example, a lot of bad frames received), the delay may be more than 20% above the specified delay.

The CNA interrupt delay (CID) field in the TCB is located in bits 28:24 of the first Dword of the TCB.

6.4.2.6 Load Microcode (101b)

Note: Documentation for microcode is beyond the scope of this manual.

The load microcode command downloads a 64 Dword microcode patch to the device's internal microcode.

The microcode that operates on one device (for example, the 82557), will not operate on another device (the 82558 or 82559). The load microcode command format is shown below:

Figure 21. Load Microcode Command Format

Offset	Command Word Bits 31:16					Status Word Bits 15:0			
00h	EL	S	I	0000000000	101	C	X	OK	XXXXXXXXXXXXX
04h	Link Address (A31:A0)								
08h	First Microcode Dword								
260h	64th Microcode Dword								

Link Address This is the 32-bit address of the next command block. It is added to the CU base to obtain the actual address.

EL (Bit 31) If this bit is set to one, it indicates that this command block is the last one on the CBL. The CU will go from the active to the idle state after the execution of the CB is finished. This transition will always cause an interrupt with the CNA/CI bit set in the SCB.

S (Bit 30) If this bit is set to one, the CU will be suspended after the completion of this CB. A CNA interrupt will be generated if the device is configured for this. The CU transitions from the active to the suspended state after the execution of the CB.

I (Bit 29) If the I bit is set to one, the device generates an interrupt after the execution of the CB is finished. If I is not set to one, the CX interrupt will not be generated.

Bits 28:19 These bits are reserved and should all be set to 0.

CMD (Bits 18:16) This is the load microcode command, which has a value of 101b.

C (Bit 15) This bit indicates the execution status of the command. Software should reset this bit before issuing the command to the device. Following a command completion, the device sets it to one.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

OK (Bit 13) The OK bit indicates that the command was executed without error. If it equals one, no error occurred (command executed OK). If the OK bit is zero and the C bit is set, then an error occurred.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

Microcode Data This field contains the 64 Dwords of microcode data downloaded to the device. This data patches the device’s hard-coded microcode, which allows the behavior of the device to be altered or adapted.

The load microcode command instructs the device to download microcode data from host memory into its internal microcode RAM. The microcode data is organized as a 64-Dword memory block that is appended to a standard command block header. The device starts execution of downloaded microcode immediately following the successful completion of the load microcode command. The device continues executing the downloaded microcode until the device is reset through its hardware or software reset mechanisms.

Note: Documentation for developing new microcode patches for the Intel® Fast Ethernet controllers is beyond the scope of this manual.

6.4.2.7 Dump (110b)

This command causes the contents of various device registers to be placed in a memory area specified by the user. It is supplied as a self diagnostic tool and provides registers of interest to the user. The format of the dump command is shown below.

Figure 22. Dump Command Format

Offset	Command Word Bits 31:16					Status Word Bits 15:0			
00h	EL	S	I	0000000000	110	C	X	OK	XXXXXXXXXXXXX
04h	Link Address (A31:A0)								
08h	Buffer Address								

Link Address This is the 32-bit address of the next command block. It is added to the CU base to obtain the actual address.

EL (Bit 31) If this bit is set to one, it indicates that this command block is the last one on the CBL. The CU will go from the active to the idle state after the execution of the CB is finished. This transition will always cause an interrupt with the CNA/CI bit set in the SCB.

S (Bit 30) If this bit is set to one, the CU will be suspended after the completion of this CB. A CNA interrupt will be generated if the device is configured for this. The CU transitions from the active to the suspended state after the execution of the CB.

I (Bit 29) If the I bit is set to one, the device generates an interrupt after the execution of the CB is finished. If I is not set to one, the CX interrupt will not be generated.

Bits 28:19 These bits are reserved and should all be set to 0.

CMD (Bits 18:16) This is the dump command, which has a value of 110b.

C (Bit 15) This bit indicates the execution status of the command. Software should reset this bit before issuing the command to the device. Following a command completion, the device sets it to one.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

OK (Bit 13) The OK bit indicates that the command was executed without error. If it equals one, no error occurred (command executed OK). If the OK bit is zero and the C bit is set, then an error occurred.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

Buffer Pointer This field is a 32-bit offset to the dump area address. The size of the dump area is 596 bytes.

Configuration parameters and contents of other registers are transferred from the CSMA/CD unit through the status FIFO by the Command Unit to memory. The CU performs the following sequence:

1. Starts the dump action command.
2. Writes the dump command byte to the transit FIFO.
3. Waits for the dump marker to return from the CSMA/CD module.
4. Dumps the FEXT and CSMA/CD registers content through the status FIFO.
5. Dumps the parallel registers.
6. Prepares the status word with C equal to 1 and the OK bit equal to 1.
7. Completes the action command.

Table 47 and Table 48 describe the dump area format.

Table 47. Dump Data Bytes (0-79)

Byte	D7	D6	D5	D4	D3	D2	D1	D0
0	FEXT RCV_WR Base Address Register (low)							
1	FEXT RCV_WR Base Address Register (high)							
2	FEXT RCV_WR Current Address Register (low)							
3	FEXT RCV_WR Current Address Register (high)							
4	FEXT RCV_RD Current Address Register (low)							

Table 47. Dump Data Bytes (0-79)

Byte	D7	D6	D5	D4	D3	D2	D1	D0
5	FEXT RCV_RD Current Address Register (high)							
6	FEXT RCV_RD Base Address Register (low)							
7	FEXT RCV_RD Base Address Register (high)							
8	FEXT EXEC_WR Current Address Counter (low)							
9	FEXT EXEC_WR Current Address Counter (high)							
10	FEXT EXEC_WR Base Address Register (low)							
11	FEXT EXEC_WR Base Address Register (high)							
12	FEXT EXEC_RD Current Address Counter (low)							
13	FEXT EXEC_RD Current Address Counter (high)							
14	FEXT EXEC_RD Base Address Register (low)							
15	FEXT EXEC_RD Base Address Register (high)							
16	FEXT RCV_WR Byte Counter (low)							
17	FEXT RCV_WR Byte Counter (high)							
18	FEXT EXEC_WR Byte Counter (low)							
19	FEXT EXEC_WR Byte Counter (high)							
20	FEXT EXEC_WR Initial Threshold Register (low)							
21	FEXT EXEC_WR Initial Threshold Register (high)							
22	FEXT EXEC_WR Current Threshold Register (low)							
23	FEXT EXEC_WR Current Threshold Register (high)							
24	Configure Byte 8							
25	Configure Byte 9							
26	Configure Byte 10							
27	Configure Byte 11							
28	Configure Byte 12							
29	Configure Byte 13							
30	Configure Byte 14							
31	Configure Byte 15							
32	Configure Byte 16							
33	Configure Byte 17							
34	Configure Byte 18							
35	Configure Byte 19							
36	Configure Byte 20							
37	Configure Byte 21							
38	Reserved							
39	Individual Address Register 1							
40	Individual Address Register 2							
41	Individual Address Register 3							
42	Individual Address Register 4							

Table 47. Dump Data Bytes (0-79)

Byte	D7	D6	D5	D4	D3	D2	D1	D0
43	Individual Address Register 5							
44	Individual Address Register 6							
45	Transmit Status (low byte)							
46	Transmit Status (high byte)							
47	Transmit CRC 0							
48	Transmit CRC 1							
49	Transmit CRC 2							
50	Transmit CRC 3							
51	Receive CRC 0							
52	Receive CRC 1							
53	Receive CRC 2							
54	Receive CRC 3							
55	Temporary Memory 0							
56	Temporary Memory 1							
57	Temporary Memory 2							
58	Temporary Memory 3							
59	Temporary Memory 4							
60	Temporary Memory 5							
61	Receive Status (low byte)							
62	Receive Status (high byte)							
63	Hash Register 0							
64	Hash Register 1							
65	Hash Register 2							
66	Hash Register 3							
67	Hash Register 4							
68	Hash Register 5							
69	Hash Register 6							
70	Hash Register 7							
71	X	X	X	X	X	X	X	X
72	X	X	X	X	X	X	X	X
73	X	X	X	X	X	X	X	X
74	1	1	1	1	1	1	X	X
75	Receive Length (high)							
76	Receive Length (low)							
77 – 79	Reserved							

Table 48. Dump Data Dwords (20-148)

<table>
<tr><th>Dword</th><th colspan="2">Byte 3</th><th colspan="2">Byte 2</th><th colspan="2">Byte 1</th></tr>
<tr><td>20 – 29</td><td colspan="6">Reserved</td></tr>
<tr><td>30</td><td colspan="6">Micromachine (MM) Register File 39</td></tr>
<tr><td>31</td><td colspan="6">Micromachine (MM) Register File 38</td></tr>
<tr><td>32</td><td colspan="6">Micromachine (MM) Register File 37</td></tr>
<tr><td>33</td><td colspan="6">Micromachine (MM) Register File 36</td></tr>
<tr><td>34</td><td colspan="6">Micromachine (MM) Register File 35</td></tr>
<tr><td>35</td><td colspan="6">Micromachine (MM) Register File 34</td></tr>
<tr><td>36</td><td colspan="6">Micromachine (MM) Register File 33</td></tr>
<tr><td>37</td><td colspan="6">Micromachine (MM) Register File 32</td></tr>
<tr><td>38</td><td colspan="6">Receive Short Frame Errors</td></tr>
<tr><td>39</td><td colspan="6">Receive CDT Errors Counter</td></tr>
<tr><td>40</td><td colspan="6">Receive Overrun Errors Counter</td></tr>
<tr><td>41</td><td colspan="6">Receive Resource Errors Counter</td></tr>
<tr><td>42</td><td colspan="6">Receive Alignment Errors Counter</td></tr>
<tr><td>43</td><td colspan="6">Receive CRC Errors Counter</td></tr>
<tr><td>44</td><td colspan="6">Receive Good Frames Counter</td></tr>
<tr><td>45</td><td colspan="6">Transmit Total Collisions Counter</td></tr>
<tr><td>46</td><td colspan="6">Transmit Multiple Collisions Counter</td></tr>
<tr><td>47</td><td colspan="6">Transmit Single Collisions Counter</td></tr>
<tr><td>48</td><td colspan="6">Transmit Deferred Counter</td></tr>
<tr><td>49</td><td colspan="6">Transmit Lost CRS Counter</td></tr>
<tr><td>50</td><td colspan="6">Transmit Underrun Errors Counter</td></tr>
<tr><td>51</td><td colspan="6">Transmit Late Collisions Errors Counter</td></tr>
<tr><td>52</td><td colspan="6">Transmit Maximum Collisions Errors Counter</td></tr>
<tr><td>53</td><td colspan="6">Transmit Good Frames Counter</td></tr>
<tr><td>54 – 64</td><td colspan="6">Reserved</td></tr>
<tr><td>65</td><td colspan="3">Reserved</td><td colspan="2">Execution Status Port</td><td></td></tr>
<tr><td>66</td><td colspan="6">SCB Port Register</td></tr>
<tr><td>67</td><td colspan="6">SCB Pointer register</td></tr>
<tr><td>68</td><td>SCB Interrupt Mask Byte</td><td>SCB CUC and RUC Byte</td><td colspan="2">SCB Interrupt Byte</td><td>SCB CUS and RUS Byte</td><td></td></tr>
<tr><td>69</td><td>SCB Interrupt Mask Byte</td><td>SCB CUC and RUC Byte</td><td colspan="2">SCB Interrupt Byte</td><td>SCB CUS and RUS Byte</td><td></td></tr>
<tr><td>70 – 81</td><td colspan="6">Reserved</td></tr>
<tr><td>82</td><td colspan="6">DMA Arbitration Registers</td></tr>
<tr><td>83 – 85</td><td colspan="6">Reserved</td></tr>
<tr><td>86</td><td colspan="6">Micromachine Register File 31</td></tr>
<tr><td>87</td><td colspan="6">Micromachine Register File 30</td></tr>
</table>

Table 48. Dump Data Dwords (20-148)

Dword	Byte 3	Byte 2	Byte 1
88	Micromachine Register File 29		
89	Micromachine Register File 28		
90	Micromachine Register File 27		
91	Micromachine Register File 26		
92	Micromachine Register File 25		
93	Micromachine Register File 24		
94	Micromachine Register File 23		
95	Micromachine Register File 22		
96	Micromachine Register File 21		
97	Micromachine Register File 20		
98	Micromachine Register File 19		
99	Micromachine Register File 18		
100	Micromachine Register File 17		
101	Micromachine Register File 16		
102	Micromachine Register File 15		
103	Micromachine Register File 14		
104	Micromachine Register File 13		
105	Micromachine Register File 12		
106	Micromachine Register File 11		
107	Micromachine Register File 10		
108	Micromachine Register File 9		
109	Micromachine Register File 8		
110	Micromachine Register File 7		
111	Micromachine Register File 6		
112	Micromachine Register File 5		
113	Micromachine Register File 4		
114	Micromachine Register File 3		
115	Micromachine Register File 2		
116	Micromachine Register File 1		
117	Micromachine Register File 0		
118	Micromachine LFSR		
119	Micromachine Bit Flag Array 6		
120	Micromachine Bit Flag Array 5		
121	Micromachine Bit Flag Array 4		
122	Micromachine Bit Flag Array 3		
123	Micromachine Bit Flag Array 2		
124	Micromachine Bit Flag Array 1		
125	Micromachine Bit Flag Array 0		

Table 48. Dump Data Dwords (20-148)

Dword	Byte 3	Byte 2	Byte 1
126	Micromachine Input Port 3		
127	Micromachine Input Port 2		
128	Micromachine Input Port 1		
129	Micromachine Input Port 0		
130	Micromachine ALU		
131	Micromachine Temporary B Register		
132	Micromachine Temporary A Register Rotated Right		
133	Micromachine Temporary A Register		
134	Transmit DMA Byte Count Register		
135	Micromachine Input Port Address Register		
136	Transmit DMA Address Register		
137	Micromachine Output Port Register		
138	Receive DMA Byte Count Register		
139	Micromachine Output Port Address Register		
140	Receive DMA Address Register		
141 – 142	Reserved		
143	DIU Control Register		
144	Micromachine ALU Shifted by Byte		
145	Reserved		
146	BIU Control Register		
147	Reserved		
148	Micromachine Status Register		

6.4.2.8 Diagnose (111b)

The diagnose command triggers an internal self-test procedure that checks the internal device hardware. Its format is illustrated below.

Figure 23. Diagnose Command Format

Offset	Command Word Bits 31:16					Status Word Bits 15:0					
00h	EL	S	I	0000000000	111	C	X	OK	X	F	XXXXXXXXXXX
04h	Link Address (A31:A0)										

Link Address This is the 32-bit address of the next command block. It is added to the CU base to obtain the actual address.

EL (Bit 31) If this bit is set to one, it indicates that this command block is the last one on the CBL. The CU will go from the active to the idle state after the execution of the CB is finished. This transition will always cause an interrupt with the CNA/CI bit set in the SCB.

S (Bit 30) If this bit is set to one, the CU will be suspended after the completion of this CB. A CNA interrupt will be generated if the device is configured for this. The CU transitions from the active to the suspended state after the execution of the CB.

I (Bit 29) If the I bit is set to one, the device generates an interrupt after the execution of the CB is finished. If I is not set to one, the CX interrupt will not be generated.

Bits 28:19 These bits are reserved and should all be set to 0.

CMD (Bits 18:16) This is the diagnose command, which has a value of 111b.

C (Bit 15) This bit indicates the execution status of the command. Software should reset this bit before issuing the command to the device. Following a command completion, the device sets it to one.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

OK (Bit 13) The OK bit indicates that the command was executed without error. If it equals one, no error occurred (command executed OK). If the OK bit is zero and the C bit is set, then an error occurred.

NOTE: The difference in the definition of the C bit for the transmit command (Section 6.4.2.5).

F (Bit 11) This bit indicates the results of the self-test procedure. A 0 indicates a pass, and a 1, failure.

The diagnose command checks the following internal device circuitry:

- Exponential backoff random number generator (linear feedback shift register).
- Exponential backoff time-out counter.
- Slot time period counter.
- Collision number counter.
- Exponential backoff shift register.
- Exponential backoff mask logic.

This procedure checks the operation of the backoff block, which resides in the serial side and is not easily controlled. The CU triggers the self-test procedure of the serial subsystem. It performs the following sequence:

1. Begins execution of the diagnose action command.
2. Waits for command completion.
3. Prepares the status word with the C bit equal to 1, OK equal to 1, and F equal to 0 if the diagnose succeeded. Otherwise, the status word has the C bit equal to 1, OK equal to 0, and F equal to 1.
4. Completes action command.

The CSMA/CD module performs the self test procedure in two phases: phase 1 tests the counters and phase 2 tests the trigger logic.

During Phase 1, the linear feedback shift register (LFSR), exponential backoff time-out, slot time, and collision counters are checked. The test is performed in the following manner:

1. All counters and shift registers are reset simultaneously.
2. The unit starts counting and shifting the registers.
3. The exponential backoff shift register reaches all ones.
4. The unit checks the exponential backoff shift register for all ones when the LFSR content is all ones in its 10 least significant bits.
5. The unit stops counting when the LFSR (30 bits) reaches a specific state, and the exponential backoff counter (10 bits) wraps from all ones to all zeroes. Simultaneously, the slot time counter switches from 01111111111 to 10000000000, and the collision counter (4 bits) wraps from all ones to all zeroes.
6. Phase 1 is successful if the 10 least significant bits (when applicable) of all four counters are all zero.

During Phase 2, the test is performed in the following steps:

1. The exponential backoff shift register, LFSR, and all counters are reset.
2. The exponential backoff logic is temporarily configured accordingly:
   a. SLOT-TIME = 3h
   b. LIN-PRIO = 6h
   c. EXP-PRIO = 3h
   d. BOF-MET = 0h
3. Transmission and collisions are emulated internally.
4. If the most significant bit of exponential backoff shift register is 0, then step 3 is repeated.

If Step 4 is successful, then a passed status is returned; otherwise, a failed status is returned.

6.4.3 Receive Operation

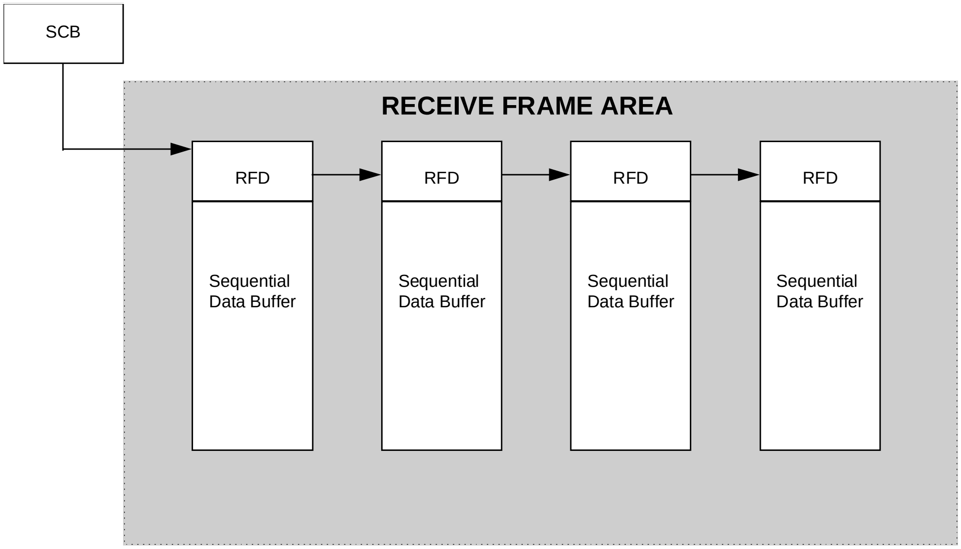
6.4.3.1 Receive Frame Area

The 8255x supports the concept of a receive frame area (RFA). The RFA is the list of free receive resources and consists of Receive Frame Descriptors (RFDs). The RFDs contain data buffers capable of holding maximum Ethernet size packets immediately following the RFD header. This constitutes the simplified memory model. Each receive frame is described by one RFD.

6.4.3.1.1 Simplified RFA Structure

In the simplified RFA structure, the data portion of the received frame (including the Ethernet header) is part of the RFD and is located in contiguous memory immediately after the size field in the RFD. The simplified memory structure is shown in the figure below.

Figure 24. Simplified Memory Structure



6.4.3.1.2 Receive Frame Descriptor Format

Figure 25. Receive Frame Descriptor Format

Offset	Command Word Bits 31:16						Status Word Bits 15:0			
00h	EL	S	000000000	H	SF	000	C	0	OK	Status Bits
04h	Link Address (A31:A0)									
08h	Reserved									
0Ch	0	0	Size				EOF	F	Actual Count	

EL (Bit 31) The EL bit indicates that this RFD is the last one in the RFA.

S (Bit 30) The S bit suspends the RU after receiving the frame.

H (Bit 20) The H bit indicates if the current RFD is a header RFD. If it equals 1, the current RFD is a header RFD, and if it is 0, it is not a header RFD.

NOTE: If a load HDS command was not previously issued, the device disregards this bit.

SF (Bit 19) The SF bit equals 0 for simplified mode.

C (Bit 15) This bit indicates the completion of frame reception. It is set by the device.

OK (Bit 13) The OK bit indicates whether the frame was received without any errors and stored in memory. If the last frame was received with sufficient memory space, the OK bit will be set, even if it was the last RFD in the RFA with the EL bit set. After receiving the frame, the device enters the no resource condition, generates an RNR interrupt, and starts discarding frames until the RU is restarted with sufficient resources.

Status Bits (Bits 12:0) This field contains the results of the receive operation:

Link Address The link address is a 32-bit offset to the next RFD. It is added to the RU base. The link address of the last frame can be used to form a cyclical link to the first RFD.

Size This field is used in the simplified mode and represents the data buffer size. In the header RFD, the size field identifies the data buffer size excluding the header area. The size value should be an even number.

EOF This bit is set by the device when it has completed placing data in the data area. Before a new RFD can be included in the RFA, the EOF bit must be cleared by software.

F This bit is set by the device when it updates the actual count field. Before a new RFD can be included in the RFA, the F bit must be cleared by software.

Actual Count The number of bytes written into the data area.

Table 49. RFD Status Bit Descriptions

Status Bit	Description
Bit 12	This bit is reserved.
Bit 11	CRC error in an aligned frame. This bit may be set only in the save bad frames mode.
Bit 10	Alignment error (CRC error in misaligned frame). This bit identifies the number of bits in the frame that were not an octet multiple. This bit may be set only in save bad frames mode.
Bit 9	Ran out of buffer space; no resources. This bit indicates that the incoming frame was larger than the possible receive data area for that frame. In simplified mode, this means that the size of the RFD was not large enough to accommodate the entire frame. **NOTE:** This status bit is not related to the RFD EL and S bits. Status bit 9 reflects the status of the current frame. The EL and S bits control the RU machine status after the reception of the current frame.
Bit 8	DMA overrun failure to acquire the system bus. This bit may be set only in save bad frames mode.
Bit 7	Frame too short. It implies that the length of the received frame was less than 64 bytes including the CRC. This bit may be set only in save bad frames mode.
Bit 6	This bit is reserved.
Bit 5	Type/Length. If this bit is set, it indicates that the received frame was a type frame (the value of the Ethernet header Type/Length field was either 0 or greater than 1500 decimal).
Bit 4	Receive Error. This bit is set if the RX_ER pin was asserted at least once during frame reception. It is set by the PHY when it detects a character error. The actual detection of character error depends on the PHY technology and the algorithm implemented. The device increments the device CRC error counter when this bit is set. This bit may be set only in save bad frames mode.
Bit 3	This bit is reserved.

Table 49. RFD Status Bit Descriptions

Status Bit	Description
Bit 2	No address match. If this bit is set to 1, the destination address of the received frame does not match the individual address, multicast address, or the broadcast address in the filter. For example, this bit will be set when the device is in promiscuous mode and the destination address does not match any of the other address filtering mechanisms.
Bit 1	IA match bit. When this bit equals 0, it implies that the destination address of the received frame matches the individual address. When it is 1, the destination address of the received frame does not match the individual address. For example, a multicast or broadcast address will set this bit to a 1.
Bit 0	Receive collision (82557 and 82558 only). When this bit is set, it indicates a collision was detected during a reception. TCO indication (82559 and later generation controllers). For the 82559 and later generation controllers, this bit no longer reflects collision detection. Instead, it indicates that the device is processing a TCO packet. In an environment with a TCO controller where receive to TCO is enabled, the controller introduces a new behavior. TCO packets are first posted to the RFA before they are transferred back to the TCO controller. In most cases this process is completely transparent to the software driver since the RFD that contains the TCO packet is reclaimed. However, if header RFDs are being used and the TCO packet size is larger than the header size, then the RFD is not reclaimed. In this case, a bad status indication will be posted to memory. The bad status indication should be used by the software driver to discard the packet.

Table 50. Actual Count in Header RFD

F	EOF	Actual Count
0	0	Invalid
0	1	Invalid
1	0	HDS size
1	1	Total byte count

6.4.3.2 Initial Receive Frame Area Structure

To enable the device to receive frames, software must setup the following structure:

1. The SCB general pointer in the SCB should point to the first RFD on the list.
2. The link offset of each RFD in the list should point to the next RFD.
3. The EL bit in the last RFD should be set.

6.4.3.3 Operation of Frame Reception

The serial subsystem of the device selects the frames destined for the station according to the destination address of the frames passing on the link. A frame is selected if it is at least 6 bytes long and its address matches either the individual address, multicast address, multiple IA, or broadcast address (promiscuous mode). It transfers the selected frames with their status to the receive FIFO. The receive DMA unit transfers the frame from the receive FIFO to host memory under control of the receive unit. If discard short frames is enabled, any receive frame shorter than 64 bytes on the link (including padding and CRC) will be completely discarded. Although these frames are discarded, they are still counted in the short frame counter.

For every frame, the RU configures a RFD in memory. The loading of each buffer is done by the receive DMA in parallel with pre-fetching the next buffer by the RU. After completing frame reception, the RU closes the last RFD and configures the structure for receiving the next frame.

6.4.3.3.1 Configuring the Next RFD

The RU performs the following sequence to set up a RFD.

1. Reads the 4 Dwords of the current RFD and saves the EL and S bits internally.
2. Analyzes the link offset of the current RFD and saves it as the address of the next RFD.
3. Initiates a receive DMA if the size of the data field in the RFD is greater than zero. The receive DMA is initiated with the address of the first byte of the destination address to the byte count specified by the RFD.
4. Forces a receive DMA completion if the size of the data field in the RFD is zero.
5. Goes to the buffer pre-fetch and transfer cycle.

6.4.3.3.2 Close Frame

When the RU reads an end of frame from the receive FIFO (indicating the end of a received or discarded frame), it performs the following sequence to close the frame:

1. Reads the status from the receive FIFO and saves it internally.
2. Skips to the completion of reception sequence if the RU is not in the ready state.
3. Reclaims the RFD if the frame status indicates that there is an error (including short frame) and the save bad frame configuration parameter is zero. In addition to reclaiming the RFD, the device backs up the current RFD to the next RFD and jumps to step 8. If a header RFD is being used and the header field was filled, the RFD is not reclaimed.
4. Writes C equal to 1, OK equal to 0, status bit 9 equal to 1, and the remaining bits from the CSMA/CD module to the status word of the current RFD if the RU ran out of resources during reception of this frame.
5. Writes C equal to 1 and the remaining bits from the CSMA/CD module to the status word if the RU did not run out of resources.
6. Requests FR interrupt.
7. Creates a new RFD if the S bit was not set and the RU did not run out of resources.
8. Goes to completion of reception of frame.

6.4.3.3.3 Completion of Reception

Reception completion occurs when the RU encounters an end of frame regardless of its state. The procedure is determined by the following: EL bit, S bit, RU start request, and RU resources. The following sequence is performed by the RU at the completion of reception:

1. If the RU ran out of buffers or frame descriptors during current frame reception, then the state changes to no resources. The RU requests an RNR interrupt, sets the internal S bit, and starts discarding frames.
2. If the S bit of current frame is set but the RU start request is not and the RU is in the ready state, an RNR interrupt request is initiated and the state is changed to suspended.
3. If the RU start request bit is set but the S bit is not, then the RU state is changed to ready and a new RFD is created.

4. If the RU start request and S bits are set, a new RFD is created and the state is changed to suspended.
5. If the RU state is not ready, frames should be discarded.
6. If the RU is in the ready state or has just exited the ready state, the following steps are performed:
   a. Update the SCB status word according to the new state and clear internal interrupt flags.
   b. Activate the hardware interrupt signal.

6.4.3.4 No Buffer Performance Improvements (82558 and 82559)

During normal receive operation, the 82558 and 82559 treat received packets in the same manner as the 82557. The data structures and driver-hardware interface remain unchanged. However, some internal modifications were made for performance improvement.

In a no resources situation, the 82558 and 82559 operate differently than the 82557. In this scenario, the 82557 starts discarding frames when it reaches a no resources or suspended state. The 82558 and 82559 start storing all incoming frames in the receive FIFO in these states. When the FIFO is full, the next frame is marked as an overrun frame while the following frames are lost. This enables the network device driver to react to the no resources situation and successfully receive the following two or more frames.

6.4.3.4.1 TCP/UDP Checksum Support (82559 only)

The 82559 provides a checksum word in the receive structure if it is configured to (Section 6.4.2.3, “Configure (010b)”). The 82557 and 82558 do not have this capability.

The checksum word is calculated on the incoming packet excluding the MAC header (first 14 bytes of the packet) and Ethernet CRC. The checksum word is always appended at the end of the data posted to the receive buffer(s) with the least significant byte first. If the 82559 is configured to post the Ethernet CRC into memory, then the checksum word follows the CRC. The byte count field in the receive memory structure(s) includes the checksum word. If software enables this feature, software must subtract 2 bytes from the reported length of the packet to determine the actual packet length.

If the incoming frame is a TCP or UDP packet, the device driver can accelerate the checksum word calculation using this capability in the 82559. The driver should subtract from the 82559 checksum word non-relevant fields within the packet, negate the result (1’s complement), and compare it to the packet’s TCP/UDP checksum word.

The 82559 calculates the checksum word as a sum of the incoming words of the received packet (not including the MAC header and CRC bytes). After the summation of each two words, the carry is added to the least significant bit of the result. If the packet has an odd byte count, the last byte is padded with 8 zeroes as its most significant bits. The following equation and example demonstrate this:

Example 1. 82559 Checksum Calculation

Assume the following incoming Packet: SA DA Type B_0 B_1 B_2 B_3 B_4 B_5 … B_{2N-2} B_{2N-1} B_{2N} CRC

Checksum = $\{ B_1B_0 + B_3B_2 + C_0 + B_5B_4 + C_1 + \ldots + B_{2N-1}B_{2N-2} + C_{N-2} + 00B_{2N} + C_{N-1} \}$,

where C_0, C_1, … C_{N-1} are the carry out results of the intermediate sum operations.

Example 2. Numerical Calculation

Assume the following incoming packet: SA DA Type F1 F5 54 79 E7 9E F5 CRC

S_0 = F5F1+7954 = 6F45,	C_0 = 1
S_1 = 6F45 + 9EE7 + 1 = 0E2D,	C_1 = 1
S_2 = 0E2C + 00F5 + 1 = 0F23,	C_2 = 0
Check Sum = S_3 = 0F21+ 0 = 0F23	

When all data bytes are written to the memory, the device writes the actual count. The device writes the frame status to the RFD status word. The device asserts an interrupt to indicate the end of receive processing. The driver can poll in memory for the frame status and mask the last interrupt.

6.5 Command Unit and Receive Unit Operation

6.5.1 Starting and Completing Control Commands

Software can issue control commands by writing to the RUC and CUC fields of the SCB command word. The SCB CU and RU command fields are two fields in the lower byte of the SCB command word, called the SCB command byte. Since the 8255x clears the SCB command byte when the control command is accepted:

- Software must wait for this byte to be cleared before the next control command can be issued.
- CU and RU control commands must never be issued together in the same SCB write cycle.

The 8255x does not necessarily accept the control commands immediately after they are written to the SCB since it may be engaged in higher priority tasks. For example, the device may be pre-fetching new buffers, handling buffer switches, or finishing frame reception. When the device is becomes available, it performs the start of control command sequence described below.

1. Reads the SCB command byte.
2. Reads the SCB general pointer.
3. The 8255x issues an internal request to the RU to perform an RU command acceptance sequence if the RUC field is not zero.
4. The acceptance sequence for a CU command is performed immediately if the CUC field is not zero.

After the CU and RU have completed the acceptance sequence, the 8255x updates the SCB status according to the internal CU status, RU status, and interrupt requests.

6.5.2 Generating and acknowledging interrupts

When the CPU is interrupted by the 8255x, it should acknowledge the interrupt by setting the corresponding acknowledge bits in the SCB interrupt acknowledge byte. When the CPU writes to this byte, the corresponding interrupt bits are immediately cleared. If all interrupt bits are cleared, the 8255x clears its INTA# line. If the interrupt service routine is likely to process all pending interrupts, then all the bits can be acknowledged in one PCI write cycle.

6.5.3 Command Unit Control

The CU is the 8255x logical unit that executes action commands from the command block list (CBL). This section describes how software controls the execution of action commands. Specifically, start, stop, suspend, or resume of the CU are discussed. The CU can be modeled as a logical machine that exists in one of the following states at any given time:

- **Idle.** The CU is currently not executing an action command and is not associated with a CB in the CBL. This is the initial state. It is also the state reached after the CU finishes executing a CBL where the last CB had an EL bit set. A CU start command must be issued to begin execution on a new CBL.
- **Suspended.** The CU is not executing a CB but has read a next link pointer in the last CB that it executed before it suspended execution. A CU resume command forces the 8255x to continue execution from the CB at the next link address.
- **Active.** The CU is currently executing an action command.

Software affects CU operation in two ways: issuing a CU control command or setting bits in the command word of the action command. This causes the CU to do one of the following:

- Start executing a list of action commands.
- Resume execution of a list of action commands.
- Stop execution after completing an action command. Usually, this is the last command block in the list.
- Suspend execution after completing an action command (if the S bit was set).
- Issue interrupts after completing action commands.

There are two important points of time in the execution of commands:

- **Acceptance Time.** This is the time following a write to the SCB command byte. It is when the CU reads the control command, clears it, and begins execution.
- **Completion of Execution.** This is when the CU completes executing a command.

The CU uses two internal flags to remember requests from acceptance time that are to be acted on at completion of execution: CU_START_REQUEST and CU_RESUME_REQUEST.

At command acceptance time, after the 8255x has finished higher priority tasks and detected a new command, the device reads and analyzes the SCB command byte. (Higher priority tasks can include: receive end of frame processing, receive or transmit buffer pre-fetching, completion of transmit DMA, completion of action commands, and dumping counters.) The CUC field indicates one of the following commands: CU_START or CU_RESUME.

6.5.3.1 CU Start Command

When the CU detects the CU Start (CU_START) command, it begins executing the first action command in the list. It is prohibited from issuing a CU_START if the CU is in the active state. This means that software must ensure that the CU is idle or suspended before issuing the CU_START command. Upon acceptance of a CU_START command, the CU initiates the following sequence:

1. Reads the CB offset from the SCB (general pointer register) and saves it as a pointer to the first CB in the CBL.
2. Becomes active and starts execution from the beginning of the CBL if the CU is not in the active state.

6.5.3.2 CU Resume Command

The CU Resume (CU_RESUME) command resumes CU operation. The 8255x completes the following sequence:

1. If the CU is in the suspended state it goes to the active state and requests the beginning of the next CB. Since the 8255x remembers the pointer to the next action command, it does not re-read the link pointer in the previous CB. However, the 8255x re-reads the S bit in the previous CB to determine if the driver has modified it while the device was suspended. If the S bit is cleared, it proceeds to execute the new CB. If the S bit is still set in the previous CB, the CU goes back to the suspended state.
2. If the CU is in the active state, it verifies the validity of the S bits in the current and next action commands. If the S bit is cleared in the current CB, it proceeds to the next CB in the list after execution of the current CB is completed.
3. If the CU is in the idle state, it ignores the CU_RESUME command.

6.5.3.3 CU Control Commands Response

The start and resume CU control (CUC) commands can be issued through the SCB CUC command field. The command block offset pointer in the general pointer register of the SCB points to the first executable action command. These action commands have a field for the end of list (EL) and suspend (S) bits. The EL bit indicates that the current action command is the last on the command list. The presence of the S bit indicates that the user wants the CU to enter the suspended state after executing the current command. Table 51 and Table 52 illustrate the various state transitions.

The mechanism used to suspend the CU is the S bit in the action command block. Suspending the CU through the action command block results in suspension after a specific command is executed. The CU can be re-activated by issuing a resume command.

Table 51. CU Control Commands: Actions at Acceptance Time

Present State	**CU Start**		**CU Resume**	
	Next State	Action	Next State	Action
Idle	Active	Start processing CB pointed to by SCB general pointer.	Idle	None.
Suspended	Active	Start processing CB pointed to by SCB general pointer.	Active	Start processing next CB if S bit is clear on current CB.
Active	Prohibited	Not applicable.	Active	Re-check S bit on current CB. If not set, process next CB.

Table 52. CU Activities Performed at the End of Execution

EL Bit	**S Bit**	**Next State**	**Action**
0	0	Active	Start processing CB
0	1	Suspended	CNA interrupt
1	0	Idle	CNA/CI interrupt
1	1	Idle	CNA/CI interrupt

6.5.4 Receive Unit Control

The receive unit (RU) is the logical unit that receives frames and stores them in memory. It uses free buffers and descriptors prepared by the CPU. This section describes how the CPU controls frame reception (starts, stops, suspends, and resumes the RU). Reception can also be halted due to a no resource condition.

The RU is modeled as a logical machine that takes one of the following states at any given time. Software can determine the current RU status by reading the SCB status word in the CSR (bits 5:2).

- **Idle (0000).** The RU has no memory resources and is discarding incoming frames. This is the initial RU state after reset.
- **No Resources Due to No More RFDs (0010).** The RU has no memory resources due to a lack of RFDs and is discarding incoming frames. This state differs from the idle state in that the RU accumulates statistics on the number of frames it has to discard. The 8255x enters this state after it processes an RFD that its EL bit set.
- **Suspended (0001).** The RU discards all incoming frames even though free memory resources exist to store incoming frames. The 8255x enters this state after it processes an RFD with its S bit set.
- **Ready (0100).** The RU has free memory resources and is ready to store incoming frames.

Table 53. RU Control Commands: Actions at Acceptance Time

		RU Start		RU Resume		RU Abort	
	Present State	**Next State**	**Action**	**Next State**	**Action**	**Next State**	**Action**
RU Not Actively Receiving	Idle	Ready	Set Up RFD	Idle	None	Idle	None
	No Resources due to no RFDs	Ready	Set Up RFD	Prohibited	N/A	Idle	None
	Suspended	Ready	Set Up RFD	Ready	Set Up RFD	Idle	None
	Ready	Prohibited	N/A	Prohibited	N/A	Idle	Start Discard - RNR Interrupt
RU Actively Receiving	Idle	Idle	Request Start	Idle	None	Idle	None
	No Resources due to No RFDs	No Resource due to No RFDs	Request Start	Prohibited	N/A	Idle	None
	Suspended	Suspend	Request Start	Suspend	Request Resume	Idle	None
	Ready	Prohibited	N/A	Prohibited	N/A	Idle	Abort DMA - Start Discard - RNR Interrupt

Frames arrive at the device independent of the state of the RU. When a frame is arriving, the 8255x is referred to as actively receiving, even when the RU is not in the ready state and the frame is being discarded.

Software can affect RU operation in three ways: by issuing an RU control command, by setting bits in the RFD. In general, software can cause the RU to do the following:

- Start frame reception.
- Resume reception if the RU is in the suspended state.
- Abort reception immediately and return to the idle state.
- Stop reception after a specified RFD is filled (frame received). This RFD is referred to by the 8255x as the last RFD on the list.
- Suspend reception after a specified RFD is filled (frame received).

The RU issues an interrupt after every received frame. The 8255x may issue two interrupts for one frame if the RU is using header RFDs. There are two important points of time in frame reception:

- **Acceptance time.** This is the time after an RU command is issued by software (written to the SCB command byte). It is when the RU reads the control command, takes initial action, and clears the SCB command byte.
- **Completion of Reception.** This is the time the RU finishes receiving or discarding an incoming frame.

The RU uses an internal RU start request flag to remember RU start requests from acceptance time that are to be acted on at completion of reception. The RU starts analyzing the command at acceptance time after the CU has passed the control command to the RU and finished accepting its own control command. The SCB RU control field can take one of the following values: RU_START, RU_ABORT, or LOAD_HDS.

6.5.4.1 RU Start Command

For the RU start (RU_START) command, the CPU activates the RU for frame reception. At acceptance time, the RU may or may not be actively receiving a frame. The RU performs the following when it is actively receiving a frame when an RU_START command is accepted:

1. Reads the RFA Offset word from the SCB and saves it internally as the pointer to the next RFD.
2. Sets the internal RU_START_REQUEST flag. When the current frame has been received or discarded the RU goes to the Ready state and sets up the next RFD.

When an RU_START command is accepted, the RU performs the following sequence when **NOT** actively receiving a frame.

Clears the internal RU_START_REQUEST flag.

Reads the RFA Offset word from the SCB and saves it internally as the pointer to the next RFD.

If the RU is not in the READY state and the DMA did not transfer any data to the current RFD, it does the following. It stops discarding, goes to the READY state, gives up the pre-fetched current buffers, and sets up a new RFD. Setting up a new RFD uses the pointer to the next RFD to prepare to receive the next frame.

6.5.4.2 RU Resume Command

The RU Resume (RU_RESUME) command resumes frame reception. The RU performs the following tasks:

1. The RU goes to the ready state and configures a new RFD if the RU is in the suspended state and not actively discarding a frame.
2. The RU sets the RU resume request flag if the RU is in the suspended state and the device is discarding a frame.
3. The RU ignores the command if it is not in the suspended state.

6.5.4.3 RU Abort Command

The RU abort (RU_ABORT) command immediately stops frame reception and enters the idle state.

1. The RU requests an RNR interrupt if the RU is in the ready state.
2. The RU stops all DMA activity and starts discarding incoming data.
3. The device transitions the RU to the idle state.

6.5.5 Updating SCB Status

The device updates the SCB status word when any of the following events occur.

1. When a control command is accepted. (The 8255x updates the SCB status and clears the SCB command word to indicate that acceptance has completed.)
2. When the CU or RU changes state.
3. After receiving a frame.
4. When a command with its I bit set is completed.

Note: The device may first clear the SCB command byte and return at a later time to update new status in the SCB status byte.

6.6 Flow Control

The 82557 does not have any support for flow control. However, the 82558 and later generation controllers support frame based (IEEE) flow control (FC). In addition, the 82558 also supports a proprietary PHY based mechanism for flow control (known as Bay flow control).

Flow control is implemented as a means of reducing the possibility of receive buffer overflows, which can result dropped received packets, and allows local control of network congestion levels. This can be accomplished by sending an indication to a transmitting station of an almost full receive buffer condition at a receiving station.

6.6.1 PHY Based Flow Control

The 82558 supports the PHY based flow control scheme known as the “Bay Flow Control” scheme. This scheme is supported only when the 82558 is operating using its internal PHY TX unit. It is not supported when an external PHY is used through the MII. The Bay Flow Control scheme is activated by setting bit 15 in MII register 16 in the 82558 PHY TX unit.

The 82558 generates flow control pause indications according to the number of bytes in its receive FIFO. This pause indication tells the 82558’s link partner to pause (or delay) future transmissions to allow the 82558 some time to empty its receive FIFO. The threshold at which the 82558 triggers the pause indication is determined by the flow control threshold register. PHY based flow control should not be used at the same time as frame based flow control.

Since frame based flow control is an IEEE 802.3x standard, it is recommended that software use the frame based FC method instead of PHY based FC.

6.6.2 Frame Based Flow Control

The 82558 and later generation devices support the frame based flow control scheme as specified in the IEEE 802.3x standard. Frame based flow control operates independently of the PHY. Thus, this method is supported with either the 8255x (except the 82557) internal PHY or an external PHY.

These devices also support a priority aware flow control scheme, which is a non-standard variation of frame based flow control and is described later in Section 6.6.3, “Priority Aware Frame Based Flow Control”.

Frame based flow control should be negotiated with a device’s link partner. This is accomplished through the N-Way auto-negotiation algorithm. To advertise frame based FC capability, the device should have their pause operation for full duplex (FDX) links bit set in the auto-negotiation advertisement register (MDI register 4, bit 10 [Section 7.2.4, “Auto-Negotiation Advertisement Register: Register 4”]). A device should enable the pause functionality only if both the local device and the link partner advertise this capability and the negotiated link utilizes a FDX technology (regardless of data rate).

The 82558 defaults to advertising that it is not capable of FDX FC. The 82559 defaults to advertising that it is capable of FDX FC. To enable this feature on the 82558, software needs to configure the corresponding bit in the ability advertisement register and force re-negotiation.

6.6.2.1 Protocol Description

IEEE 802.3x flow control (FC) is defined as point to point flow control. This means that it is relevant only for two devices connected by a dedicated link. This form of flow control is implemented in full duplex only. It includes a set of commands (and indications) for stopping and re-starting frame transportation between the two connected devices. The commands are encoded into special frames.

Flow control frames are the minimum Ethernet frame length (64 bytes). The flow control frame includes the following fields:

> Special DA (6 bytes), SA (6 bytes), Type (2 bytes), Command (2 bytes), Parameters (2 bytes), Pad (42 bytes), CRC (4 bytes).

Table 54. Flow Control Frame Format

S	one byte
Pre-amble	up to 6 bytes
SFD	one byte
Destination Address	6 bytes
Source Address	6 bytes
Type/Length	2 bytes
MAC Control Opcode	2 bytes
MAC Control Parameters	
FCS	4 bytes
T	one byte

A flow control frame is identified by a special type field (bytes transmitted left to right): 88 to 80. The reception of a FC frame can be done either through the regular individual address filtering mechanism or by a special multicast address detection mechanism. Since the transmitting MAC may not know the individual address of the other station, it may not be able to send FC frames to the individual address of the other node. A special multicast destination address is defined for use in FC frames. Special address filtering is used to receive frames addressed to that multicast address. The address is (bytes transmitted left to right): 01-80-C2-00-00-01.

6.6.2.2 Pause Operation

When a station's receive buffer reaches a high watermark (running out of storage space for incoming packets), it may send its link partner a pause frame (also known as an XOFF frame).

When the station receives the pause request, it stops transmitting for a number of slot times as specified by the time parameter in the pause frame. The time parameter is a 2 byte field.

The reaction time for a transmitting station (the congester) is 512 bit times from the end of the received pause frame. If the congester is transmitting when the time expires, it should complete the transmitted frame. The congester should not start transmitting a frame after the 512-bit time period. Transmission is restarted upon expiration of the pause timer or upon reception of a pause request with a time parameter of zero (also known as an XON frame).

6.6.2.3 Flow Control Functionality

Note: The 82557 does not support flow control functionality.

Flow control is set by three configuration bits: one for transmit and two for receive flow control. The default setting is off (Section 6.4.2.3, "Configure (010b)"). Software can interrogate the N-Way registers to determine whether to turn it on. Flow control cannot be activated by auto-negotiation alone.

The 8255x (except the 82557) provides status information to software regarding flow control. If the device is currently paused from sending frames or if it is sending a pause frame, it sets an indication in the flow control register. The device also has statistical counters that count the number of pause requests sent and received (Section 6.3.2.4, "Statistical Counters").

6.6.2.3.1 Transmit Flow Control

In FC mode, the controller receives FC frames independently of the state of the receive FIFO. This allows other stations to put pressure on the device. The received FC frame is passed to the FIFO only if it is configured for this (according to the reject FC bit, Section 6.4.2.3, "Configure (010b)") and only if there is room for it. The reject FC bit causes the device to reject frames sent to the special FC multicast address. However, if the FC frame was destined to the device individual address, the frame is passed to the FIFO regardless of the reject FC bit.

After a pause request is received, a 16 bit count down register is loaded with the time parameter value that was in the pause frame. The counter decrements each slot time and stops when it reaches zero. Upon receiving another pause frame, the device reloads the counter with the new time parameter.

If the value of the counter is not zero, transmission of future frames is inhibited. This does not interfere with the current transmission. The currently transmitted frame is completed and further transmits are paused. This does not prevent the device from transmitting FC frames itself if it is required to.

6.6.2.3.2 Receive Flow Control

In FC mode, the controller detects potential overrun conditions and sends a pause frame to the other node. The transmission of the pause frame occurs after the current transmit is completed (there is no immediate termination of transmissions).

The time parameter for the pause frame assembly is based on a pre-configured value that comes from the configure command.

The prediction of overruns occurs through detection of a threshold crossing in the receive FIFO, as defined in Section 6.3.8, "Flow Control Register". When the receive FIFO is filled beyond threshold value, a pause command is sent (with the pre-configured time value). Once a predicted overrun is detected, the device remains in the congested state until the receive FIFO is absolutely empty (no bytes in the receive FIFO).

The device has two additional configurable modes that affect transmit flow control. These modes are "ReStop" and "ReStart" and are detailed below:

- **ReStop.** The controller has sent a pause command. Just before pause time (sent by the device) expires, if the receive FIFO is not empty, the device sends another pause command. The additional pause is sent so that it is assured to reach the congester before the pause time expires.
- **ReStart.** The controller has sent a pause command. When the receive FIFO is empty, an XON (PAUSE(0)) command is sent. The same mechanism used for sending pause commands is used for sending pause (time equals 0) commands. In this mode, the device sends a pause command and waits for the receive FIFO to empty. When it is empty, the device sends an XON command

immediately. If the pause time (from the device) has expired, the controller starts monitoring the FIFO again to determine if a new pause command should be sent.

The two modes, ReStop and ReStart, can be used together. The controller sends a pause command. If the receive FIFO is empty, the device sends an XON command. If the time-out interval is exceeded, it sends another pause command.

In the ReStop mode, after the 8255x (excluding the 82557) transmits a pause command, the device may be required to extend the pause time by sending subsequent pause frames. The device is required to send subsequent pause commands far enough in advance to insure that it reaches the congester before its own timer expires. To achieve this, the device needs to attempt to transmit the pause command more than 27 slot times before the timer expires. These 27 slot times account for the time period required to transmit the pause command (1 slot time), the propagation delay (1 slot time), the congester’s reaction time (1 slot time), and a possible delay in the transmission of the pause command due to the transmission of a data frame (up to 24 slot times). To guarantee this time delta, the device driver needs to set the FC delay configuration value with its least significant 5 bits exceeding 27. The controller ignores these 5 bits in its internal counting. In order to guarantee that the time delta is sufficient to prevent transmission from the congester, it is recommended to set the least significant bits to 1Fh (31 decimal).

If the pre-configured value is less than 32, then:

- In ReStop mode (or both modes active), the device sends pause commands continuously until its receive FIFO is empty.
- In ReStart mode, the device sends one pause command without any following XON commands.

Two examples are presented below:

Example 3. Configured “FC Delay” (011Fh)

- — Send a pause command with 011Fh (287 decimal) slot time parameter.
- — Load counter with 100h (256 decimal) value.
- — ReStop Mode: After 256 slot times, if the FIFO is not empty, send another pause command.
- — ReStart Mode: When the FIFO is empty (presumably before 256 slot times), send an XON command.

Example 4. Configured “FC Delay” (0017h)

- — Send a PAUSE command with 17h (23 decimal) slot time parameter.
- — Do not load the counter.
- — ReStop Mode (not recommended): If the FIFO is not empty, send another pause command immediately (this will saturate the link).
- — ReStart Mode: Nothing further should be done.

For convenience, flow control related configuration bits (Section 6.4.2.3, “Configure (010b)” contains more detail) are described in the table below:

Table 55. Flow Control Configuration Bits

Name	Configuration Byte Map Location	Description	Default
Transmit FC	Byte 19, bit 2	IEEE frame based transmit flow control.	0 (on)
Receive FC ReStart	Byte 19, bit 4	IEEE frame based receive flow control - ReStart mode.	0 (off)
Receive FC ReStop	Byte 19, bit 3	IEEE frame based receive flow control - ReStop mode.	0 (off)
Reject FC	Byte 19, bit 5	When this bit is set, FC frames will not be passed to the receive FIFO like regular frames.	0 (off)
FC Delay	Byte 16, bits 7:0 (LSB) Byte 17, bits 7:0 (MSB)	This is the slot time delay number used for the time parameter in the assembly of pause frames (for pausing the other node transmissions).	4000h

6.6.3 Priority Aware Frame Based Flow Control

The 82558 and later generation controllers have the ability to respond to priority aware frame based flow control frames. Their operation relates to multiple queues.

6.6.3.1 Priority Flow Control Operation

The 82558 and later generation controllers can receive two types of flow control frames:

- Pause. The normal IEEE pause frames that stop all transmission (as discussed above).
- Pause Low. This is a new low priority pause frame that stops only the low priority queue (LPQ).

When the device receives a pause frame, it stops all transmission at the CSMA level as defined in IEEE draft standard 802.3x.

When the controller receives a pause low frame, it stops transmitting the LPQ at the micromachine level. When a pause low flow control frame is received, the device continues to transmit all frames in its transmit FIFO. When the device is used with priority aware FC it is recommended that the two frames in FIFO configuration bit is set so that the number of frames is no more than two.

Note: The 82558 and 82559 do not transmit pause low frames.

When the device receives a flow control frame (either a pause frame or a pause low frame), this frame overrides any flow control frame previously received. If the device was paused due to a pause frame and it receives a pause low frame, it starts transmitting high priority frames (and any low priority packets that were in the transmit FIFO). If the device was paused low due to a pause low frame and it receives a pause frame, it stops transmitting high priority frames immediately.

The pause and pause low frames do not affect the device’s CU state (as reflected in the CUS field in the SCB).

6.6.3.2 Flow Control Frame Format

The pause and pause low flow control frames share the same frame format. Table 54 on page 110 illustrates a flow control frame.

The priority field is the field that differentiates between pause and pause low frames. Only the three least significant bits in this byte are considered. These three bits are compared to the FC priority threshold configuration field:

- If the received value is equal to or less than the configured value, the frame is a pause frame.
- If the received value is greater than the configured value, the frame is a pause low frame.

This scheme enables the link partner to support up to 8 different priority queues. If a queue is full, the link partner should send a FC frame with the number of the queue in the priority field. The device is guaranteed to stop transmitting to that queue and lower priority queues.

Some examples are listed below:

- Assume the configured FC priority threshold value is 4 (100b).

  Any FC frame with the values 000b through 100b in the priority field will cause the device to pause (stop all transmissions). Any FC frame with the values 101b through 111b in the priority field will cause the device to pause low (stop only transmissions in the LPQ).
- Assume the configured FC priority threshold value is 0 (000b).

  Any FC frame with the values 000b in the Priority field will cause the device to pause (stop all transmissions). Any FC frame with the values 001b through 111b in the Priority field will cause the device to pause low (stop only transmissions in the LPQ).
- Assume the configured FC priority threshold value is 7 (111b).

  Any FC frame with any value in the priority field will cause the device to pause (stop all transmissions). This is the IEEE compliant mode.

6.6.4 Half Duplex Flow Control

The 82558 and 82559 support frame based flow control frames in both full duplex and half duplex switched environments. It is not intended for flow control to be enabled in a shared media environment. A flow control frame may encounter a collision in half duplex and will be retransmitted after the backoff time. Wait After Win (described in Section 6.7, “Collision Backoff Modification in Switched Environments”) guarantees that the flow control frame will be transmitted within a bounded delay.

6.7 Collision Backoff Modification in Switched Environments

The 82558 and 82559 support a modification of the CSMA/CD backoff algorithm named “Wait After Win” (WAW). When WAW is enabled, the device extends the interframe spacing gap to one slot time after the successful retransmission of a frame that previously encountered a collision. WAW is activated by a separate WAW configuration bit.

The WAW feature, combined with the “Half Slot Time BackOff” feature in the switch, guarantees that any collision that occurring on a dedicated half duplex link will be resolved after the first collision.

Physical Layer Interface 7

Intel Fast Ethernet adapters all have a physical layer (PHY) component that interfaces the network adapter to the wire. The MAC component of the adapter interfaces to the PHY component via the IEEE Media Independent Interface (MII). Sometimes it is necessary for software to access these PHY registers to properly configure the PHY. The Management Data Interface (MDI) in the 82557, 82558 and 82559 allows communication across the MII bus to registers on these PHYs.

For 100Mbps applications, the 82557 contains an IEEE MII compliant interface to the Intel 82555 physical layer device (or other MII compliant PHYs) which allow connections to both 10Mbps and 100Mbps networks. The 82558 and 82559 contain an embedded 82555 module. Software still communicates with the embedded 82558 through the MDI port. The MDI register sets for several different PHYs, including the 82555, are included in Section 7.2, “MDI Register Set” on the Physical Layer Interface.

PRO/100B adapters use both Intel (and third-party) PHYs that support 10BASE-T, 100BASE-TX, and/or 100BASE-T4 physical layers and are capable of auto-negotiation. The PHY module on the PRO/100B adapter is a separate discreet component from the 82557. There are different versions and generations of the PRO/100B that use different PHYs. Because certain vendor-specific programming hooks may be required to fully support various PHYs, software should determine at runtime which specific PHY is on the PRO/100B adapter being driven (Section 8.1.2, “PHY Detection and Initialization” contains more details).

PRO/100+ adapters use the 82558’s embedded 82555, which supports auto-negotiation, 10BASE-T, and 100BASE-TX. The 82559 also contains an embedded 82555. Although the PHY is embedded in the 82558 and 82559, software still accesses the PHY via the MDI interface in the manner that software uses on 82557 based adapters.

This section includes information on MDI, the 82555 MDI register sets, and auto-negotiation (N-Way) functionality. It also includes information for items specific to working with the 82555 TX PHY as well as the 82558 and 82559 embedded PHYs.

7.1 Management Data Interface (MDI)

The 82553, 82555, and other MII compliant devices provide status and accept management information via the Management Data Interface (MDI). This is accomplished via read and write operations to various registers according to the IEEE 802.3u MII specification. A read or write of a particular register is called a management frame, which is sent serially over the MDIO pin synchronous to MDC. Read and Write cycles are from the perspective of the controller. Therefore, the controller would always drive the Start, Opcode, PHY Address and Register Address on to the MDIO pin. For a write, the controller would also drive the transition bits and the data. For a read, the PHY drives the transition bits and data onto the MDIO pin. The controller should drive address and data on the falling edge of MDC and the PHY latches that data on the rising edge of MDC. In an application where only one PHY is present, the PHY uses a default PHY address of 00001b. The management frame structure is as follows:

Figure 26. Management Frame Structure

Start	Opcode	PHY Address	Register Address	Transition	Data
READ	< 01 > < 10 >	< AAAAA >	< RRRRR >	< Z0 >	16 Bits
WRITE	< 01 > < 01 >	< AAAAA >	< RRRRR >	< 10 >	16 Bits

This structure allows a controller, or other management hardware, to query the PHY for the status of the link or configure the PHY to one of many modes. The next section discusses the MDI registers.

7.2 MDI Register Set

The generic MDI register set is defined as follows:

Table 56. MDI Register Set

Register Address	Register Name and Function
00000	Control Register (MDI Standard Register)
00001	Status Register (MDI Standard Register)
00010	PHY Identification Register (Word 1)
00011	PHY Identification Register (Word 2)
00100	Auto-Negotiation Advertisement Register
00101	Auto-Negotiation Link Partner Ability Register
00110	Auto-Negotiation Expansion Register
00111-01111	Reserved

The Intel® 82555-specific (and thus 82558 and 82559 specific as well) MDI registers are listed in the table below. (These registers also apply to the 82558 and 82559.)

Table 57. 82555 MDI Register Set

Register Address	Register Name and Function
10000	Status and Control
10001	Special Control
10010	Clock Synthesis Test and Control
10011	100BASE-TX Receive False Carrier Counter
10100	100BASE-TX Receive Disconnect Counter
10101	100BASE-TX Receive Error Frame Counter
10110	Receive Symbol Error Counter
10111	100BASE-TX Receive Premature End of Frame Error Counter
11000	10BASET Receive End of Frame Error Counter
11001	10BASE-T Transmit Jabber Detect Counter
11010	Equalizer Control and Status
11011	Special Control

The individual registers are defined in the following subsections using the following conventions:

R: Read
W: Write
RO: Read only
SC: Self clearing

Note: The default values listed for the 82555 registers also apply to the registers in the embedded PHY modules of the 82558 and 82559.

7.2.1 Control Register: Register 0

Bit	Name	R / W	Description	Default
15	Reset	R/W SC	1 = PHY reset 0 = normal operation	0
14	Loopback	R/W	1 = loopback mode 0 = normal operation	0
13	10/100	R/W	1 = 100 Mbps 0 = 10 Mbps	1
12	Auto-Negotiation Enable	R/W	1 = Enable Auto-Negotiation 0 = Disable Auto-Negotiation	1
11	Power Down	R/W	1 = power down 0 = normal operation	0
10	Isolate	R/W	1 = electrically isolate PHY from MII 0 = normal operation	0
9	Restart auto configuration	R/W SC	1= Restart the auto negotiation 0 = Normal operation	0
8	Full/Half	R/W	1= Full duplex 0 = Half duplex	1
7	Collision Test Enable	R/W	1 = enable the collision CDT test during loopback	0
6:0	Reserved	R/W	Written as 0, don't care on read	0

7.2.2 Status Register: Register 1

Bit	Name	R / W	Description	Default
15	T4 capable	RO	1= T4 capable 0 = Not T4 capable	0
14	TX full duplex capable	RO	1= TX full duplex capable 0 = Not TX full duplex capable	1
13	TX half duplex capable	RO	1 = TX half duplex capable 0 = Not TX half duplex capable	1
12	10BASE-T full duplex capable	RO	1 = 10BASE-T full duplex capable 0 = Not 10BASE-T full duplex capable	1
11	10BASE-T half duplex capable	RO	1 = 10BASE-T half duplex capable 0 = Not 10BASE-T half duplex capable	1
10:6	Reserved	RO	These bits are reserved and written as 0.	0
5	Auto-negotiation complete	RO	1 = Auto-negotiation complete 0 = Auto-negotiation not complete	Sticky bit
4	Remote fault detect	RO	1 = Remote fault detected 0 = Remote fault not detected	0
3	Auto-negotiation ability	RO	1 = Auto-negotiation enabled 0 = Auto-negotiation disabled	1
2	Link Status (LINKFLT)	RO	1 = Valid link 0 = Invalid link	Sticky bit
1	Jabber Detect (JABFLT)	RO	1 = Jabber condition detected 0 = No jabber condition detected **NOTE:** This bit is only valid in 10 Mbps mode.	Sticky bit
0	Extended Capabilities	RO	1= Extended register capabilities 0 = Basic register set capabilities only	1

7.2.3 Identification Registers: Registers 2 and 3

The 32-bit ID register provides a mechanism for software to determine which PHY is present. The contents of these registers differ depending on the PHY. There are three values encoded in registers 2 and 3 that uniquely identify the PHY device:

- The OUI of the PHY manufacture.
- The model number of the PHY.
- The revision number of the PHY.

The OUI is a 24-bit manufacturer identification number. Only the least significant 22 bits of the OUI are stored in the registers. The two most significant bits, bits 23:22, are fixed at 0. Bits 21:6 of the OUI are stored in register 2 while the remaining bits of the OUI, bits 5:0, are stored in bits 15:10 of register 3. (The Intel manufacturer OUI number is “00AA00h.”)

The model number is a 6-bit value assigned by the manufacturer identifying the PHY model. It is stored in bits 9:4 of register 3. (The Intel PHY model number is “010101.”)

The revision number is a 4-bit value assigned by the manufacturer identifying the PHY revision number. The Intel devices use revision numbers “0000” through “0100.” The Intel 82558 has a revision ID number of 0000b, and the 82559, 0100b.

The map below shows how the these three numbers (OUI, model and revision numbers) are mapped into the MDI registers.

Table 58. 24-bit OUI Identification Number

Manufacturer	OUI	OUI			Model	Revision
		23:22	21:6	5:0		
Intel	00AA00h	00b	00 0000 1010 1010 00b	00 0000b		
Intel 82555	00AA00h	00b	00 0000 1010 1010 00b	00 0000b	010101b	0000b

The PHY identification register values are as follows:

Table 59. MDI Identification Registers 2 and 3: PHY ID Encoding

MDI Register	Intel 82555 (Revision 4)	Intel 82555 (Revision 1)	Intel 82555 (Revision 0)
Register 2	02A8h	02A8h	02A8h
Register 3	0154h	0151h	0150h

7.2.4 Auto-Negotiation Advertisement Register: Register 4

This register contains the advertisement ability of the PHY. It is used by software to determine the highest common denominator technology after the auto-negotiation process has finished. Any changes to this register prior to auto-negotiation must be followed by setting the Renegotiate bit in the Command register.

Bit	Name	R / W	Description	Default
15	Next Page	R/W	Next page is not supported and this bit should be set to 0b.	0
14	Acknowledge	RO	This bit position in the transmitted code word is used to indicate an acknowledge. It is set only by the auto-negotiation logic after receiving three consecutive and matching code words from the link partner.	0
13	Remote Fault	R/W	This bit indicates remote fault in the local station. It may be set by management to indicate the remote fault condition of the partner.	0
12:11	Reserved	RO	These bits are reserved.	000
10	Pause	RO	This bit indicates that Pause is supported by the local node.	82558: 0 82559: 1
9	100BASE-T4[a]	R/W	This field indicates that 100BASE-T4 is supported by the local node.	0
8	100BASE-TX Full Duplex[a]	R/W	This indicates that 100BASE-TX full duplex is supported by the local node.	1
7	100BASE-TX[a]	R/W	This bit indicates that 100BASE-TX is supported by the local node.	1

Bit	Name	R / W	Description	Default
6	10BASE-T Full Duplex[a]	R/W	This bit indicates that 10BASE-T full duplex is supported by the local node.	1
5	10BASE-T[a]	R/W	This bit indicates that 10BASE-T is supported by the local node.	1
4:0	Selector Field	R/W	This field identifies the protocol supported. IEEE 802.3 is indicated by a selector value 00001b.	00001

[a] During normal operation, the driver (management agent) does not need to change this register value. If a certain ability is not advertised, the respective bit in the Technology Ability field must be cleared. A bit that is not supported by the device must not be set within this field. Otherwise, the auto-negotiation protocol will be violated.

7.2.5 Auto-Negotiation Link Partner Ability Register: Register 5

This register holds the link code word captured from the link partner’s PHY (in other words, the device at the other end of the link segment). Its value is valid only when the Auto-Negotiation Complete bit is set in the Status register.

Bit	Name	R / W	Description	Default
15	Next Page	RO	Next page is not supported.	0
14	Acknowledge	RO	This bit indicates if a device has successfully received the link code word from its link partner.	0
13	Remote Fault	RO	This bit identifies if the remote fault bit is set in the received code word.	0
12:11	Technology Ability Field	RO	These bits are reserved.	000
10	Pause	RO	This bit indicates if Pause is supported by the link partner.	0
9	100BASE-T4[a]	RO	This bit indicates if 100BASE-T4 is supported by the link partner.	0
8	100BASE-TX Full Duplex[a]	RO	This bit indicates if 100Base-TX full duplex is supported by the local node.	0
7	100BASE-TX[a]	RO	This bit indicates if 100BASE-TX is supported by the link partner	0
6	10BASE-T Full Duplex[a]	RO	This bit indicates it 10BASE-T full duplex is supported by the link partner.	0
5	10BASE-T[a]	RO	This bit indicates if 10BASE-T is supported by the link partner.	0
4:0	Selector Field	RO	This field contains the Selector value from the link partner.	00000

[a] During normal operation, the driver (management agent) does not need to change this register value. If a certain ability is not advertised, the respective bit in the Technology Ability field must be cleared. A bit that is not supported by the device must not be set within this field. Otherwise, the auto-negotiation protocol will be violated.

NOTE: The Auto-Negotiation Link Partner Ability Register is read only.

7.2.6 Auto-Negotiation Expansion Register: Register 6

Register 6 contains supplemental information used by the auto-negotiation process.

Bit	Name	R / W	Description	Default
15:5	Reserved	RO	These bits are reserved and should be set to 0.	0
4	Parallel Detection Fault	RO	1 = More than one of the 10BASE-T, 100BASE-TX, or 100BASE-T4 PMAs detects a valid link. 0 = One or zero of the 10BASE-T, 100BASE-TX, or 100BASE-T4 PMAs detects a valid link.	0
3	Link Partner Next Page Able	RO	1 = Link partner supports Next Page. 0 = Link partner does not support Next Page.	0
2	Next Page Able	RO	1 = Local device supports Next Page. 0 = Local device does not support Next Page.	0
1	Page Received	RO	1 = Three identical and consecutive link code words have been received. 0 = Three identical and consecutive link code words have not been received.	0
0	Link Partner Auto-Negotiation Able	RO	1 = Link partner supports auto-negotiation. 0 = Link partner does not support auto-negotiation.	0

7.3 Intel 82555 Specific Registers

Note: The Intel MAC/PHY silicon devices (82558, 82559, 82550, and 82551) use the 82555 as the base for their integrated PHY units. Therefore, the information contained in this section and the following subsections apply to the all 8255x Fast Ethernet controllers except the 82557.

7.3.1 Status and Control Register: Register 16

Bit	Name	R / W	Description	Default
15	Flow Control	RW	1 = Flow control enabled. 0 = Flow control disabled. **NOTE:** This bit should always equal 0 for the 82559.	0
14	T4 Enable	RW	This bit enable T4 when auto-negotiation is disabled. 1 = Enable T4 technology 0 = Disable T4 technology	0
13	CRS Disconnect Control	RW	This bit controls the RX100 CRS disconnect function in repeater mode. **NOTE:** This bit should always equal 0 for the 82559.	0 = DTE 1 = Rptr
12	Reserved		This bit is reserved.	0
11	RCV De-Serializer In-Sync Indication	RO	This bit is used as the 100BASE-TX RCV De-Serializer In Sync Indication	0
10	100 Power Down	RO	This bit provides 100BASE-T Power Down Indication. 1 = Power down 0 = Normal operation	0
9	10 Power Down	RO	This bit provides 10BASE-T Power Down Indication. 1 = Power down 0 = Normal operation	0
8	Polarity	RO	This bit indicates 10BASE-T polarity. 1 = Reverse polarity 0 = Normal polarity	0
7:3	Reserved		These bits are reserved.	0
2	T4	RO	This bit is a result of the auto-negotiation process. 1 = 100BASE-T4. 0 = No 100BASE-T4	0
1	Speed	RO	This bit is a result of the auto-negotiation process. 1 = 100 Mbps 0 = 10 Mbps	0
0	Duplex	RO	This bit is a result of the auto-negotiation process. 1 = Full duplex 0 = Half duplex	0

7.3.2 Special Control Register: Register 17

Bit	Name	R / W	Description	Default
15	Scrambler Bypass	RW	1 = Bypass Scrambler 0 = Normal operation	0
14	4/5 Bypass	RW	1 = Bypass 4-bit to 5-bit 0 = Normal operation	0
13	Force Transmit H Pattern	RW	1 = Force H pattern 0 = Normal operation	0
12	Force 34 Transmit Pattern	RW	1 = Force 34 pattern 0 = Normal operation	0
11	Good Link	RW	1 = 100BASE-TX good link indication, forcing to ASD output 0 = Normal operation	0
9	MDI Tristate	RW	1 = Tri-state MDI interface 0 = Normal operation	0
8	Dynamic Power Down Disable	RW	1 = Disable Dynamic Power Down. 0 = Normal operation	0
7	Auto-Negotiation Loopback	RW	1 = Auto-negotiation loopback 0= Auto negotiation normal Mode.	0
6	Reserved		This bit is reserved.	0
5	Filter Bypass	RW	1 = Bypass filter 0 = Normal operation	0
4	Auto-Polarity Disable	RW	1 = Disable auto-polarity 0 = Normal operation	0
3	Squelch Test Disable	RW	1 = Disable 10BASE-T squelch test 0 = Normal squelch operation	0
2	Extended Squelch	RW	This bit indicates extended squelch control	0
1	Link Integrity Disable	RW	1 = Disable link integrity operation 0 = Normal link integrity operation	0
0	Jabber Function disable	RW	1 = Disable jabber function 0 = Normal jabber operation.	0

7.3.3 Clock Synthesis Test and Control Register: Register 18

Bit	Name	R / W	Description	Default
15	Clock Timing	RW SC	Clock Synthesizer Shift command. One shot signaling YS10ACLK domain. Can be active only when bit 14 is ‘0	
14	Clock Timing	RW SC	Clock Synthesizer load command. One shot signaling YS10ACLK domain. Can be active only when bit 15 is ‘0	
13	Break Down Timer Enable	RW	Logic 1 enables manipulate Break Down counter with phya1, phya4 and test high.	
12	Equalizer Probe Mode Enable		Logic 1 enables the Equalizer output through the Speed LED.	0
11	10BASE-T Probe Mode Enable	RW	1 = Enable 10BASE-T dig outputs through the LEDs **NOTE:** This function is only present on the 82559.	0
10:8	Reserved		These bits are reserved.	0
4:0	PHY Address	RO	This field contains the PHY address.	00001

7.3.4 100BASE-TX Receive False Carrier Counter: Register 19

Bit	Name	R / W	Description	Default
15:0	Receive False Carrier	RO SC	This register contains a 16-bit counter for false carrier events. A false carrier event occurs when a frame that does not start with “JK” is detected. When the counter is full, additional false carrier events are not counted. This counter is self-clearing on read.	0

7.3.5 100Base-TX Receive Disconnect Counter: Register 20

Bit	Name	R / W	Description	Default
15:0	Disconnect Event	RO SC	This register contains a 16-bit counter for disconnect events. The counter is incremented for each frame detected in repeater mode that does not start with a “JK.” When the counter is full, additional disconnect events are not counted. This counter is self-clearing on read.	0

7.3.6 100BASE-TX Receive Error Frame Counter: Register 21

Bit	Name	R / W	Description	Default
15:0	Receive Error Frame	RO SC	This register contains a 16-bit counter for receive error frames. It is incremented for frames with a receive error condition (frames containing a symbol error or frames with a premature end of frame). When the counter is full, additional error frames are not counted. This counter is self-clearing on read.	0

7.3.7 Receive Symbol Error Counter: Register 22

Bit	Name	R / W	Description	Default
15:0	Symbol Error	RO SC	This register contains a 16-bit counter and increments for each symbol error. The counter stop counting additional symbol errors when it is full. This counter is self-clearing on read.	0

7.3.8 100BASE-TX Receive EOF Error Counter: Register 23

Bit	Name	R / W	Description	Default
15:0	Premature End of Frame	RO SC	This register contains a 16-bit counter and increments for each premature end of frame event. It stops counting additional premature end of frame events when it is full. It is self-clearing on read.	0

7.3.9 10BASE-T Receive EOF Error Counter: Register 24

Bit	Name	R / W	Description	Default
15:0	End of Frame	RO SC	This register is a 16-bit counter that increments for each end of frame error event. The counter stops counting additional errors when it is full. It is self-clearing on read.	0

7.3.10 10BASE-T Transmit Jabber Detect Counter: Register 25

Bit	Name	R / W	Description	Default
15:0	Jabber Detect	RO SC	This register is a 16-bit counter that increments for each jabber detection event. The counter stops counting additional events when it is full. It is self-clearing on read.	0

7.3.11 Equalizer Control and Status Register: Register 26

This register is used to control and monitor the operation of the 8255x PHY module equalizer (excluding the 82557 since it does not have an integrated PHY unit). Bits 15:13 specify the command, and bits 12:0 contain the data field for the command.

Bit	Name	R / W	Description	Default
15:0	Equalizer Control and Status	RW	Bits 15:13 contain the opcode command while bits 12:0 hold the command data.	

Opcode Command (bits 15:13)	Command Data (bits 12:0)
000 NOP	
001 Write to ASD configuration register 0	[12] Set zero command. Set value of bits 3:0. [11:10] FSM high threshold transitions: 00 = FM: 2.19 ms; SM: 2.01 ms (0.5 ms - 2.03 ms) 01 = 2.19 ms 10 = FM:2.19; SM:2.03 (0.5 ms - disabled) 11 = Disabled [9:8] FSM low threshold transitions: 00 = FM: 1.83 ms; SM: 1.99 ms (0.5 ms - 1.97 ms) 01 = 1.83 ms 10 = FM: 1.83 ms; SM: 1.97 ms (0.5 ms - disabled) 11 = 1 ms [7] Signal squelch force enable. [6] Squelch signal forcing value. [5] Reserved. [4] Enable/disable zero forcing. [3:0] Coded zero 0 through 15.
010 Write to ASD configuration register 1	[12:11] Reserved. [10:9] TMD100 transition ration bits/LPF ratio: 00 = 0.5 / 6 01 = 0.5 / 5 10 = 0.25 / 6 11 = 0 / 6 [8:7] Signal detect 5-bit counter setting value: 00 = 10h 01 = 18h 10 = 1Ch 11 = 1Fh [6] Set signal detect counter command. [5] Reserved. [4] Disable lock adaptation mechanism. [3:1] Reserved. [0] Force test mode and activate LFSR register.

Opcode Command (bits 15:13)	Command Data (bits 12:0)
011 Write to ASD configuration register 2	[9] Breakdown ASD counters. [8] Selects signal detections or transitions. [7:6] Slow mode adaptation time configure: 00 = 67 ms (default) 01 = 0.5 ms 10 = 16.8 ms 11 = 134 ms [5] Signal detect force enable. [4] Signal detect force value. [3:2] Reserved. [1] Disable signal detect high threshold value. [0] Disable signal detect low threshold value.
100 Read status register	[15:14] Reserved. [13:11] ASD command/address register. [10:9] TMD100 transition ratio bits. [8:7] ASD state machine state. [6] Lock adaptation signal. [5] Signal detect. [4] Squelch signal. [3:0] Coded zero.
101 Read jitter register	Read cycle: [15:0] Reflects the jitter register bits. Write cycle: [1:0] Selects the register lines reflected by read. 00 = Bits 15:0 01 = Bits 21:6 1x = Bits 26:11
110 Read clock register	Read cycle: [15:0] Clock counter value. Write cycle: [0] Selects the window reflected by read. 0 = 15:0 1 = 23:8
111 Reserved	

7.3.12 Special Control Register: Register 27

Bit	Name	R / W	Description	Default
15:0	Special Control Register	RW	Bits 15:3 are reserved, and bits 2:0 are used for the LED switch control.	

Table 60. LED Switch Control

Bits 2:0	**ACTLED# Pin**	**LILED# Pin**
000	Activity	Link
001	Speed	Collision
010	Speed	Link
011	Activity	Collision
100	Off	Off
101	Off	On
110	On	Off
111	On	On

7.4 Auto-Negotiation Functionality

The PHY units of the 8255x devices (excluding the 82557) all support auto-negotiation (N-Way). Auto-negotiation is an automatic configuration scheme designed to manage interoperability in heterogeneous LAN environments. It allows two stations with "N" different modes of communication to establish one common mode of operation. Upon power-up, auto-negotiation establishes a link using one or more devices capable of auto-negotiating. An hub that supports auto-negotiation can detect and automatically configure its ports to take maximum advantage of common modes of operation without user intervention or prior knowledge by either station. The possible common modes of operation are: 100BASE-TX, 100BASE-TX full duplex, 10BASE-T, and 10BASE-T full duplex.

7.4.1 Description

Auto-Negotiation selects the fastest operating mode (in other words, the highest common technology denominator) available to hardware at both ends of the cable. A PHY's capability is encoded by bursts of link pulses called Fast Link Pulses (FLPs). Connection is established by FLP exchange and handshake during initialization time. Once the link is established by this handshake, the native link pulse scheme resumes (for example, 10BASE-T or 100BASE-TX). If only one end of the twisted pair is auto-negotiation capable, then the FLP exchange fails and another scheme called parallel detection is used to determine the link settings. A PHY reset, or management renegotiate command (through the MDI interface), restarts the auto-negotiation process. To enable auto-negotiation, bit 12 of the MDI Control Register must be set. If the PHY cannot perform auto-negotiation, this bit is set to 0.

Since only one technology can be used at a time (after every renegotiate command), a prioritization scheme must be used to ensure that the highest common denominator ability is chosen. Table 61 lists the technology ability field bit assignments. Each bit in this table is set according to the PHY capability. Table 62 lists the priority of each of the technologies.

Table 61. Technology Ability Field Bit Assignments

Bit Setting	Technology
0	10BASE-T
1	10BASE-T Full Duplex
2	100BASE-TX
3	100BASE-TX Full Duplex
4	100BASE-T4
5	Reserved
6	Reserved
7	Reserved

Table 62. Technology Priority

Priority	Technology
1	100BASE-TX Full Duplex
2	100BASE-T4
3	100BASE-TX
4	10BASE-T Full Duplex
5	10BASE-T

To detect the correct technology, the two register fields are ANDed together to obtain the highest common denominator. This value is used to map into a priority resolution table used by the MAC driver to select the appropriate technology. The auto-negotiation process occurs in the following steps:

1. Receive 3 consecutive, matching code words.
2. Set acknowledge bit in transmit code word.
3. Receive 3 consecutive, matching code words with acknowledge bit set.
4. Transmit 6 to 8 more code words with acknowledge bit set.
5. Use priority table to determine operating mode.
6. FLP received from link partner is recorded in MDI register.

7.4.2 Parallel Detection

The key to auto-negotiation's interoperation with installed legacy LANs is the Parallel Detection function. Parallel Detection can be used to determine what the line speed is if the link partner does not support an N-Way (the FLP exchange is not supported) repeater or switch. Parallel Detection works by passing the signals present on the receiver to the 100BASE-TX and 100BASE-T4 link monitor functions. If one link monitor function indicates a valid link, then it connects that technology to the media.

The 82555 PHY and the 8255x (excluding the 82557) PHY modules support N-Way and Parallel Detection (in the event that their link partner does not respond to FLPs).

7.5 Vendor-Specific PHY Programming

The Intel® PRO/100B adapters are designed to support Intel and third-party PHYs using TX and T4 PHYs. The PHYs will be capable of auto-negotiation, but certain vendor specific programming hooks may be required to fully support these PHYs. These issues are addressed in this section.

7.5.1 Intel 82555 TX PHY

The 82555 is a 100BASE-TX PHY that supports all standard MDI registers described in Section 7.2, “MDI Register Set” of this document.

The best method for software to detect the speed and duplex of operation for the 82555, assuming it successfully completed auto-negotiation, is by reading bits 1 and 0 of MDI register 16 (10h).

7.5.2 82558 and 82559 Embedded PHY Unit

The embedded PHY used in the 82558 and 82559 supports all standard MDI registers described in Section 7.2, “MDI Register Set” of this document. In addition, the embedded PHY module also supports some registers and bits specific to the 82555 which are detailed above.

The best method for software to detect the speed and duplex operation of the 82558 or 82559, assuming it successfully completed auto-negotiation, is by reading bits 1 and 0 of MDI register 16 (10h).

7.5.2.1 PHY Stand Alone (PHYSA) Mode

Only the 82558 supports a special mode where its PHY unit can be used with an external controller through an MII-like interface. This mode is not fully MII compliant and should be used with care.

The PHY Stand Alone (PHYSA) mode is enabled if the PHYSA bit (EEPROM word 0Ah, bit D4) is set in the EEPROM. If this bit is set, the 82558 regards the FLD3 pin as a PHYSA input pin. When this pin is asserted (high), the 82558 enters PHYSA mode in which:

- The 82558 MII is enabled in PHY mode (TXD are inputs, RXD are outputs, etc.).
- The 82558 operates through this MII as if it is an 82555 in DTE mode.
- The 82558 CSMA unit is disabled (has no clock).
- The PCI unit is functional but is expected to be non-operational (isolated).

The MII-like interface is intended for use with one on-board controller and may not meet the MII timing specification. No additional PHY device (or MII connector) should be connected to this MII bus.

The CLK signal must be provided for at least 6000 clocks after Reset or Alternate Reset is de-asserted. The Isolate signal is to be de-asserted and the Clock signal active for at least 4 clocks if a Reset signal or Alternate Reset signal is to be propagated into the 82558.

The 82558 is not automatically reset upon entering PHYSA mode. In PHYSA mode the Alternate Reset pin affects only the PCI and CSMA units. It does not propagate into the PHY unit. The PCI and CSMA should be reset by asserting the Alternate Reset pin.

The settings used by the PHY prior to entering PHYSA mode may not be supported by the external controller (for example, full duplex flow control).

In PHY Stand Alone mode, a typical event sequence is as follows:

- RST# or ALTRST# is asserted.
- ISOLATE# should be asserted after at least 4 CLK clocks.
- The PCI bus may be stopped 4 CLKs after ISOLATE# is asserted (excluding CLK).
- RST# or ALTRST# is de-asserted, but CLK is still running.
- The 82558 reads the EEPROM (6000 CLK clocks).
- The 82558 enters PHYSA enable state.
- PHYSA pin is asserted.
- ALTRST# should be asserted after at least 4 PHY clocks.

To exit the PHY Stand Alone mode, the following sequence of events occurs:

- PHYSA pin is de-asserted.
- The PCI bus is activated and CLK is valid.
- ISOLATE# is de-asserted.
- RST# or ALTRST# or a software reset is recommended.

Programming Recommendations 8

8.1 Adapter Initialization

The initialization code can be broadly split in the following code modules:

- 8255x initialization
- PHY detection and initialization
- NOS specific initialization

The sample source code provides specific coding examples. A list of programming issues to consider is listed in the following subsections by code module.

8.1.1 8255x Initialization

- The device is internally fully reset on power up. However, the adapter and the 8255x are not fully reset after a warm reboot. Thus, it is recommended that the software issues a Port Software Reset command before accessing the device.

  Note: All Port commands cause a complete internal reset requiring software to re-initialize the device.

- The interrupt mask bit (SCB Command word, Mask bit) is set to 0 after a complete reset, implying that interrupts are enabled. It is software responsibility to ensure that this bit is set to disable interrupts after a Port command is issued and before polled mode commands are executed.
- If promiscuous mode is enabled, the following Configure command bits are also affected:
  - — SAVE BAD FRAMES. This feature should be enabled.
  - — DISCARD SHORT FRAMES. This feature should be disabled.
  - — PADDING. This feature should be disabled.
  - — STRIPPING. This feature should be disabled.
- It is recommended that all non-transmit action commands are executed in polled mode. Software should maintain a single list of transmit CBs and modify a CB to a non-transmit CB when required. The 8255x supports dynamic CU Resume commands on non-transmit commands; therefore, no checks are required before such a command is issued.

8.1.2 PHY Detection and Initialization

It is recommended that drivers support the Intel® 82503 and 82555 and the National Semiconductor* DP83840 devices as well as the integrated PHY units of the 82558, 82559, 82550, and 82551. In addition, a vendor specific routine should be executed immediately after PHY detection to initialize specific registers in other third party PHYs (for example, the DP83840).

8.1.3 NOS Specific Initialization

Software should be written so that NOS specific interface routines call lower level driver routines. This will enable code re-use.

8.2 Transmit Processing

Frame transmission is the most critical part of driver software, especially in the server environment. The 8255x supports dynamic chaining that allows frames to be chained to the active transmit queue, even as the device processes frames in the queue. A simplified pseudo code flow follows:

```
Get free TxCB    Set the S bit in the TxCB;
                 Set up other fields in the TxCB;
                 Reset S bit in the previous TxCB;
                 Wait for previous command to be accepted by the 8255x (SCB command = 0);
                 CU_RESUME
```

For dynamic chaining:

1. TxCBs should be initialized in a static circularly linked list at initialization time. Subsequently, software should not change these links. The 8255x remembers the next CB link if it suspends. When a new CU Resume is issued, it follows the link to the next CB without accessing host memory to read the next link.
2. The device can be configured to generate interrupts in one of 2 ways:
    a. If the device is configured to generate CNA interrupts, an interrupt is generated when the CU becomes inactive as a result of entering the suspended state. If a CU Resume command is issued, when the CU actively transmits the current last TxCB the 8255x re-reads the current last TxCB S bit. If the S bit is still set, it ignores the CU Resume command. If the bit is reset, it proceeds to execute the next TxCB.

    b. If interrupts are not desired each time the CU suspends, the 8255x can be configured for CI interrupts. In this case, an interrupt is generated only when the CU encounters a TxCB with the I bit set.

8.3 Frame Reception

The 8255x supports early receive interrupts only in the simplified memory model. It should be noted that:

- The 8255x generates an interrupt when a frame is received from the wire. If early receive interrupts are enabled, the first interrupt is generated after HDS bytes are deposited in host memory and the second (final) interrupt is generated when the frame is fully received in memory. Software cannot selectively disable these interrupts.
- An RNR interrupt is generated on the first frame that the device fails to transfer to host memory due to a condition of unavailable resources.

8.4 Interrupt Processing

The 8255x supports latched level triggered interrupts. Interrupts can be shared in the system if the software and NOS support this mechanism. The SCB Command and Status words provide the necessary interface for interrupt management. The Mask bit in the SCB Command word should be set to 1 to disable interrupts at the adapter. Writing this bit to 0 re-enables adapter interrupts. The device supports interrupts from multiple sources as defined by the interrupt status bits in the SCB. If more than one interrupt source is active, the device interrupt will stay asserted until all bits are acknowledged.

Another useful interrupt source is the Software Interrupt (SWI) bit. Setting this bit in the SCB Command word generates an interrupt. Setting the corresponding bit in the SCB Status word de-asserts the interrupt line. Software that need to generate interrupt requests can use this feature. However, if the Mask bit and SWI bit are set at the same time, the Mask bit takes precedence. The SWI bit will cause an interrupt request to wait. As soon as the Mask bit is de-asserted, the interrupt will propagate to the CPU.

Wake-up Functionality A

Note: This appendix applies only to the 82558 and subsequent devices.

Wake-up functionality was first introduced with the 82558 A-step. This component is capable of being brought out of a power managed stated by programming it to wake on reception of a packet addressed to it, a multicast or broadcast packet, any received packet, or a Magic Packet*. These wake-up capabilities are enabled through the Configuration Command bits.

This functionality was carried forward to the 82558 B-step and all subsequent Fast Ethernet devices. However, the wake on reception of a broadcast packet was changed to wake on reception of broadcasted ARP or VLAN ARP. The Configuration Command bits used to enable these wake-up capabilities were also modified from the 82558 A-step. In addition, the 82558 B-step and all subsequent silicon also support the ability to wake on link status change.

If the device is enabled, it notifies the system of a wake-up packet event by asserting the PME# pin. The PME enable bit is located in the PMCSR. It is system responsibility to ensure that this bit is set only when the device is in the D1, D2, or D3 states (as required by the PCI Power Management Specification). After a wake-up event, the PME status bits in the PMCSR and CSR are set regardless of the value of the PME enable bit.

There are two types of wake-up events: reception of a wake-up packet and a link status event. The detection mechanism for wake-up packets (packet filtering) is further categorized by fixed packet filtering implemented by the hardware in the CSMA unit and flexible packet filtering implemented by the loadable microcode.

A.1 Wake-up Capability

The device uses three different mechanisms to wake the system up:

1. Fixed packet wake up. This is implemented in hardware using the clock driven on the X1 pin.
2. Flexible packet filtering. This is implemented in the loadable microcode using the PCI CLK pin as clock. (The discussion of microcode is beyond the scope of this document.)
3. Link status change event.

The device is able to detect fixed packet and link status wake-up events to the system if power is supplied to the device. This may be the main PC power supply or an auxiliary power supply that is active whenever the system is plugged in.

The ability to use loadable microcode for flexible frame filtering requires a valid clock on the PCI CLK pin. This may be the CLK signal from the PCI bus or other valid clock signal connected to the device clock pin while the system is in low power mode.

The following table describes wake-up capabilities in different combinations of power supply and clock:

External Clock Circuit	Auxiliary Power	Fixed Filtering Wake Up[a]	Flexible Filtering Wake Up	Link Status Change Wake Up	PME Support in PMC
No	No	D1, D2, D3h	D1	D1, D2, D3h	01111b
No	Yes	D1, D2, D3h, D3c	D1	D1, D2, D3h, D3c	11111b
Yes	No	D1, D2, D3h	D1,D2,D3h	D1, D2, D3h	01111b
Yes	Yes	D1,D2, D3h, D3c	D1, D2, D3h, D3c	D1,D2, D3h, D3c	11111b

[a] If the PME enable bit is set in the D0 state, the device will assert PME# for every wake-up event. It is BIOS responsibility to disable PME# in D0.

A.2 Low Power Modes

The device wake-up capabilities require internal PHY and CSMA blocks to be fully active. When the controller is set into the D2 or D3 power state and wake up is disabled, the internal PHY and CSMA units are set into low power modes. In this mode, the device loses the link connection. In D1, the PHY and CSMA units are always active.

Note: Placing the device into low power mode (D2 or D3 power state) with the PME enable bit cleared immediately turns off the internal PHY unit. Setting the PME enable bit while the controller is in a low power state will power up the internal PHY and establish link connection. If the wake up on link status change feature is enabled, the 82558 (or later generation controller) will wake up the system. To prevent the above scenario, the PME enable bit should be active before the device is set into low power mode.

A.3 Power Management Context After Reset

The controller has two hardware reset signals that are both active low:

- RST#. This is the PCI bus reset, which is active on power up of the PCI power source.
- ALTRST#. This is the power on reset (POR) and is active on power up of the auxiliary power source. It is connected to the system auxiliary power good signal.

The PCI Power Management Specification requires that the PME status and PME enable bits in the PMCSR will not be cleared by PCI RST# if the function supports wake up from the D3$_{cold}$ state. If the device samples link status (LISTAT) at 0 after a hardware reset, it assumes that an auxiliary power source is used and wake up from D3$_{cold}$ is possible. In this case, assertion of RST# does not clear these bits.

In order to support Wake on LAN (WOL) mode (preboot wake up), the controller should be able to generate a wake up event without any software configuration, including PCI power management configuration. In this scenario, the default configuration of the power management bits should enable wake up after a hardware reset. The auxiliary power support indication is used to set these bits after reset.

A.3.1 Auxiliary Power Support

The LISTAT signal should be 0 after a hardware reset. For WOL mode, the default value after power up reset (ALTRST# is asserted) of the PME enable and status bits are:

PME_Enable = 1 (wake up is enabled)

PME_status = 0 (no wake-up event)

Since the OS/BIOS expect unknown values in these bits after power up, they will clear both bits using PCI configuration command. If WOL mode is enabled, the device will wake up the system when it receives a Magic Packet* after power up without any need for software interaction.

A.3.2 Auxiliary Power Non-support

If the system does not support and auxiliary power source, LISTAT is floating after a hardware reset. In this case wake up from $D3_{cold}$ is not supported. Both the PME enable and status bits should equal 0 after any hardware reset. To enable the WOL functionality, the PME enable bit is bypassed if WOL mode is enabled and auxiliary power indication is not active (same as 82558A).

The table below summarizes the PME enable and status bits default values and properties in different configurations.

Auxiliary Power (indicated by LISTAT)	WOL bit (EEPROM)	PME Enable Default (ALTRST#)	PME Status Default (ALTRST#)	RST# clears bits to 0?	PME Enable bypassed on wake-up event?
No	0	0	0	Yes	No
No	1	0	0	Yes	Yes
Yes	X	1	0	No	No

NOTE: When the WOL bit in the EEPROM is set and auxiliary power is not supported, the PME enable bit is always ignored regardless of the wake-up event type.

A.4 Fixed Packet Filtering

Fixed packet filtering includes the wake up capabilities implemented in the hardware with a pre-defined fixed pattern. It includes Address Match packets, Magic Packets, and ARP packets. The device filters all incoming frames, monitoring for one of the packet types mentioned, regardless of the power state. Each fixed filter can be enabled using the configuration command. When the corresponding filter bit is enabled, the device asserts PME# for frames passing the filter with a correct CRC value. This filtering mechanism is active when power is supplied to the device with or without an active clock on the PCI bus.

Any packet that passes address filtering is transferred to the internal micromachine. The micromachine may store or further analyze the packet if the following conditions are met:

- Loadable microcode for analyzing and storing the packet in the micromachine resources is active.
- The clock signal is active on the PCI CLK pin. This may be the PCI bus clock or any other clock generated by an external clock switching circuitry.

A.4.1 Magic Packet*

The 82558 and later generation controllers (except the 82559ER) are capable of generating a wake-up event upon reception of a Magic Packet. This feature is enabled by setting a bit in the Configuration command.

A.4.2 Address Matching

The controller may be configured to wake up the system on any packet that passes the Individual Address filtering or the multicast address filtering. The wake up enable control is done by the individual address match wake enable bit and the multicast match wake enable bit of the configuration structure.

Note: The 82558 B-step has the following modifications:

- Address match wake up depends on two configuration bits:
  - — Bit 0 in Byte 19 of the configuration structure changed its role from Address Wake Enable to IA Match Wake Enable.
  - — Bit 7 in Byte 9 of the configuration structure is MC Match Wake Enable.
- Regardless of the power management event configuration, packets that pass address filtering are transferred to the micromachine for either storage or further processing if the clock signal is active on the PCI CLK pin.
- The 82558 B-step will not wake up the system on a broadcast address match (except for ARP frames and frames that match the flexible filtering definition).

A.4.3 ARP Packet Filtering

Address Resolution Protocol (ARP) is used for MAC address resolution of a machine. This protocol generally precedes any IP transaction. The controller can wake up the system when an ARP frame is received if configured by the ARP Wake Enable (ARP_Wake_En) configuration bit.

Note: The broadcast disable configuration bit should equal 0 (broadcast enabled).

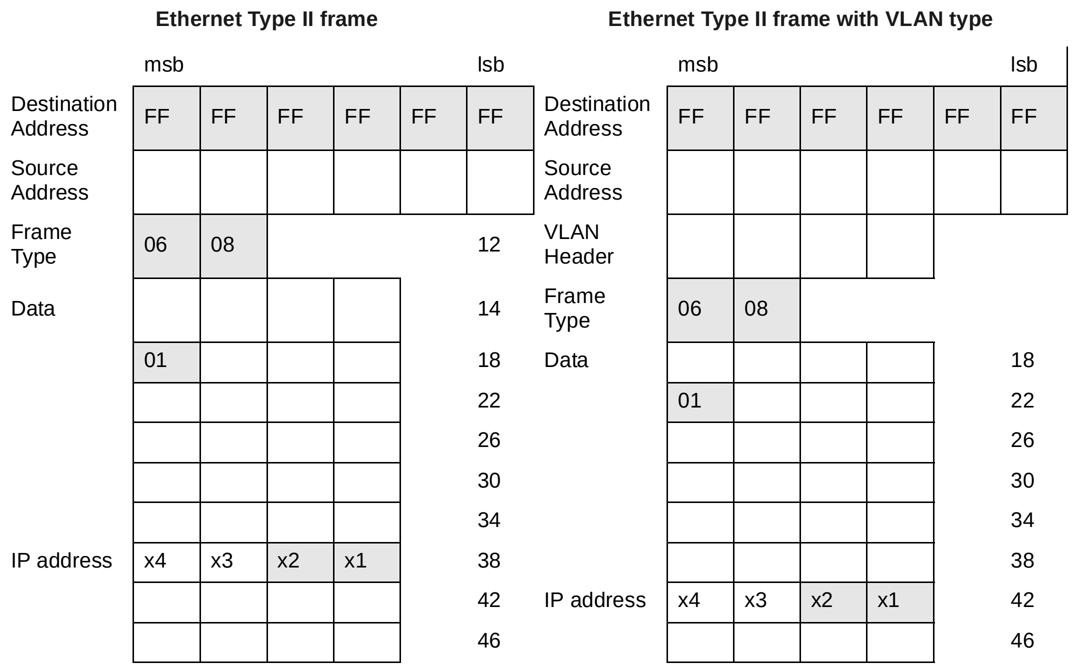
The 82558 and 82559 devices are capable of handling 802.1q VLAN headers if they are configured by the VLAN ARP configuration bit. If this bit is set to 1, the devices can handle ARP frames with or without a 4-byte VLAN header. If it is set to 0 (default), only ARP frames without the VLAN header pass the filter. Multiple IP addresses are not supported by the fixed ARP filter. However, they can be handled through flexible filtering.

The controller needs to be configured with the 16 least significant bits of the IP address (IP_Address configuration word) if ARP filtering is enabled.

The ARP frame format is illustrated below.

Offset	Hexadecimal Pattern	Description
0-5	FF, ... FF	Broadcast destination address

Offset	Hexadecimal Pattern	Description
12	0806	Protocol type (0806 = ARP)
21	01	ARP opcode (01 = request)
38	X1, ... X4	A sequence of 4 bytes, which is the requested IP address



Ethernet Type II frame

	msb					lsb	
Destination Address	FF	FF	FF	FF	FF	FF	
Source Address							
Frame Type	06	08					12
Data							14
	01						18
							22
							26
							30
							34
IP address	x4	x3	x2	x1			38
							42
							46

Ethernet Type II frame with VLAN type

	msb					lsb	
Destination Address	FF	FF	FF	FF	FF	FF	
Source Address							
VLAN Header							
Frame Type	06	08					
Data							18
	01						22
							26
							30
							34
							38
IP address	x4	x3	x2	x1			42
							46

The controller only filters the shaded fields in the frame format above. Only the two low bytes of the IP address are compared. The controller does not check for VLAN type (any frame with type different than 0806 is treated as a VLAN frame if VLAN ARP is enabled).

A.4.4 Configuration Bits for Fixed Wake-up filters

The following configuration bits were added to the 82558 B-step to support fixed wake-up filtering:

Table 63. Fixed Wake-up Configuration Bits

Configuration Bit	Description
VLAN ARP	This bit enables a wake-up event upon reception of ARP frames with a dynamic presence of a VLAN header. This bit takes effect only if the ARP enable bit is also set.
Link Status Change Wake Enable	This bit enables assertion of the PME# signal upon a link status change event. The PME# signal is further gated by the PME enable bit in the PMCSR.
ARP Wake Enable	This bit enables assertion of the PME# signal upon reception of ARP frames. The PME# signal is further gated by the PME enable bit in the PMCSR.

Table 63. Fixed Wake-up Configuration Bits

Configuration Bit	Description
Multicast Match Wake Enable	This bit enables assertion of the PME# signal upon reception of packets that pass through the multicast address filtering. The PME# signal is further gated by the PME enable bit in the PMCSR.
IP Address Low Byte	These bits are used for ARP frame filtering. These are the least significant bits that make the low byte of the IP address, which is located in byte number 40 in a packet without a VLAN header and byte number 44 in a packet with a VLAN header.
IP Address High Byte	These bits are used for ARP frame filtering. They make the high byte of the IP address, which is located in byte number 41 in a packet without a VLAN header and byte number 45 in a packet with a VLAN header.
IA Match Wake Enable	This bit enables the assertion of the PME# signal upon reception of packets that pass through the Individual Address filtering. The PME# signal is further gated by the PME enable bit in the PMCSR.
Magic Packet Wake-up Disable	This bit disables wake up upon reception of a Magic Packet frame.

A.5 Link Status Event

The controller may be configured to wake up the system on link disconnect and connect events. The link status wake-up enable bit was added to the configuration command for the 82558 and later generation devices. If this bit is set, the device generates a power management event when it detects a link disconnect or connect. A link status event is indicated through the Power Management Driver Register (PMDR).

Note: In the 82558 B-step, there is no indication to differentiate this event from any other wake-up event. Frame storage is also not associated with it.

A.6 Flexible Packet Filtering

Programmable packet filtering is targeted to support wake-up packets that are not supported by the hardware filters. Known wake-up packets are listed below.

Wake-up packets supported by the hardware filters are:

- ARP packets at EII packet with single IP address filtering
- Directed packet
- Multicast Address
- Magic Packet
- Link event

Wake-up packets not supported by hardware filters are:

- Directed IP packet
- Net BIOS queries
- 802.2 ARP

- Multiple IP address recognition

A.6.1 Flexible Filtering Terminology

Filter. A filter is a set of a signature and segments generated for a specific frame format. Each filter defines one frame that causes the controller to wake the system. The Intel Fast Ethernet controllers support 3 different filters.

Segment. A segment defines a continuous sequence of bytes in a frame that should be filtered by the controller. Any byte in a frame, which is not included in any segment, is ignored during filtering. A segment is defined by 3 parameters:

- Location of first Dword in the segment
- Location of last Dword in the segment
- Byte masks for first and last Dwords

The Intel controllers support a variable number of segments per filter.

Mask. A mask defines which bytes in each segment should be filtered by the device. Since each segment holds a continuous sequence of bytes, the mask information is required only for the first and last Dwords of the segment.

Signature. A special type of CRC is calculated for each filter. There is one 32-bit signature for all segments of each filter. For each frame, the controller compares the filter signature with the one calculated on all segments and wakes the system if the signatures are equal.

A.6.2 Flexible Filtering Limitations

The flexible filtering feature in the device requires an active clock at the CLK input, which can be the PCI clock while the system bus is at B0 or B1. An active clock should be provided externally while the host bus is in the B2 or B3 states. The controller provides a 25 MHz output clock that can be used to feed the clock during these states. External clock switching circuitry is required to ensure a smooth transition between the PCI and the 25 MHz clock. For more details refer to the device data sheet.

Flexible filtering and frame storing are disabled while the Force TCO pin is asserted. In this case, the device wakes the system when it receives a packet that matches one of the fixed packet filters or when a link status change event occurs. However, no frames will be stored.

Only the first 124 bytes of each frame may be filtered and stored. The frame length is not limited.

The segment limitations are:

- The maximum total number of segments in all filters is 10.
- The first filter may hold 1 to 4 segments.
- The 2nd filter may hold 0 to 6 segments.
- The 3rd filter may hold 0 to 2 segments.

A.6.3 Wake-up Packet Storage

The device uses its internal registers for packet storage during power down mode. Only the first 124 bytes of a frame may be filtered and stored by the device. The residual section of the packet beyond the first 124 bytes is discarded by the device and cannot be used by the software driver after the system is awake.

Note: Any packet passing the fixed packet filtering mechanism (such as ARP) can be stored in the temporary storage area of the micromachine and will be available to the driver when the system is awake. The same limitation for the flexible packet filter storage holds to the hardware filter storage. However, in cases where the frame is inherently lost (for example, in an overrun) or the flexible packet filter is not active (either not loaded or temporarily disabled as in TCO mode), the controller may wake up the system without storing the packet.

A.7 82559 and Later Generation Device Implementation

The 82559 uses different methods for loading programmable wake-up packet filters (both fixed and flexible filters) that are more elegant sophisticated than the mechanisms used by the 82558. The following subsections detail wake-up packet filtering for the 82559.

All of the following subsections refer to the 82559 and 82559ER. However, the 82559ER does not support Magic Packet.

A.7.1 Load Programmable Filter Command Structures

The load programmable filter command of the 82559 is not backward compatible to the 82558.

The 82559 has no explicit limitation on the number of programmable filters, but the length of the command structure is limited to 18 Dwords. The command structure is presented here.

Figure 27. Command Block Structure

<table>
<tr><th>31 16</th><th>15 0</th></tr>
<tr><td>CB Command Word</td><td>CB Status Word</td></tr>
<tr><td colspan="2">Next CB Pointer = FFFF FFFFh</td></tr>
<tr><td colspan="2">Programmable Filters Data (up to 16 Dwords total)</td></tr>
</table>

- **Command Word.** The end of list (EL) and start (S) bits are contained in this word. Either of these must be active (1). The load programmable filters command should be the last command issued to the 82559 by the driver. Therefore, the Next CB Pointer field should be a null pointer (FF,…Fh), even if the S bit is used. The CB opcode is 0008 0000h.
- **Status Word.** The status word contains the complete (C) and OK bits. After completing the load programmable filters command, the 82559 posts a complete OK word equal to A000h.
- **Next CB Pointer.** The Next CB Pointer should be set by the software driver to FF,…Fh (null pointer).
- **Programmable Filters Data Structure.** Programmable filters may have one of two types: flexible filters or pre-defined filters. Any receive packet is filtered first by the MAC address filtering and then by the filters defined below:

— Pre-Defined Filters.

The 82559 contains pre-defined filters for both Ethernet Type II and 802.2 snap. The 82559 distinguishes between these two types according to the MAC Type/Length field. The 82559 also handles VLAN tagging per packet filter as described below. The Magic Packet filter is a unique pre-defined filter, enabled by the configuration command. All other pre-defined filters are controlled by the load programmable filter command.

Filter Type	Reserved (0)	Word Match

Filter Type:

31	30	29	28	27	26	25	24
EL	FIX=1	VLAN	NBH	ARP	IA-Type	IA-Match	TCO

EL The EL bit indicates if this filter is the last active one.

FIX If this bit is set 1, it indicates a pre-defined filter.

VLAN When this bit is active (1), the 82559 skips a 4-byte field of the VLAN type tagging (byte offsets 12 through 15). The VLAN bit affects only the filters defined in the same byte. Other filters that may possibly be affected are the ARP and IA-Type filters.

NBH The neighborhood filter looks for a word match defining a 2-byte field within the MAC destination address to be compared. These two bytes compose a hash value of the node IP address located in offset 02h and 03h of the MAC address. The active fields of the NBH filter are shown below:

Byte Offset	Pattern	Meaning
0	Odd value	LSB of the MAC destination address.
3, 2	Word match	Hash value of the destination IP address.

ARP The ARP filter is used to find a word match with the lower 16 bits of the IP address. Typically, it is ideal for IP filtering. ARP filtering is based on the active fields shown below. The byte offset of the active fields in 802.2 snap type are shifted by 8 bytes ahead. If the filter VLAN flag is set, it is shifted by 4 bytes. For example, the opcode byte is at offset 21 in EII with no VLAN and at offset 33 in 802.2 snap with active VLAN.

EII Byte Offset	Pattern	Meaning
13, 12	06, 08	ARP type = 0806h.
21	01	ARP opcode (01 = request).
41, 40	Word match	Two LSB of the destination IP address.

IA-Type The IA with Type match bit monitors for the destination MAC address of the incoming packet identical to the 6 bytes of the 82559 Individual Address, and the MAC Type/ Length field must be identical to the filter word match fields. The byte offset of the MAC Type/Length field in 802.2 snap type is shifted by 8 bytes ahead and shifted by an additional 4 bytes if the filter VLAN flag is set.

IA-Match The destination MAC address of the incoming packet is identical to the 6 bytes of the 82559 Individual Address.

TCO The 82559 supports a TCO packet filter. The 82559 recognizes TCO packets for power management events at Ethernet Type II with or without VLAN tagging according to the following fields.

Byte Offset	Pattern	Meaning
5, 4, ..., 0	IA Address	MAC destination address.
13, 12	00, 08	IP type = 0800h.
14	45	IPv4, 4 Dword length.
23	11/04	Protocol: either UDP or TCP.
37, 36	02, 6F	TCO Port number 026Fh (0623 decimal).

NOTE: If receive to a TCO controller is enabled, then this bit has no affect on PME.

There are two groups of pre-defined filters:

— The NBH, ARP and IA-Type pre-defined filters each use the word match field.

— The IA-Match and TCO pre-defined filters do not use the word match field.

A single programming word may contain both a pre-defined filter from group 1 and any of the pre-defined filters in group 2. For example, a single programming word may be used to activate: ARP IA-Match and TCO with or without VLAN tagging (depending on the VLAN bit). The examples below describe 82559 wake-up programming for:

— ARP without VLAN, IP address equals 8F.B9.3F.3Dh

— Individual IP with VLAN tagging

— Wake on TCO packet without VLAN (due to unavailable hardware management on the SMB)

— Neighborhood name query (which is described in more detail under the Flexible Filter bullet)

Example 5. 82559 Wake-up Programming

CB Command / Status Word	8008 0000	The EL bit is active.
Next CB PTR	yyyy yyyy	This is the pointer to the next CB structure (no affect if the EL bit is active).
Pre-defined Filter 1	4900 3F3D	ARP and TCO share the same programming filter.
Pre-defined Filter 2	6400 0800	IA-Type with VLAN tagging with IP type (0800h)
Flexible Filter	98xx xxxx zzzz zzzz zzzz zzzz zzzz zzzz	This is the flexible filter for neighborhood name query. The CRC word equals xxxxxxh. The filter mask is composed of 3 Dwords. If the filter resources are a concern, the driver may use a more efficient mask for this filter.

- **Flexible Filter**

Filter type	CRC word
Filter mask (up to 4 Dwords: DW0, DW1, DW2, DW3)	

Filter type:

31	30	29	28	27	26	25	24
EL	FIX=0	MLEN			Reserved (0)		

EL The end of list bit indicates if this filter is the last active one.

FIX Clearing this bit to 0 indicates a flexible filter.

MLEN This field defines the mask length. Each bit in the mask defines a byte in the packet. For example, byte 0 in DW0 starts with byte 0 of the packet, and bit 31 of DW2 defines byte 95 of the packet.

000	Single Dword mask	001	2 Dword mask
011	3 Dword mask	111	4 Dword mask

A.7.2 CRC Word calculation of a Flexible Filter

```
COEFFICIENTS = 0x04C11DB7;                 // CRC Polynomial Coefficients
Signature = 0;                             // Initialize the CRC Signature

for(n=i=0; n<128 & n<FrameLength; ++n)     // Search the frame, using only bytes
{                                          that are part of the flexible filter
    if(Byte n of the Frame is in the       pattern mask.
Flexible Filter)
    {                                      // Shift factor to move the frame byte to
        ShiftBy = (i modulo 3) * 8;        bit position 0, 8 or 16 of the 32-bit
                                           Dword.

                                           // Bit position at 24 of the Dword is not
        if(Signature AND 0x80000000)       used, because the signature is only 24
                                           bits.
        {
            Signature = (Signature << 1)   // If the most significant bit is 1, XOR
XOR                                        in the coefficients, previous signature
                (FrameByte << ShiftBy)     and frame byte shifted over to the
XOR                                        proper position.
                COEFFICIENTS;
        }
        else                               // Otherwise, just XOR the previous
        {                                  signature and frame byte shifted over to
        Signature = (Signature << 1) XOR   the proper position.
            (FrameByte << ShiftBy);
        }                                  // Include the Dword slot for the next
        ++i;                               frame byte that is in the pattern mask.
    }
}

output bits 0-23 of Signature
```

A.7.3 Port Dump Wake Up Packet

The 82559 Port commands are summarized in Table 64, which also includes the new Dump Wake-up Packet command:

Table 64. 82559 Port Commands

Function	Pointer Field 31 4	Opcode Field 3 0
Software Reset	Don't care	0000
Self Test	Self test results pointer (16-byte alignment)	0001
Selective Reset	Don't care	0010
Dump	Dump area pointer (16-byte alignment)	0011
Dump Wake-up	Dump area pointer (16-byte alignment)	0111

Following the Dump Wake-up command, the 82559 writes the stored data of the wake-up packet to the host memory, starting at the address specified in the pointer field. The dump data structure is shown below.

Table 65. Dump Data Structure

Offset	D31		D0
0	Reserved		Status Word (A000h)
1	Reserved		Byte Count
2:N	Wake-up Packet		

The sequence of events after a Dump Wake-up command that the 82559 performs are:

1. Write the byte count field at Dword 1. This field contains the actual number of bytes posted in the host memory. A value of FFh indicates that the Wake-up packet length exceeded the 120 bytes. In this case only the first 120 bytes are posted.
2. Write the wake-up packet data starting at Dword 2.
3. Write a status word composed of the Complete OK bits equal to A000h at Dword 0.

Prior to the Dump Wake-up packet command, the driver should first initialize the status word to 0. After the Dump Wake-up packet command, it should pole the Status word for the Complete bit.

Note: The port dump wake-up packet causes an internal selective reset to the 82559.

Note: The interesting packet bit in the PMDR is set together with the PME status bit when an interesting packet is received.

Note: The 82559 uses the statistic counters resources to store the wake-up packet. The software driver should assume that the statistic counters are infected at power down state. Therefore, it should issue a Dump Reset Statistic Counters command before it resumes nominal operation.

Note: Magic Packets are exceptional to all other wake-up packets. The Magic Packets may cause a power management event and set an indication bit in the PMDR. However, it is not stored by the 82559 for system use when it is awake. The 82559ER does not support Magic Packet wake up.

A.7.4 Power Management Software Flow

The 82559 adheres to the PCI Power Management Specification supporting all three power states D0 through D3. This section describes the required flow of events for software to set the 82559 into the power down states.

A.7.4.1 Power Down without Wake-up Capabilities

The 82559 supports a very low power state if wake-up capabilities are not required. The OS should follow these steps:

1. Clear the PME enable bit in the PMCSR to the PME disable state.
2. Set the 82559 to the D3 power state (by OS).

   Note: Step 2 may be executed together with step 1 (same cycle) but not before. If step 2 is executed before step 1, the 82559 might assert PME# if wake on link status change is enabled. At this state, the 82559 enters the deep power down state, where the PHY is turned off. At this state, the 82559 consume less than 7 mA if the PCI

CLK is inactive and 10 mA if it is active. The deep power down state due to PME disable is enabled in the EEPROM.

A.7.4.2 Power Down with Wake-up Capabilities

The 82559 provides wake-up capabilities at all power states. At the D0 state, the 82559 provides a wake on link status change capability, while wake on interesting packets is operating at the other power states: D1, D2, $D3_{hot}$, $D3_{cold}$.

Note: The 82559 requires a presence of a system power supply to support wake-up capabilities in the $D3_{cold}$ state (bit 4 of the PMC register). The OS should follow the following steps listed:

1. Set the RU into the idle state.
2. Issue a configure command to the 82559 setting the wake-up on Magic Packet (except for the 82559ER as it does not support Magic Packet) and wake on link status change (which is supported on 82559ER) as required. If the 82559 has been configured at initialization time as required for the power down state, then this step is redundant.
3. Load the programmable filter if wake on interesting packet is required. This command must be issued at each entrance to the power down state. It must be the last command issued to the 82559 by the driver before it is set by the OS to the power down state as shown in the next step(s).
4. Wait for the command to be completely executed and the CU is at the idle state.
5. Set the PME enable bit in the PMCSR to the active state (by OS).
6. Set the 82559 to the selected Dx (x > 0) power state (by OS).

Note: Step 6 may be executed together with step 5 (same cycle) but not before. Otherwise, the 82559 will issue a redundant power management event. If the deep power down due to PME disable is enabled and the 82559 is set into the D3 state while PME is not enabled, the PHY unit will be turned off. After the PME enable bit is set to the active state, the PHY is turned on. After link is established, the 82559 issues a power management event due to link status change if wake on link status change is enabled.

A.7.4.3 Wake-up Sequence

To initiate the wake-up sequence, the software (OS and driver) are expected to follow the steps below:

1. OS sets the 82559 to the D0 state.
2. OS is expected to read and clear the PME bit in the PMCSR. If the PME bit is not cleared, the 82559 will hold the PME# signal active.
3. If the device is set to D0 from D3, initialization of the base address registers are required.
4. The driver should read and clear the PME indication bits in the PMDR. The PME indication bits are: link status change bit, Magic Packet bit, and the interesting packet bit. The PME status bit in the PMDR may already be cleared by the OS through the PMCSR.
5. The driver should either issue a selective reset command or dump wake-up packet command, depending whether the interesting packet bit is set in the PMDR. If the interesting packet bit is active, the driver should read the wake-up packet using the port dump wake-up packet command.

6. The device statistic counters are corrupted during power down state. Therefore, the driver should clear the statistic counters by first issuing load dump counters address and then a dump and reset statistic counters.
7. If the 82559 was at the D3 power state, a full initialization sequence is required. Otherwise, the driver should initialize the CU and RU base addresses.
8. The driver should issue an RU start command pointing to a valid RFA.
9. The driver is now ready to issue transmit commands and receive packets.

A.7.4.4 Dummy Wake-up Sequence

After the OS requests to set the 82559 in the power down state, the driver aborts the receive activity. Then, the driver loads the required programmable filters to wake-up the system. At this point, the 82559 is capable of waking up the system if the power management event is enabled.

The driver informs the OS that it may place the 82559 in the power down state. The OS may proceed with the power down sequence, or it might resume nominal operation. If the 82559 driver is requested to resume nominal operation, it should follow the same flow as if the 82559 was at the power down state.

82550 and 82551 Specific Information B

This appendix applies to the Intel® 82550 and 82551 devices.

B.1 IPCB

The IP command block (IPCB) is new and used to activate the new offloading features of the 82550 and 82551. The value of the command field for IPCB is 9h. The relevant aspects of the IPCB for each feature is described in the following subsections. This section summarizes the most useful combinations of the IPCB fields.

Table 66. IPCB Structure

<table>
<tr><th colspan="7">Odd Word (D31:D16)</th><th colspan="5">Even Word (D15:D0)</th><th>Offset</th></tr>
<tr><td>EL</td><td>S</td><td>I</td><td>CID (5 bits)</td><td>000</td><td>NC</td><td>CMD = 1001</td><td>C</td><td>X</td><td>OK</td><td>U</td><td>XXXX XXXX XXXX (12 bits)</td><td>0h</td></tr>
<tr><td colspan="12">Link Address (32 bits)</td><td>4h</td></tr>
<tr><td colspan="12">TBD Array Address (32 bits)</td><td>8h</td></tr>
<tr><td colspan="7">Maximum TCP Payload</td><td rowspan="2">EOF</td><td rowspan="2">0</td><td colspan="3" rowspan="2">IPCB Byte Count (14 bits)</td><td rowspan="2">Ch</td></tr>
<tr><td colspan="4">TBD Number</td><td colspan="3">Transmit Threshold</td></tr>
<tr><td colspan="6">IP Activation (12 bits)</td><td colspan="6">Activation (20 bits)</td><td>10h</td></tr>
<tr><td colspan="4">TCP Header Offset (8 bits)</td><td colspan="3">IP Header Offset (8 bits)</td><td colspan="5">VLAN (16 bits)</td><td>14h</td></tr>
<tr><td colspan="12">Transmit Buffer #0 Address (32 bits)</td><td rowspan="2">18h</td></tr>
<tr><td colspan="12">Reserved (32 bits)</td></tr>
<tr><td colspan="7" rowspan="2">Total TCP Payload (16 bits)</td><td rowspan="2">EL</td><td rowspan="2">0</td><td colspan="3">Transmit Buffer #0 Size (14 bits)</td><td rowspan="2">1Ch</td></tr>
<tr><td colspan="3">Reserved (14 bits)</td></tr>
</table>

NOTE: Source address insertion is not allowed when IPCB is used. Thus, the no source address insertion (NSAI) bit of the configure command must be 1.

Table 67. IP Activation Bits (Byte 13)

15	14	13	12	11	10	9	8
0	0	0	Scheduled Send	Reserved	Reserved	Insert VLAN	Hardware Parse

Table 68. IP Activation Bits (Byte 12)

7	6	5	4	3	2	1	0
Large Send	TCP/UDP Number	TCP/UDP Checksum	IP Checksum	X	X	X	X

The location and definition of the IPCB fields are summarized in the following table.

Table 69. IPCB Fields

Field Name	Byte	Bit(s)	Function	Description
Total TCP/UDP Payload	1Fh: 1Eh	15:0	Parameter, Status	This field specifies the cumulative number of TCP payload bytes requested for transmission as part of this IPCB instance. This field must be valid and will be used when the Hardware Parsing bit is clear. The Total TCP/UDP Payload field is ignored when the Large Send bit is clear.
TCP Header Offset	17h	7:0	Parameter	This field defines the offset in bytes from first frame byte retrieved in memory to the first byte of the TCP or UDP header. This field is ignored when the Hardware Parsing bit is set. When the IP Checksum and TCP/UDP Checksum bits are set and the Hardware Parsing is clear, the TCP Header Offset parameter must be greater than IP Header Offset parameter and IP header length combined, as derived from the data.
IP Header Offset	16h	7:0	Parameter	This field defines the offset in bytes from first frame byte retrieved in memory to the first byte of the IP header. The 4 VLAN bytes are not included in the count only if it is inserted by hardware. This field is ignored when the Hardware Parsing bit is set.
VLAN	15h: 14h	15:0	Parameter	If the Insert VLAN bit is set, the controller inserts the VLAN type (8100h, also known as QTAG protocol type). This 16-bit VLAN ID number and the user priority (32 bits total) are inserted in the transmitted frame.
Reserved	13h	7:5	Reserved	These bits are reserved and should be set to 000.
Scheduled Send	13h	4	Mode	When this bit is set, the controller transmits the associated frame using the scheduling mechanism.
Insert VLAN	13h	1	Mode	If this bit is set, the controller inserts the VLAN type and tag as defined in the VLAN bits description.
Hardware Parsing	13h	0	Mode	When this bit is clear, hardware parsing of the transmitted packet is disabled. If this bit equals 0 (clear), the IP Header Offset, TCP Header Offset and total TCP/UDP payload must be specified to enable checksum operation.
Large Send	12h	7	Mode	When this bit is set, it indicates that the associated packet should be transmitted using the Large Send mechanism. Buffer(s) chained to the IPCB contain data larger than what may be sent in a single Ethernet frame. The 82550 and 82551 break large packets into smaller ones and transmit them using several Ethernet frames.
TCP/UDP Number	12h	6	Parameter	This bit is used to differentiate between TCP and UDP packets. When it is set, it indicates a TCP frame; when clear, a UDP frame. This parameter is relevant and required if the Hardware Parsing bit is clear and the TCP/UDP Checksum is set.
TCP/UDP Checksum	12h	5	Mode	This bit indicates that the TCP/UDP header checksum should be performed in hardware.
IP Checksum	12h	4	Mode	This bit indicates that the IP header checksum should be performed in hardware.

Table 69. IPCB Fields

Scheduling	12h: 10h	19:0	Parameter	If the Scheduled Send bit is set, the transmission of the frame associated with the IPCB is delayed until the internal scheduling counter of the controller reaches the scheduling value. Bits 16:0 are the time stamp and bits 19:17 may be used to specify time window.
Maximum TCP Payload	Fh:Eh	15:0	Parameter	This field specifies the maximum number of bytes to be transmitted in the TCP payload of an Ethernet frame that is part of the Large Send. All frames use this size except the last one. This field is ignored when the Large Send bit is clear.
Transmit Threshold	Eh	7:0	Parameter	In units of 8 bytes, this field defines the number of bytes that should be present in the Transmit FIFO before transmission starts. It is relevant only if the checksum requires the whole frame to be in Transmit FIFO. The value of this parameter should be in the range of 1h to E0h.

Note: Using software parsing is only allowed with legal TCP/IP or UDP/IP packets. When software parsing is used, IP and TCP offsets in the IPCB must point to the appropriate headers and the total TCP/UDP payload should be specified. For all other datagrams, hardware parsing must be used.

Note: VLAN tagging (either in memory or by hardware) and SNAP headers are allowed when software parsing is used as long as the above guideline is met. IP and TCP header offsets are calculated according to memory data structures. For example, if VLAN tag is inserted by hardware, offsets do not change.

The IPCB byte count specifies the number of bytes of transmit data in the immediate data area (from offset 20h and on).

B.1.1 Maximum TCP Payload / TBD Number and Transmit Threshold

When the Large Send mode is selected, this field is interpreted as the Maximum TCP Payload. Otherwise, this 16-bit field is interpreted as the TBD number and Transmit Threshold. The TBD number specifies the total number of TBDs associated with this IPCB. The Transmit Threshold is defined in Table 69 above.

Note: A mode bit puts the 82550 and 82551 into a specific mode. When the mode bit is clear, all parameters that relate to that mode are ignored by the device hardware. Mode bits should obey consistency rules since not all combinations are allowed. The result of a non-supported combination is unpredictable.

Note: It is driver responsibility to guarantee that the total length of a frame is not larger than the allowed MTU size for that connection (which is dependent upon the IP address pair). This is relevant in all modes of operation. For instance, the driver must guarantee proper values for the Maximum TCP Payload in Large Send mode and VLAN length inclusion.

Note: IP fragmentation is not supported by the 82550. Therefore, the driver should not request any offload features for an IP fragment.

B.2 82550 Checksum Operation

The 82550 hardware supports an IP header (including options), UDP and TCP checksum computation. The 82550 computes the checksum using two’s complement addition, and therefore performs end around carry. In other words, it adds the carry generated from bit 15 to the least significant bit in checksum computation.

Note: The 82550 is capable of computing one level of IP header and one TCP/UDP header and payload. In case of multiple IP headers, the driver should compute all except but the outer most IP header checksum.

Note: Per RFC768, if the computed UDP checksum is zero, it is transmitted as all ones. A UDP checksum transmitted as all zeros means that the transmitter generated no UDP checksum.

B.2.1 Driver Interface

Software should compute the partial checksum of the pseudo header and store it in the checksum field of the TCP or UDP header. The partial checksum is the 16-bit one’s complement sum of the 6 words in the pseudo header. It is not the one’s complement of the one’s complement sum. If this is not done by the stack, the driver must be aware of the IP header offset and base offset and the length of the TCP segment to compute and store the pseudo header checksum.

The IPCB structure used for the checksum offload is presented below. The fields relevant to checksum operation are shaded.

Table 70. IPCB Structure Checksum Offload

<table>
<tr><th colspan="7">Odd Word (D31:D16)</th><th colspan="5">Even Word (D15:D0)</th><th>Offset</th></tr>
<tr><td>EL</td><td>S</td><td>I</td><td>CID (5 bits)</td><td>000</td><td>NC</td><td>CMD (4 bits)</td><td>C</td><td>X</td><td>OK</td><td>U</td><td>XXXX XXXX XXXX (12 bits)</td><td>0h</td></tr>
<tr><td colspan="12">Link Address (32 bits)</td><td>4h</td></tr>
<tr><td colspan="12">TBD Array Address (32 bits)</td><td>8h</td></tr>
<tr><td colspan="7">Maximum TCP Payload</td><td rowspan="2">EOF</td><td rowspan="2">0</td><td colspan="3" rowspan="2">IPCB Byte Count (14 bits)</td><td rowspan="2">Ch</td></tr>
<tr><td colspan="4">TBD Number</td><td colspan="3">Transmit Threshold</td></tr>
<tr><td colspan="6">IP Activation (12 bits)</td><td colspan="6">Scheduling (20 bits)</td><td>10h</td></tr>
<tr><td colspan="4">TCP Header Offset (8 bits)</td><td colspan="3">IP Header Offset (8 bits)</td><td colspan="5">VLAN (16 bits)</td><td>14h</td></tr>
<tr><td colspan="12">Transmit Buffer #0 Address (32 bits)</td><td rowspan="2">18h</td></tr>
<tr><td colspan="12">Reserved (32 bits)</td></tr>
<tr><td colspan="7" rowspan="2">Total TCP Payload (16 bits)</td><td rowspan="2">EL</td><td rowspan="2">0</td><td colspan="3">Transmit Buffer #0 Size (14 bits)</td><td rowspan="2">1Ch</td></tr>
<tr><td colspan="3">Reserved (14 bits)</td></tr>
</table>

B.2.2 IPCB Field Assignment

The mode bits in the IP Activation field control the checksum operation of the transmit command.

- IPv4 Checksum (1bit). When this bit is set to 1, the device is forced to perform checksum operation using the IPv4 header and option fields (when present) prior to transmission. This mode can be used with any frame and when the TCP/UDP checksum, Large Send, VLAN, Hardware Parsing, or Scheduling modes are used.

  Note: The stack can command the driver to compute a checksum or not on a frame by frame basis (or specific to the IP or TCP/UDP fields). As a result, the driver may command the hardware not to perform the checksum operation for an IP header. This would be the case when a frame is not a native TCP/UDP/IPv4 frame.

- TCP/UDP Checksum (1bit). When this bit is set to 1, the controller is forced to perform TCP checksum operations (including option fields when present) or UDP checksum operations, based on the TCP/UDP parameter defined prior to transmission. This mode can be used with any frame and when IPv4 checksum, Large Send, VLAN, Hardware Parsing, or Scheduling.

  Note: The stack can command the driver to compute the checksum or not on a frame by frame basis (or specific to IP or TCP/UDP fields). As a result, the driver may command the hardware not to perform the checksum operation for a TCP/UDP frame. This would be the case where the frame is not a native TCP/UDP/IPv4 frame.

- Hardware Parsing (1 bit). When this bit is set to 1, the controller is forced to use its internal transmit parser (autonomous mode is set). The internal transmit parser generates the IP header offset and TCP/UDP header offset for internal consumption by the device blocks. In autonomous mode, the IP header offset, TCP/UDP number and TCP/UDP header offset parameters are ignored. When the Hardware Parsing bit is clear (semi-autonomous mode), the controller uses the IP header offset and TCP/UDP header offset parameters to locate the IPv4 header and TCP/UDP header offsets in the transmit data stream. These are fetched from host memory. Hardware Parsing mode can be used when the IPv4 checksum, TCP/UDP checksum, VLAN, Scheduling, or Large Send bit is set. The Hardware Parsing bit must be cleared when the Security bit is set.

B.2.2.1 Parameters

- TCP/UDP Number (1bit). This parameter is used to denote the frame type for the checksum operation when the TCP/UDP checksum mode is set. When this bit is set, the TCP checksum is performed, and when it is clear, UDP checksum is performed. This parameter is relevant if the Hardware Parsing bit is clear and the TCP/UDP checksum is set. It is also used to select the location of the checksum field.

- IP Header Offset (8 bits). This parameter denotes the offset from the first byte of the destination address (DA) field, which is read by the controller from host memory, to the first byte of the IPv4 header. The 82550 reads this parameter when the Hardware Parsing bit is clear and the IPv4 checksum set.

  Note: VLAN mode will not impact this number since the 4 VLAN bytes are not included in the count since they are inserted by hardware.

- TCP/UDP Header Offset (8 bits). This parameter denotes the offset from the first byte of the destination address (DA) field, which is read by the network device from host memory, to the first byte of the TCP or UDP header. The controller reads this parameter when Hardware Parsing is clear and TCP/UDP checksum is set.

Note: If TCP/UDP headset offset is specified, then the IP header offset must also be specified.

Note: VLAN mode does not impact this number since the 4 VLAN bytes are not included in the count if they are inserted by hardware.

- Transmit Threshold.

B.2.2.2 Status

- If IPv4 checksum mode is set in the IPCB, but the IPv4 header was not found or had a length field smaller than 5, the 82550 may corrupt the frame.
- If TCP/UDP checksum mode is set, but a TCP/UDP header was not found or the length of the TCP/UDP packet was smaller than the offset of the checksum, the 82550 may corrupt the frame.

B.2.2.3 Configuration

IPCB usage is enabled only if the 82550 is in the Extended TxCB mode.

B.2.3 Data Flow

The flow of events for a single transmit frame with checksum offloading is described below.

1. The 82550 enters checksum mode if IPv4 checksum or TCP/UDP checksum is set.
2. If Hardware Parsing is clear, the 82550 uses the IP header offset and TCP/UDP header offset from IPCB. Otherwise, the 82550 parses the frame to locate these offsets.
3. The 82550 computes the checksum over the IP header while it is being access from memory using the IP header length field from the frame (regardless of Hardware Parsing). The checksum is stored in the appropriate location.
4. The 82550 computes the checksum over the TCP/UDP header and payload according to the TCP/UDP header offset, the IP header total length, and the TCP or UDP bit (according to the IP header protocol field if Hardware Parsing is set or according to the TCP/UDP bit if Hardware Parsing is clear). The checksum is stored in the appropriate location.
5. At this point, the frame is ready for transmission.

The TCP/UDP checksum includes the pseudo header partial checksum that should be stored by the driver in the checksum field. In Large Send mode, the driver should store the partial checksum of the pseudo header of the first packet (which includes the header and payload length [or TCP segment size] of the first frame). The controller generates the correct payload length and corresponding pseudo header checksum for each frame transmitted within the Large Send.

Note: The partial checksum required by the 82550 is not the partial checksum passed by the Microsoft* IP stack per Microsoft offloading specification, v0.106.

For TCP/UDP checksum computation, the 82550 requires that the whole frame is copied into its internal FIFO. The result is stored in the proper header fields and only then can the frame be cleared for transmission or further processing. The 82550 pipelines transmission of frame N-1 and the checksum computation of frame N such that back-to-back frames on the transmit queue will flow at maximal DMA rate and checksum operation and transmission are at wire speed. This minimizes the performance hit to “first frames events,” where the 82550 transmit FIFO was empty prior to the transmission command.

B.2.4 Tunneling Support

The 82550 and 82551 provide one level of IPv4 header and TCP/UDP checksum computation. In IP tunnel mode, the driver is expected to compute all IP headers checksum except one that may be handled by the device. As these frame formats are not supported in autonomous mode, the driver is expected to provide the semi-autonomous mode parameters to the 82550.

B.3 Large Send

B.3.1 Rationale

Large Send (also known as TCP Segmentation Offload [TSO]) allows the stack to transfer a frame (larger than the MTU for that media) to the driver. (The maximum frame size for 10/100 Mbps Ethernet is 1518 bytes.) For TCP/IP, a typical frame size may be the transmit window (the default is 8760 bytes, which is approximately 6 full size frames). The 82550 supports Large Send of frames up to 64 Kbytes. Performance increases would result from:

- Stack computes only one header per block.
- Stack does not segment the block to fit the MTU size. This decreases the overhead of multiple virtual address mapping and linked list management.

The 82550 loads the prototype header from host memory only once (and stores it in the device) to reduce PCI overhead.

The Large Send operation is supported for TCP only (not for UDP). Since servers are typically more involved in transmission than client systems, the Large Send feature primarily benefits these server systems. The estimated reduction in CPU utilization is 20%.

B.3.2 Driver interface

The IPCB structure used for Large Send operation is presented below. The fields relevant for the Large Send feature are shaded. The driver must ensure that the TCP payload of a Large Send frame is greater than the maximum TCP payload. In other words, there must be more than one frame in a Large Send.

Table 71. IPCB Structure Large Send

<table>
<tr><th colspan="7">Odd Word (D31:D16)</th><th colspan="5">Even Word (D15:D0)</th><th>Offset</th></tr>
<tr><td>E
L</td><td>S</td><td>I</td><td>CID (5 bits)</td><td>000</td><td>N
C</td><td>CMD = 1001</td><td>C</td><td>X</td><td>O
K</td><td>U</td><td>XXXX XXXX XXXX (12 bits)</td><td>0h</td></tr>
<tr><td colspan="12">Link Address (32 bits)</td><td>4h</td></tr>
<tr><td colspan="12">TBD Array Address (32 bits)</td><td>8h</td></tr>
<tr><td colspan="7">Maximum TCP Payload</td><td>E
O
F</td><td>0</td><td colspan="3">IPCB Byte Count (14 bits)</td><td>Ch</td></tr>
<tr><td colspan="6">IP Activation (12 bits)</td><td colspan="6">Scheduling (20 bits)</td><td>10h</td></tr>
<tr><td colspan="4">TCP Header Offset (8 bits)</td><td colspan="3">IP Header Offset (8 bits)</td><td colspan="5">VLAN (16 bits)</td><td>14h</td></tr>
<tr><td colspan="12">Transmit Buffer #0 Address (32 bits)</td><td rowspan="2">18h</td></tr>
<tr><td colspan="12">Reserved</td></tr>
<tr><td colspan="7">Total TCP Payload (16 bits)</td><td>E
L</td><td>0</td><td colspan="3">Transmit Buffer #0 Size (14 bits)</td><td>1Ch</td></tr>
</table>

Note: To use Large Send, the 82550 should be configured to use dynamic Transmit Buffer Descriptors (TBDs). The driver should ensure transmit buffers associated with the Large Send IPCB contain enough bytes (headers size and total TCP payload). An end of list (EL) bit should be placed in the last TBD.

B.3.3 IPCB Large Send

The mode bits in the IP Activation field, control the Large Send operation of the transmit command. When the Large Send bit is set to 1, the 82550 is forced into Large Send mode for the processing of this IPCB. The data bytes attached are subject to segmentation by the 82550 and multiple frames of MTU size (except for the last frame less than or equal to the MTU size) will be transmitted.

B.3.3.1 Parameters

- Maximum TCP Payload. This parameter specifies the maximum number of bytes that can be transmitted in the TCP payload portion of an Ethernet frame that is part of this Large Send. All frames in this Large Send, except the last one, will have exactly the maximum TCP payload bytes of TCP payload. The last frame TCP payload size will be the remainder (in other words, it will be less than or equal to the maximum TCP payload bytes).

  Note: The driver must guarantee that the maximum TCP payload value will not cause the complete frame (including VLAN) to be larger than the allowed MTU for that connection based on the IP address pair used.

  Note: The IEEE 802.3ac committee plans to extend the 10/100 Mbps Ethernet frame size to account for VLAN fields. This should not impact the TCP/IP MTU.

- Total TCP Payload. This parameter defines the total number of TCP payload bytes (header not included) requested for transmission as part of this Large Send. This parameter should not be used as the sole indication for Large Send activation. For example, 1500 bytes can be a Large Send on an MTU of 576 bytes.

Note: If the IPv4 checksum and TCP/UDP checksum are clear (checksum offload is not requested), frames will be transmitted without computing and replacing the checksum fields content. Therefore, the driver should set both bits so that the generated frames carry a proper checksum.

Note: When the Large Send feature is used, the 82550 should be configured to pad outgoing packets to 64 bytes minimum size.

For Large Send packets, the driver should prepare all header fields so that they can be transmitted in the first Large Send frame. For frame length related fields, the modifications may be required before IPCB is submitted for transmission. Relevant header fields and their required operation value for the prototype header are listed below.

- SNAP length (if SNAP is used). This should be adjusted for first frame transmission.
- IP total length. This should be adjusted for first frame transmission.
- TCP pseudo header partial sum. This should be adjusted according to first frame length.
- TCP push flag. This cannot be adjusted for the first frame. The driver leaves this flag unchanged. If it is set, the 82550 turns it off in all frames except the last one transmitted.
- TCP FIN flag. This cannot be adjusted for the first frame. The driver leaves this flag unchanged. If it is set, the 82550 turns it off in all frames except the last one transmitted.

B.3.4 Data Flow

The 82550 enters Large Send mode when the Large Send bit is set.

B.3.4.1 Headers Parsing

When the 82550 detects a Large Send IPCB, it parses its L2, IP and TCP headers to obtain header offsets and additional headers information. For this purpose, the 82550 performs a PCI read DMA from the beginning of the first active transmit buffer (either the IPCB immediate area or area pointed to by a TBD). The size of that DMA is minimum. All headers (L2, IP and TCP) must be contained in the first active transmit buffer.

The size of the on-chip prototype header memory is 80 bytes. The driver should ensure that the header of Large Send packets fit in this buffer. This limits the size of the options that can be included in the IP and TCP headers.

Note: Large Send IPCB must be used with the Hardware Parsing bit set.

Note: When Large Send is used, the 82550 should be configured to pad outgoing packets to a minimum size of 64 bytes minimum.

B.3.4.2 First Frame Processing

- L2 header remains unchanged.
- IP header is processed:
  - — IP total length is unchanged.
  - — MF bit equals 0. This is expected but not checked by hardware.

- — Fragment offset equals 0. This is expected but not checked by hardware.
  - — IP options are not altered by hardware if they are present.
  - — IP header checksum is calculated by the checksum hardware if the IP checksum bit is set in the IPCB.
- TCP header is processed:
  - — Sequence number is not changed.
  - — URG, RST, and SYN flags are expected to be clear but are not checked by hardware.
  - — ACK flag value will be transmitted as in the prototype header.
  - — Urgent pointer equals 0. This is expected but not checked by hardware.
  - — If the FIN flag equals 1, then it is cleared in the first frame.
  - — If the PSH flag equals 1, then it is cleared in the first frame.
  - — TCP options are not altered by hardware if they are present. TCP options include time stamp.
  - — TCP checksum is calculated by checksum hardware if the TCP checksum bit is set in the IPCB.

B.3.4.3 Intermediate Frames Processing

The number of bytes left for transmission after the first frame is calculated by subtracting the N * Maximum TCP Payload from the Total TCP/UDP Payload. In other words,

(N * MAX TCP PAYLOAD) – (TOTAL TCP/UDP PAYLOAD)

If the next frame is not the last frame, the number of bytes transmitted is updated and the frame processing rules below should be followed.

- L2 header should be left intact as in the first frame.
- IP header is processed:
  - — IP identification is incremented from last value.
  - — IP options are not altered by hardware if they are present.
  - — IP header checksum is calculated by checksum hardware if the IP checksum bit is set in the IPCB.
- TCP header is processed:
  - — Sequence number is updated for outgoing packets. The previous frame size (TCP payload only) is added to the previous frame sequence number (modulo 2^{32}).
  - — All flags are left intact as in the first frame.
  - — Urgent pointer is left intact as in the first frame.
  - — TCP options are not altered by hardware if they are present
  - — Pseudo header checksum remains unchanged.
  - — TCP checksum is calculated by checksum hardware if the TCP checksum bit is set in the IPCB.

When the headers are finished, they are subject to checksum for processing. The rest of the transmission process is similar to transmission of a small send. This process repeats itself until the last frame is about to be transmitted.

B.3.4.4 Handling the Last Frame

The number of bytes left for transmission after the first frame is calculated by subtracting the N * Maximum TCP Payload from the Total TCP/UDP Payload. In other words:

(N * MAX TCP PAYLOAD) – (TOTAL TCP/UDP PAYLOAD)

To obtain the last frame TCP payload size, the above calculation is performed. The last frame TCP payload size is less than or equal to the maximum TCP payload. The last frame is processed using the rules listed below.

- L2 header is processed. If SNAP is used, the SNAP length field is calculated using the IP total length computations for the IP total length.

  SNAP length field = (SNAP length of first packet) – (maximum TCP payload - TCP payload of last packet)

- IP header is processed:
  - IP total length equals the IP total length of the first packet.

    (MAX TCP PAYLOAD) – (TCP payload of last packet)
  - IP identification is incremented from the last value.
  - IP options are not altered by hardware if they are present.
  - IP header checksum is calculated by checksum hardware if the IP checksum bit is set in the IPCB.
- TCP header is processed:
  - Sequence number is updated for outgoing packets. The previous frame size (TCP payload only) is added to the previous frame sequence number (modulo 2^{32}).
  - ACK, URG, RST and SYN flag values are transmitted as in the prototype header.
  - If the FIN flag equals 1 in the prototype header, it should also be set in this last frame.
  - If the PSH flag equals 1 in the prototype header, it should also be set it in this last frame.
  - Urgent pointer is left intact as in the first frame.
  - Pseudo header checksum is fixed with the TCP length as above.
  - TCP options are not altered by hardware if they are present.
  - TCP checksum is calculated by checksum hardware if the TCP checksum bit is set in the IPCB.

The 82550 reports the transmit status in the IPCB upon completion for all data in the Large Send. This enables the whole Large Send buffer to be reclaimed. In the case of a transmission failure (for example, the maximum collision was exceeded), the Large Send session is not terminated and the transmit operation proceeds with the next Large Send packet.

Note: The stack is responsible for maintaining the TCP sequence number and IP identification integrity.

Note: The 82550 is configured to use dynamic TBDs for Large Send since TBDs are read from host memory in pairs and the number of TBDs is not known in advance. When it is configured in this way, it may read an extra TBD (or 8 bytes) after the last valid TBD. The driver is responsible for allocating enough memory for the TBD array. Otherwise, it needs to ensure that an extra 8 byte read access from the PCI will not affect anything.

B.3.4.5 Performance Considerations

For better performance, the maximum TCP payload should be a multiple of 4. It is recommended to use the largest possible value for this parameter.

B.3.5 Features Co-existence

B.3.5.1 Large Send and Checksum

Large Send and checksum are orthogonal.

B.3.5.2 Large Send and Software Parsing

Large Send cannot be used with software parsing.

B.3.5.3 Large Send and Scheduling Assist

All frames of one Large Send instance are subject to the same scheduling policy. Specifically, if the prototype has a time value for transmission, all subsequent frames are considered ready for transmission after that time and may be transmitted back-to-back.

B.4 RCV Checksum Processing

The 82550 and 82551 compute and verify the IP header checksum and the TCP/UDP checksum on frames successfully parsed. It indicates a checksum match or mismatch in the RFD. An incoming UDP frame with a checksum field of 0 is treated as a checksum match. If the 82550 or 82551 cannot decrypt a frame, the checksum can only be verified to the first IP header only (relevant to types IV, V, and VI).

B.4.1 Data Flow

In checksum modes, the device checks incoming frames for format correctness. If the check fails, a receive parser failure occurs and the frame is passed as is. The incoming frame is subject to all MAC layer checks (for example, CRC, minimum length, address filtering). If the value of the word following the DA and SA fields is less than 5DCh (1500), the controller assumes it is an Ethernet v2 frame. If the word value is greater than or equal to 5DCh (1500), the 82550 and 82551 compare in the incoming frame format to SNAP.

In the 82559 compatibility mode, incoming frames are not verified for their format. The checksum is calculated on all non-MAC frame bytes. Supported frame formats are Ethernet v2 and SNAP.

B.4.1.1 Frame Types

- **Ethernet v2**. If the Ethernet v2 type field equals 0800h, the first byte of IP header is expected right after the optional VLAN field.
- **SNAP**. The 82550 and 82551 skip the DSAP and SSAP fields. The Control field is expected to have a value of 03h. If a match occurs, the first byte of IP header is expected right after the optional VLAN field.

B.4.1.2 Verification Types

- **IPv4 header checks**. For IPv4 frames, no fragmentation occurs. The protocol field should equal:
  - 06 for TCP
  - 17 for UDP
  - 04 for IP
  - 50 for ESP
  - 51 for AH

  Other protocols will not have a checksum.
- **TCP header checks**. The packet length should be greater than or equal to the TCP header.
- **UDP header checks**. The packet length should be greater than or equal to the UDP header.
- **Checksum**. The receive checksum machine computes the IP header checksum, the tunneled IP header checksum (if one is present), the TCP/UDP pseudo header checksum, and the TCP/UDP checksum. If a mismatch occurs, the frame is transferred to memory with an error indication according to the configuration.

B.4.2 82559 Compatibility

In 82559 compatibility mode, the 82550 and 82551 compute the checksum of the whole incoming packet excluding the MAC header and CRC. The checksum word is appended at the end of the packet in the receive buffer in host memory. If the CRC is posted to host memory, the checksum word follows the CRC. The byte count field in the receive memory structures includes the checksum word.

B.5 VLAN Tagging

When the 82550 or 82551 receives a frame with a VLAN tag, it sets bit T in the RFD status word (bytes 1h:0h) and copies the tag (the two bytes after the constant 8100h) to the VLAN field in the RFD (bytes 11h:10h). If VLAN stripping is enabled, the device strips the four VLAN bytes, and the received packet is copied to host memory without these four bytes.