PCI/PCI-X Family of Gigabit Ethernet Controllers Software Developer's Manual

82540EP/EM, 82541xx, 82544GC/EI, 82545GM/EM, 82546GB/EB, and 82547xx

317453006EN.PDF
Revision 4.0

Intel Corporation
P.O. Box 5937
Denver, CO 80217-9808

or call in North America 1-800-548-4725, Europe 44-0-1793-431-155, France 44-0-1793-421-777, Germany 44-0-1793-421-333, other Countries 708-296-9333.

Revision History

Date	Version	Comments
Mar 2009	4.0	Updated Section 5 (added Table 5-15 “LED Configuration Defaults).
Oct 2008	3.9	Updated section 8.7.4 “Discard PAUSE Frames and Pass MAC Control Frames”.
June 2008	3.8	Updated EEPROM Word 21h bit descriptions (section 5.6.18).
June 2008	3.7	Updated Sections 13.4.30 and 13.4.31 (added text stating to use the Interrupt Throttling Register (ITR) instead of registers RDTR and RADV for applications requiring an interrupt moderation mechanism).
Jan 2007	3.6	Added a note to sections 13.4.20 and 13.4.21 for the 82547Gi/EI.
Sept 2007	3.5	Updated section 13.4.16.
May 2007	3.4	Updated section 6.4.1. Changed acronym “WCR” to “WUC”.
Dec 2006	3.3	Updated Table 13-87. Changed bit 24 settings to: 0b = Cache line granularity. 1b = Descriptor granularity.
June 2006	3.2	Updated Table 13.47. Changed the default setting of reserved bit 3 from 0b to 1b.
April 2006	3.1	Added bit definitions (bits 9:8) to PHY register PSCON (16d). Updated Figure 3.2 (added Receive Queue artwork). Changed 81541ER-C0 to 82541ER-CO in Table 5-1.
Nov 2005	3.0	Updated Device Control/Status, EEPROM Flash Control & Data, Extended Device Control, and TCTL register bit assignments. Updated PHY register 00d - 03d, 07d, 09d, 17d - 21d, and 23d bit assignments.
July 2005	2.5	**Initial Public Release.**

Note: This page is intentionally left blank.

Contents

Note: This page intentionally left blank.

Introduction 1

1.1 Scope

This document serves as a software developer's manual for **82546GB/EB**, **82545GM/EM**, **82544GC/EI**, **82541(PI/GI/EI)**, **82541ER**, **82547GI/EI**, and **82540EP/EM** Gigabit Ethernet Controllers. Throughout this manual references are made to the PCI/PCI-X Family of Gigabit Ethernet Controllers or Ethernet controllers. Unless specifically noted, these references apply to all the Ethernet controllers listed above.

1.2 Overview

The PCI/PCI-X Family of Gigabit Ethernet Controllers are highly integrated, high-performance Ethernet LAN devices for 1000 Mb/s, 100 Mb/s and 10 Mb/s data rates. They are optimized for LAN on Motherboard (LOM) designs, enterprise networking, and Internet appliances that use the Peripheral Component Interconnect (PCI) and PCI-X bus.

Note: The **82541xx** and **82540EP/EM** do not support the PCI-X bus.

The **82547GI(EI)** connects to the motherboard chipset through a Communications Streaming Architecture (CSA) port. CSA is designed for low memory latency and higher performance than a comparable PCI interface.

The remaining Ethernet controllers provide a 32-/64-bit, 33/66 MHz direct interface to the PCI Local Bus Specification (revision 2.2 or 2.3), as well as the emerging PCI-X extension to the PCI Local Bus (revision 1.0a).

The Ethernet controllers provide an interface to the host processor by using on-chip command and status registers and a shared host memory area, set up mainly during initialization. The controllers provide a highly optimized architecture to deliver high performance and PCI/CSA/PCI-X bus efficiency. By implementing hardware acceleration capabilities, the controllers enable offloading various tasks such as TCP/UDP/IP checksum calculations from the host processor. They also minimize I/O accesses and interrupts required to manage the Ethernet controllers and provide a highly configurable design that can be used effectively in various environments.

The PCI/PCI-X Family of Gigabit Ethernet Controllers handle all IEEE 802.3 receive and transmit MAC functions. They contain fully integrated physical-layer circuitry for 1000 Base-T, 100 Base-TX, and 10 Base-T applications (IEEE 802.3, 802.3u, and 802.3ab) as well as on-chip Serializer/Deserializer (SerDes)[1] functionality that fully complies with IEEE 802.3z PCS.

1. The **82541xx**, **82547GI/EI**, and **82540EP/EM** do not support any SerDes functionality.

For the **82544GC/EI**, when connected to an appropriate SerDes, it can alternatively provide an Ethernet interface for 1000 Base-SX or LX applications (IEEE 802.3z).

Note: The **82546EB/82545EM** is SerDes PICMG 2.16 compliant. The **82546GB/82545GM** is SerDes PICMG 3.1 compliant.

82546GB/EB Ethernet controllers also provide features in an integrated dual-port solution comprised of two distinct MAC/PHY instances. As a result, they appear as multi-function PCI devices containing two identically-functioning Ethernet controllers. See Section 12 for details.

1.3 Ethernet Controller Features

This section describes the features of the PCI/PCI-X Family of Gigabit Ethernet Controllers.

1.3.1 PCI Features

- 32/64-bit 33/66 MHz, PCI Rev 2.3 and PCI-X 1.0a compliant Host interface (**82546GB/82545GM**)
- 32/64-bit 33/66 MHz, PCI Rev 2.2 and PCI-X 1.0a compliant Host interface (**82546EB, 82545EM**, and **82544GC/EI**)
- 32/64-bit 33/66 MHz, PCI Rev 2.3 compliant Host interface (**82541xx**)
- 32/64-bit 33/66 MHz, PCI Rev 2.2 compliant Host interface (**82540EP/EM**)
- 64-bit addressing for systems with more than 4 GB of physical memory
- Efficient PCI bus master operation
- Command usage optimization for advanced PCI commands

1.3.2 CSA Features (82547GI/EI Only)

- Uses dedicated port for client LAN controller directly on an MCH device
- High-speed interface with twice the peak bandwidth of a 32-bit 33 MHz PCI bus
- PCI power management registers recognized by the MCH
- Interface only uses 13 signals

1.3.3 Network Side Features

- Auto-Negotiation and Link Setup
  - Automatic link configuration including speed, duplex and flow control under IEEE 802.3ab for copper media
  - For GMII/MII mode, the driver complies with the IEEE 802.3ab standard requirements for speed, duplex, and flow control Auto-Negotiation capabilities
- Supports half and full duplex operation at 10 Mb/s and 100 Mb/s speeds while working with the internal PHY

- IEEE 802.3x compliant flow control support
  - Enables control of the transmission of Pause packets through software or hardware triggering
  - Provides indications of receive FIFO status
- State-of-the-art internal transceiver (PHY) with DSP architecture implementation
  - Digital adaptive equalization and crosstalk
  - Echo and crosstalk cancellation
  - Automatic MDI/MDI-X crossover at all speeds and compensation for cable length
  - Media Independent Interfaces (MII) IEEE 802.3e for supporting 10/10BASE-T transceivers
- Integrated dual-port solution comprised of two distinct MAC/PHY instances (**82546GB/EB**)
- Provides on-chip IEEE 802.3z PCS SerDes functionality (**82546GB/EB** and **82545GM/EM**)

1.3.4 Host Offloading Features

- Receive and transmit IP and TCP/UDP checksum offloading capabilities
- Transmit TCP Segmentation (operating system support required)
- Packet filtering based on checksum errors
- Support for various address filtering modes:
  - 16 exact matches (unicast, or multicast)
  - 4096-bit hash filter for multicast frames
  - Promiscuous, unicast and promiscuous multicast transfer modes
- IEEE 802.1q VLAN support[1]
  - Ability to add and strip IEEE 802.1q VLAN tags
  - Packet filtering based on VLAN tagging, supporting 4096 tags
- SNMP and RMON statistic counters
- Support for IPv6 including (not applicable to the **82544GC/EI**):
  - IP/TCP and IP/UDP receive checksum offload
  - Wake up filters
  - TCP segmentation

1. Not applicable to the **82541ER**.

1.3.5 Additional Performance Features

- Provides adaptive Inter Frame Spacing (IFS) capability, enabling collision reduction in half duplex networks (**82544GC/EI**)
- Programmable host memory receive buffers (256 B to 16 KB)
- Programmable cache line size from 16 B to 128 B for efficient usage of PCI bandwidth
- Implements a total of 64 KB (40 KB for the **82547GI/EI**) of configurable receive and transmit data FIFOs. Default allocation is 48 KB for the receive data FIFO and 16 KB for the transmit data FIFO
- Descriptor ring management hardware for transmit and receive. Optimized descriptor fetching and write-back mechanisms for efficient system memory and PCI bandwidth usage
- Provides interrupt coalescing to reduce the number of interrupts generated by receive and transmit operations (**82544GC/EI**)
- Supports reception and transmission of packets with length up to 16 KB
- New intelligent interrupt generation features to enhance driver performance (not applicable to the **82544GC/EI**):
    - Packet interrupt coalescing timers (packet timers) and absolute-delay interrupt timers for both transmit and receive operation
    - Short packet detection interrupt for improved response time to TCP acknowledges
    - Transmit Descriptor Ring “Low” signaling
    - Interrupt throttling control to limit maximum interrupt rate and improve CPU utilization

1.3.6 Manageability Features (Not Applicable to the 82544GC/EI or 82541ER)

- Manageability support for ASF 1.0 and AoL 2.0 by way of SMBus 2.0 interface and either:
  - TCO mode SMBus-based management packet transmit / receive support
  - Internal ASF-compliant TCO controller

1.3.7 Additional Ethernet Controller Features

- Implements ACPI[1] register set and power down functionality supporting D0 and D3 states
- Supports Wake on LAN (WoL)[1]
- Provides four wire serial EEPROM interface for loading product configuration information
  - Allows use of either 3.3 V dc or 5 V dc powered EEPROM
- Provides external parallel interface for up to 512 KB of FLASH memory for support of Pre-Boot Execution Environment (PXE)
- Provides seven general purpose user mode pins
- Provides Activity and Link LED indications
- Supports little-endian byte ordering for 32- and 64-bit systems
- Provides loopback capabilities under TBI (**82544GC/EI**)[2] (internal SerDes for the **82546GB/EB** and **82545GM/EM**) and GMII/MII modes of operation
- Provides IEEE JTAG boundary scan support
- Four programmable LED outputs (Not applicable to the **82544GC/EI**).
  - For the **82546GB/EB**, four programmable LED outputs for each port
- Detection and improved power-management with LAN cable unconnected (**82546GB/EB**)

1.3.8 Technology Features

- Implemented in 0.15µ CMOS process (0.13µ for the **82541xx** and **82547GI/EI**)
- Packaged in 364 PBGA.
  - For the **82544EI**, packaged in 416 PBGA.
  - For the **82540EP/EM**, **82541xx**, and **82547GI/EI**, packaged in 196 PBGA.
- Implemented in low power (3.3 V dc or 5 V dc compatible PCI signaling) CMOS process

1. Not applicable to the **82541ER**.
2. Not applicable to the **82541xx**, **82547GI/EI** or **82540EP/EM**.

1.4 Conventions

This document uses notes that call attention to important comments:

Note: Indicates details about the hardware's operations that are not immediately obvious. Read these notes to get information about exceptions, unusual situations, and additional explanations of some PCI/PCI-X Family of Gigabit Ethernet Controller features.

1.4.1 Register and Bit References

This document refers to Ethernet controller register names using all capital letters. To refer to a specific bit in a register the convention REGISTER.BIT is used. For example, CTRL.ASDE refers to the Auto-Speed Detection Enable bit in the Device Control Register (CTRL).

1.4.2 Byte and Bit Designations

This document uses "B" to abbreviate quantities of bytes. For example, a 4 KB represents 4096 bytes. Similarly, "b" is used to represent quantities of bits. For example, 100 Mb/s represents 100 Megabits per second.

1.5 Related Documents

- IEEE Std. 802.3, 2000 Edition. Incorporates various IEEE standards previously published separately.
- PCI Local Bus Specification, Revision 2.2 and 2.3, PCI Local Bus Special Interest Group.

1.6 Memory Alignment Terminology

Some PCI/PCI-X Family of Gigabit Ethernet Controller data structures have special memory alignment requirements. This implies that the starting physical address of a data structure must be aligned as specified in this manual. The following terms are used for this purpose:

- **BYTE** alignment: Implies that the physical addresses can be odd or even. Examples: 0FECBD9A1h, 02345ADC6h.
- **WORD** alignment: Implies that physical addresses must be aligned on even boundaries. For example, the last nibble of the address can only end in 0, 2, 4, 6, 8, Ah, Ch, or Eh (0FECBD9A2h).
- **DWORD** (Double-Word) alignment: Implies that the physical addresses can only be aligned on 4-byte boundaries. For example, the last nibble of the address can only end in 0, 4, 8, or Ch (0FECBD9A8h).
- **QWORD** (Quad-Word) alignment: Implies that the physical addresses can only be aligned on 8-byte boundaries. For example, the last nibble of the address can only end in 0 or 8 (0FECBD9A8h).
- **PARAGRAPH** alignment: Implies that the physical addresses can only be aligned on 16-byte boundaries. For example, the last nibble must be a 0 (02345ADC0h).

Architectural Overview 2

2.1 Introduction

This section provides an overview of the PCI/PCI-X Family of Gigabit Ethernet Controllers. The following sections give detailed information about the Ethernet controller's functionality, register description, and initialization sequence. All major interfaces of the Ethernet controllers are described in detail.

The following principles shaped the design of the PCI/PCI-X Family of Gigabit Ethernet Controllers:

1. Provide an Ethernet interface containing a 10/100/1000 Mb/s PHY that also supports 1000 Base-X implementations.
2. Provide the highest performance solution possible, based on the following:
   — Provide direct access to all memory without using mapping registers
   — Minimize the PCI target accesses required to manage the Ethernet controller
   — Minimize the interrupts required to manage the Ethernet controller
   — Off-load the host processor from simple tasks such as TCP checksum calculations
   — Maximize PCI efficiency and performance
   — Use mixed signal processing to assure physical layer characteristics surpass specifications for UTP copper media
3. Provide a simple software interface for basic operations.
4. Provide a highly configurable design that can be used effectively in different environments.

The PCI/PCI-X Family of Gigabit Ethernet Controllers architecture is a derivative of the 82542 and 82543 designs. They take the MAC functionality and integrated copper PHY from their predecessors and adds SMBus-based manageability and integrated ASF controller functionality to the MAC[1]. In addition, the **82546GB/EB** features this architecture in an integrated dual-port solution comprised of two distinct MAC/PHY instances.

1. Not applicable to the **82544GC/EI** or **82541ER**.

2.2 External Architecture

Figure 2-1 shows the external interfaces to the **82546GB/EB**.

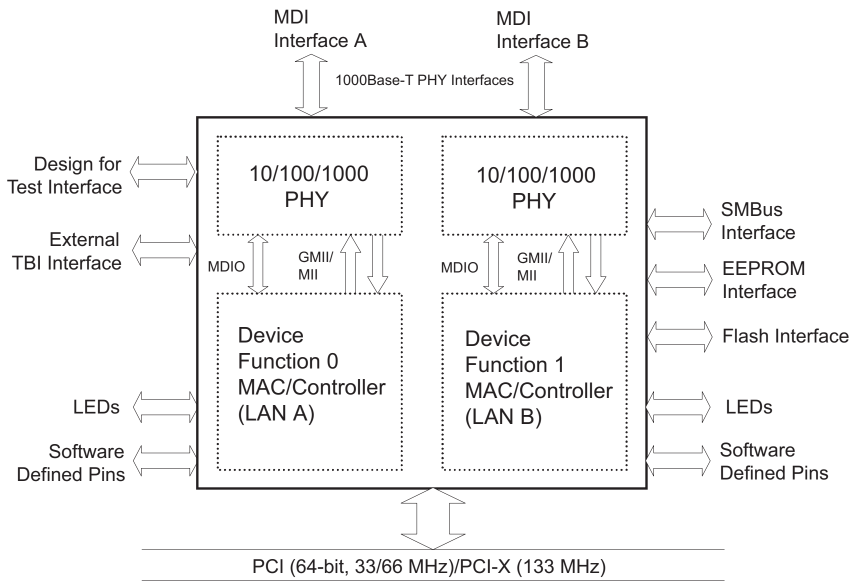
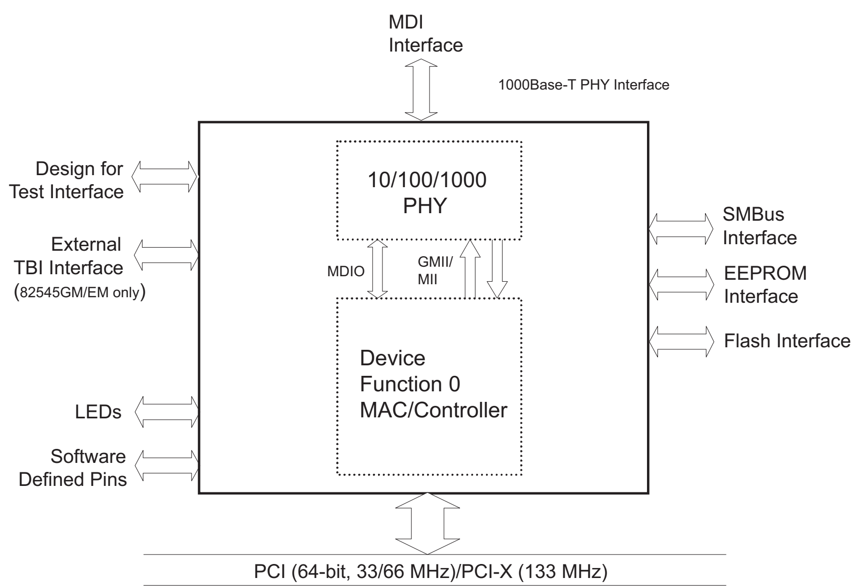


Figure 2-1. 82546GB/EB External Interface

Figure 2-2 shows the external interfaces to the **82545GM/EM**, **82544GC/EI**, **82540EP/EM**, and **82541xx**.



Note: 82540EP/EM and 82541xx do not support PCI-X; 82544GC/EI and 82541ER do not support SMBus interface

Figure 2-2. 82545GM/EM, 82544GC/EI, 82540EP/EM, and 82541xx External Interface

Figure 2-3 shows the external interfaces to the **82547GI/EI**.

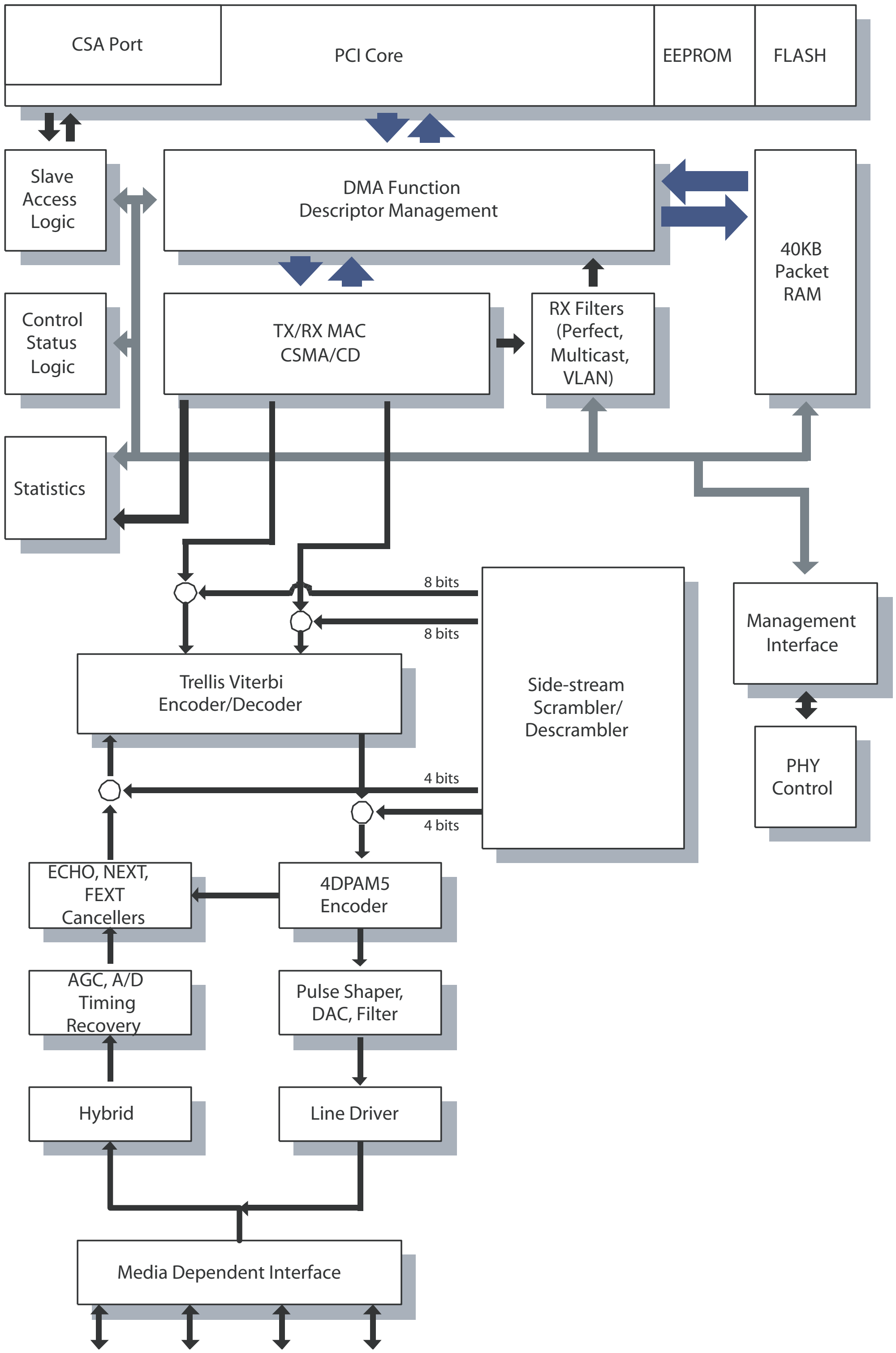


Figure 2-3. 82547GI(EI) External Interface

2.3 Microarchitecture

Compared to its predecessors, the PCI/PCI-X Family of Gigabit Ethernet Controller's MAC adds improved receive-packet filtering to support SMBus-based manageability, as well as the ability to transmit SMBus-based manageability packets. In addition, an ASF-compliant TCO controller is integrated into the controller's MAC for reduced-cost basic ASF manageability.

Note: The **82544GC/EI** and **82541ER** do not support SMBus-based manageability.

For the **82546GB/EB**, this new functionality is packaged in an integrated dual-port combination. The architecture includes two instances of both the MAC and PHY along with a single PCI/PCI-X interface. As a result, each of the logical LAN devices appear as a distinct PCI/PCI-X bus device.

The following sections describe the hardware building blocks. Figure 2-4 shows the internal microarchitecture.

2.3.1 PCI/PCI-X Core Interface

The PCI/PCI-X core provides a complete glueless interface to a 33/66 MHz, 32/64-bit PCI bus or a 33/66/133 MHz, 32/64 bit PCI-X bus. It is compliant with the PCI Bus Specification Rev 2.2 or 2.3 and the PCI-X Specification Rev. 1.0a. The Ethernet controllers provide 32 or 64 bits of addressing and data, and the complete control interface to operate on a 32-bit or 64-bit PCI or PCI-X bus. In systems with a dedicated bus for the Ethernet controller, this provides sufficient bandwidth to support sustained 1000 Mb/s full-duplex transfer rates. Systems with a shared bus (especially the 32-bit wide interface) might not be able to maintain 1000 Mb/s, but can sustain multiple hundreds of Mbps.

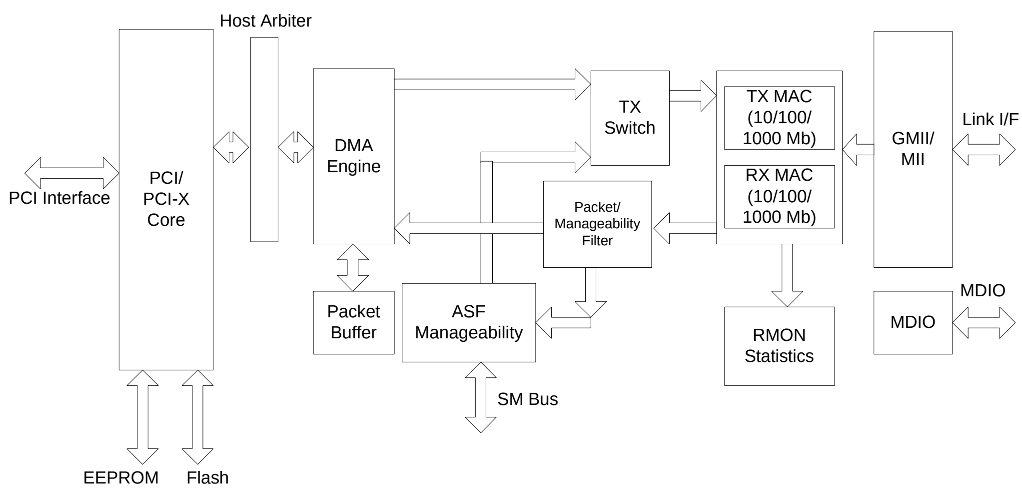


Figure 2-4. Internal Architecture Block Diagram

When the Ethernet controller serves as a PCI target, it follows the PCI configuration specification, which allows all accesses to it to be automatically mapped into free memory and I/O space at initialization of the PCI system.

When processing transmit and receive frames, the Ethernet controller operates as master on the PCI bus. As a master, transaction burst length on the PCI bus is determined by several factors, including the PCI latency timer expiration, the type of bus transfer being made, the size of the data transfer, and whether the data transfer is initiated by receive or transmit logic.

The PCI/PCI-X bus interfaces to the DMA engine.

2.3.2 82547GI/EI CSA Interface

CSA is derived from the Intel® Hub Architecture. The 82547EI Controller CSA port consists of 11 data and control signals, two strobes, a 66 MHz clock, and driver compensation resistor connections. The operating details of these signals and the packet data protocol that accompanies them are proprietary. The CSA port has a theoretical bandwidth of 266 MB/s — approximately twice the peak bandwidth of a 32-bit 33 MHz PCI bus.

The CSA port architecture is invisible to both system software and the operating system, allowing conventional PCI-like configuration.

2.3.3 DMA Engine and Data FIFO

The DMA engine handles the receive and transmit data and descriptor transfers between the host memory and the on-chip memory.

In the receive path, the DMA engine transfers the data stored in the receive data FIFO buffer to the receive buffer in the host memory, specified by the address in the descriptor. It also fetches and writes back updated receive descriptors to host memory.

In the transmit path, the DMA engine transfers data stored in the host memory buffers to the transmit data FIFO buffer. It also fetches and writes back updated transmit descriptors.

The Ethernet controller data FIFO block consists of a 64 KB (40 KB for the **82547GI/EI**) on-chip buffer for receive and transmit operation. The receive and transmit FIFO size can be allocated based on the system requirements. The FIFO provides a temporary buffer storage area for frames as they are received or transmitted by the Ethernet controller.

The DMA engine and the large data FIFOs are optimized to maximize the PCI bus efficiency and reduce processor utilization by:

- Mitigating instantaneous receive bandwidth demands and eliminating transmit underruns by buffering the entire out-going packet prior to transmission
- Queuing transmit frames within the transmit FIFO, allowing back-to-back transmission with the minimum interframe spacing
- Allowing the Ethernet controller to withstand long PCI bus latencies without losing incoming data or corrupting outgoing data
- Allowing the transmit start threshold to be tuned by the transmit FIFO threshold. This adjustment to system performance is based on the available PCI bandwidth, wire speed, and latency considerations

- Offloading the receiving and transmitting IP and TCP/UDP checksums
- Directly retransmitting from the transmit FIFO any transmissions resulting in errors (collision detection, data underrun), thus eliminating the need to re-access this data from host memory

2.3.4 10/100/1000 Mb/s Receive and Transmit MAC Blocks

The controller's CSMA/CD unit handles all the IEEE 802.3 receive and transmit MAC functions while interfacing between the DMA and TBI/internal SerDes/MII/GMII interface block. The CSMA/CD unit supports IEEE 802.3 for 10 Mb/s, IEEE 802.3u for 100 Mb/s and IEEE 802.3z and IEEE 802.3ab for 1000 Mb/s.

The Ethernet controller supports half-duplex 10/100 Mb/s MII or 1000 Mb/s GMII mode and all aspects of the above specifications in full-duplex operation. In half-duplex mode, the Ethernet controller supports operation as specified in IEEE 802.3z specification. In the receive path, the Ethernet controller supports carrier extended packets and packets generated during packet bursting operation. The **82554GC/EI**, in the transmit path, also supports carrier extended packets and can be configured to transmit in packet burst mode.

The Ethernet controller offers various filtering capabilities that provide better performance and lower processor utilization as follows:

- Provides up to 16 addresses for exact match unicast/multicast address filtering.
- Provides multicast address filtering based on 4096 bit vectors. Promiscuous unicast and promiscuous multicast filtering are supported as well.
- The Ethernet controller strips IEEE 802.1q VLAN tag and filter packets based on their VLAN ID. Up to 4096 VLAN tags are supported[1].

In the transmit path, the Ethernet controller supports insertion of VLAN tag information, on a packet-by-packet basis.

The Ethernet controller implements the flow control function as defined in IEEE 802.3x, as well as specific operation of asymmetrical flow control as defined by IEEE 802.3z. The Ethernet controller also provides external pins for controlling the flow control function through external logic.

2.3.5 MII/GMII/TBI/Internal SerDes Interface Block

The Ethernet controller provides the following serial interfaces:

- A GMII/MII interface to the internal PHY.
- Internal SerDes interface[2] (**82546GB/EB** and **82545GM/EM**)/Ten Bit Interface (TBI)[2] for the **82544GC/EI**: The Ethernet controller implements the 802.3z PCS function, the Auto-Negotiation function and 10-bit data path interface (TBI) for both receive and transmit operations. It is used for 1000BASE-SX, -LX, and -CX configurations, operating only at 1000 Mb/s full-duplex. The on-chip PCS circuitry is only used when the link interface is configured for TBI mode and it is bypassed in internal PHY modes.

1. Not applicable to the **82541ER**.
2. Not applicable to the **82544GC/EI**, **82540EP/EM**, **82541xx**, and **82547GI/EI**.

Note: Refer to the Extended Device Control Register (bits 23:22) for mode selection (see Section 13.4.6).

The link can be configured by several methods. Software can force the link setting to Auto-Negotiation by setting either the MAC in TBI mode (internal SerDes for the **82546GB/EB** and **82545GM/EM**), or the PHY in internal PHY mode.

The speed of the link in internal PHY mode can be determined by several methods:

- Auto speed detection based on the receive clock signal generated by the PHY.
- Detection of the PHY link speed indication.
- Software forcing the configuration of link speed.

2.3.6 10/100/1000 Ethernet Transceiver (PHY)

The Ethernet controller provides a full high-performance, integrated transceiver for 10/100/1000 Mb/s data communication. The physical layer (PHY) blocks are 802.3 compliant and capable of operating in half-duplex or full-duplex modes.

Highlights of the PHY blocks are as follows:

- Data stream serializers and encoders. Encoding techniques include Manchester, 4B/5B and 4D/PAM5. These blocks also perform data scrambling for 100/1000 Mb/s transmission as a technique to minimize radiated Electromagnetic Interference (EMI).
- A multi-mode transmit digital to analog converter, which produces filtered waveforms appropriate for the 10BASE-T, 100BASE-TX or 1000BASE-T Ethernet standards.
- Receiver Analog-to-Digital Converter (ADC). The ADC uses a 125 MHz sampling rate.
- Receiver decoders. These blocks perform the inverse operations of serializers, encoders and scramblers.
- Active hybrid and echo canceller blocks. The active hybrid and echo canceller blocks reduce the echo effect of transmitting and receiving simultaneously on the same analog pairs.
- NEXT canceller. This unit removes high frequency Near End Crosstalk induced among adjacent signal pairs.
- Additional wave shaping and slew rate control circuitry to reduce EMI.

Because the Ethernet controller is IEEE-compliant, the PHY blocks communicate with the MAC blocks through an internal GMII/MII bus operating at clock speeds of 2.5 MHz up to 125 MHz.

The Ethernet controller also uses an IEEE-compliant internal Management Data interface to communicate control and status information to the PHY.

2.3.7 EEPROM Interface

The PCI/PCI-X Family of Gigabit Ethernet Controllers provide a four-wire direct interface to a serial EEPROM device such as the 93C46 or compatible for storing product configuration information. Several words of the data stored in the EEPROM are automatically accessed by the Ethernet controller, after reset, to provide pre-boot configuration data to the Ethernet controller before it is accessible by the host software. The remainder of the stored information is accessed by various software modules to report product configuration, serial number and other parameters.

2.3.8 FLASH Memory Interface

The Ethernet controller provides an external parallel interface to a FLASH device. Accesses to the FLASH are controlled by the Ethernet controller and are accessible to software as normal PCI reads or writes to the FLASH memory mapping area. The Ethernet controller supports FLASH devices with up to 512 KB of memory.

Note: The **82540EP/EM** provides an external interface to a serial FLASH or Boot EEPROM device. See Appendix B for more information.

2.4 DMA Addressing

In appropriate systems, all addresses mastered by the Ethernet controller are 64 bits in order to support systems that have larger than 32-bit physical addressing. Providing 64-bit addresses eliminates the need for special segment registers.

Note: The PCI 2.2 or 2.3 Specification requires that any 64-bit address whose upper 32 bits are all 0b appear as a 32-bit address cycle. The Ethernet controller complies with the PCI 2.2 or 2.3 Specification.

PCI is little-endian; however, not all processors in systems using PCI treat memory as little-endian. Network data is fundamentally a byte stream. As a result, it is important that the processor and Ethernet controller agree about the representation of memory data. The default is little-endian mode.

Descriptor accesses are not byte swapped.

The following example illustrates data-byte ordering for little endian. Bytes for a receive packet arrive in the order shown from left to right.

```
01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e
```

Example 2-1. Byte Ordering

There are no alignment restrictions on packet-buffer addresses. The byte address for the major words is shown on the left. The byte numbers and bit numbers for the PCI bus are shown across the top.

Table 2-1. Little Endian Data Ordering

		63							0
		7	6	5	4	3	2	1	0
Byte Address	0	08	07	06	05	04	03	02	01
	8	10	0f	0e	0d	0c	0b	0a	09
	10	18	17	16	15	14	13	12	11
	18	20	1f	1e	1d	1c	1b	1a	19

2.5 Ethernet Addressing

Several registers store Ethernet addresses in the Ethernet controller. Two 32-bit registers make up the address: one is called "high", and the other is called "low". For example, the Receive Address Register is comprised of Receive Address High (RAH) and Receive Address Low (RAL). The least significant bit of the least significant byte of the address stored in the register (for example, bit 0 of RAL) is the multicast bit. The LS byte is the first byte to appear on the wire. This notation applies to all address registers, including the flow control registers.

Figure 2-5 shows the bit/byte addressing order comparison between what is on the wire and the values in the unique receive address registers.

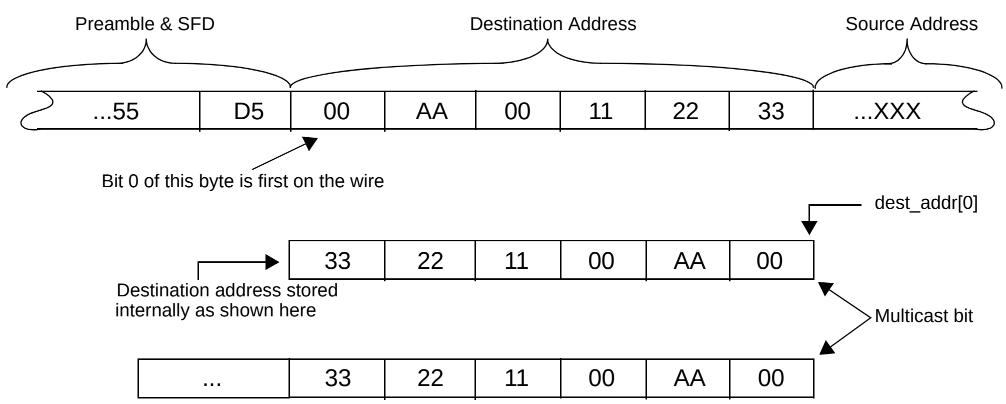


Figure 2-5. Example of Address Byte Ordering

The address byte order numbering shown in Figure 2-5 maps to Table 2-2. Byte #1 is first on the wire.

Table 2-2. Intel® Architecture Byte Ordering

IA Byte #	1 (LSB)	2	3	4	5	6 (MSB)
Byte Value (Hex)	00	AA	00	11	22	33

Note: The notation in this manual follows the convention shown in Table 2-2. For example, the address in Table 2-2 indicates 00_AA_00_11_22_33h, where the first byte (00h_) is the first byte on the wire, with bit 0 of that byte transmitted first.

2.6 Interrupts

The Ethernet controller provides a complete set of interrupts that allow for efficient software management. The interrupt structure is designed to accomplish the following:

- Make accesses "thread-safe" by using 'set' and 'clear-on-read' rather than 'read-modify-write' operations.
- Minimize the number of interrupts needed relative to work accomplished.
- Minimize the processing overhead associated with each interrupt.

Intel accomplished the first goal by an interrupt logic consisting of four interrupt registers. More detail about these registers is given in sections 13.4.17 through 13.4.21.

- Interrupt Cause 'Set' and 'Read' Registers

  The Read register records the cause of the interrupt. All bits set at the time of the read are auto-cleared. The cause bit is set for each bit written as a 1b in the Set register. If there is a race between hardware setting a cause and software clearing an interrupt, the bit remains set. No race condition exists on writing the Set register. A 'set' provides for software posting of an interrupt. A 'read' is auto-cleared to avoid expensive write operations. Most systems have write buffering, which minimizes overhead, but typically requires a read operation to guarantee that the write operation has been flushed from the posted buffers. Without auto-clear, the cost of clearing an interrupt can be as high as two reads and one write.
- Interrupt Mask 'Set' (Read) and 'Clear' Registers

  Interrupts appear on PCI only if the interrupt cause bit is a 1b, and the corresponding interrupt mask bit is a 1b. Software can block assertion of the interrupt wire by clearing the bit in the mask register. The cause bit stores the interrupt event regardless of the state of the mask bit. The Clear and Set operations make this register more "thread-safe" by avoiding a 'read-modify-write' operation on the mask register. The mask bit is set to a 1b for each bit written in the Set register, and cleared for each bit written in the Clear register. Reading the Set register returns the current value.

Intel accomplished the second goal (minimizing interrupts) by three actions:

- Reducing the frequency of all interrupts (see Section 13.4.17). Not applicable to the **82544GC/EI**.
- Accepting multiple receive packets before signaling an interrupt (see Section 3.2.3)
- Eliminating (or at least reducing) the need for interrupts on transmit (see Section 3.2.7)

The third goal is accomplished by having one interrupt register consolidate all interrupt information. This eliminates the need for multiple accesses.

Note that the Ethernet controller also supports Message Signaled Interrupts as defined in the PCI 2.2, 2.3, and PCI-X specifications. See Section 4.1.3.1 for details.

2.7 Hardware Acceleration Capability

The Ethernet controller provides the ability to offload IP, TCP, and UDP checksum for transmit. The functionality provided by these features can significantly reduce processor utilization by shifting the burden of the functions from the driver to the hardware.

The checksum offloading feature is briefly outlined in the following sections. More detail about all of the hardware acceleration capabilities is provided in Section 3.2.9.

2.7.1 Checksum Offloading

The Ethernet controller provides the ability to offload the IP, TCP, and UDP checksum requirements from the software device driver. For common frame types, the hardware automatically calculates, inserts, and checks the appropriate checksum values normally handled by software.

For transmits, every Ethernet packet might have two checksums calculated and inserted by the Ethernet controller. Typically, these would be the IP checksum, and either the TCP or UDP checksum. The software device driver specifies which portions of the packet are included in the checksum calculations, and where the calculated values are inserted via descriptors (refer to Section 3.3.5 for details).

For receives, the hardware recognizes the packet type and performs the checksum calculations and error checking automatically. Checksum and error information is provided to software through the receive descriptors (refer to Section 3.2.9 for details).

2.7.2 TCP Segmentation

The Ethernet controller implements a TCP segmentation capability for transmits that allows the software device driver to offload packet segmentation and encapsulation to the hardware. The software device driver can send the Ethernet controller the entire IP, TCP or UDP message sent down by the Network Operating System (NOS) for transmission. The Ethernet controller segments the packet into legal Ethernet frames and transmit them on the wire. By handling the segmentation tasks, the hardware alleviates the software from handling some of the framing responsibilities. This reduces the overhead on the CPU for the transmission process thus reducing overall CPU utilization. See Section 3.5 for details.

2.8 Buffer and Descriptor Structure

Software allocates the transmit and receive buffers, and also forms the descriptors that contain pointers to, and the status of, those buffers. A conceptual ownership boundary exists between the driver software and the hardware of the buffers and descriptors. The software gives the hardware ownership of a queue of buffers for receives. These receive buffers store data that the software then owns once a valid packet arrives.

For transmits, the software maintains a queue of buffers. The driver software owns a buffer until it is ready to transmit. The software then commits the buffer to the hardware; the hardware then owns the buffer until the data is loaded or transmitted in the transmit FIFO.

Descriptors store the following information about the buffers:

- The physical address
- The length
- Status and command information about the referenced buffer

Descriptors contain an end-of-packet field that indicates the last buffer for a packet. Descriptors also contain packet-specific information indicating the type of packet, and specific operations to perform in the context of transmitting a packet, such as those for VLAN or checksum offload.

Section 3 provides detailed information about descriptor structure and operation in the context of packet transmission and reception.

Receive and Transmit Description 3

3.1 Introduction

This section describes the packet reception, packet transmission, transmit descriptor ring structure, TCP segmentation, and transmit checksum offloading for the PCI/PCI-X Family of Gigabit Ethernet Controllers.

Note: The **82544GC/EI** does not support IPv6.

3.2 Packet Reception

In the general case, packet reception consists of recognizing the presence of a packet on the wire, performing address filtering, storing the packet in the receive data FIFO, transferring the data to a receive buffer in host memory, and updating the state of a receive descriptor.

3.2.1 Packet Address Filtering

Hardware stores incoming packets in host memory subject to the following filter modes. If there is insufficient space in the receive FIFO, hardware drops them and indicates the missed packet in the appropriate statistics registers.

The following filter modes are supported:

- ***Exact Unicast/Multicast*** — The destination address must exactly match one of 16 stored addresses. These addresses can be unicast or multicast.
- ***Promiscuous Unicast*** — Receive all unicasts.
- ***Multicast* —** The upper bits of the incoming packet's destination address index a bit vector that indicates whether to accept the packet; if the bit in the vector is one, accept the packet, otherwise, reject it. The controller provides a 4096 bit vector. Software provides four choices of which bits are used for indexing. These are [47:36], [46:35], [45:34], or [43:32] of the internally stored representation of the destination address.
- ***Promiscuous Multicast* —** Receive all multicast packets.
- ***VLAN*** — Receive all VLAN[1] packets that are for this station and have the appropriate bit set in the VLAN filter table. A detailed discussion and explanation of VLAN packet filtering is contained in Section 9.3.

Normally, only good packets are received. These are defined as those packets with no CRC error, symbol error, sequence error, length error, alignment error, or where carrier extension or receive errors are detected. However, if the store–bad–packet bit is set in the Device Control register (RCTL.SBP), then bad packets that pass the filter function are stored in host memory. Packet errors are indicated by error bits in the receive descriptor (RDESC.ERRORS). It is possible to receive all packets, regardless of whether they are bad, by setting the promiscuous enables (RCTL.UPE/MPE) and the store–bad–packet bit (RCTL.SBP).

1. Not applicable to the **82541ER**.

If manageability is enabled and if RCMCP is enabled then ARP request packets can be directed over the SMBus or processed internally by the ASF controller rather than delivered to host memory (not applicable to the **82544GC/EI** or **82541ER**.

3.2.2 Receive Data Storage

Memory buffers pointed to by descriptors store packet data. Hardware supports seven receive buffer sizes:

- 256 B
- 512 B
- 1024 B
- 2048 B
- 4096 B
- 8192 B
- 16384 B

Buffer size is selected by bit settings in the Receive Control register (RCTL.BSIZE & RCTL.BSEX). See Section 13.4.22 for details.

The Ethernet controller places no alignment restrictions on packet buffer addresses. This is desirable in situations where the receive buffer was allocated by higher layers in the networking software stack, as these higher layers may have no knowledge of a specific Ethernet controller's buffer alignment requirements.

Although alignment is completely unrestricted, it is highly recommended that software allocate receive buffers on at least cache-line boundaries whenever possible.

3.2.3 Receive Descriptor Format

A receive descriptor is a data structure that contains the receive data buffer address and fields for hardware to store packet information. Table 3-1 lists where the shaded areas indicate fields that are modified by hardware upon packet reception.

Table 3-1. Receive Descriptor (RDESC) Layout

	63 48	47 40	39 32	31 16	15 0
0	Buffer Address [63:0]				
8	Special	Errors	Status	Packet Checksum (See Note)	Length

82544GC/EI only

	63 48	47 40	39 32	31 16	15 0
0	Buffer Address [63:0]				
8	Reserved	Errors	Status	Reserved	Length

Note: The checksum indicated here is the unadjusted "16 bit ones complement" of the packet. A software assist may be required to back out appropriate information prior to sending it to upper software layers. The packet checksum is always reported in the first descriptor (even in the case of multi-descriptor packets).

Upon receipt of a packet for Ethernet controllers, hardware stores the packet data into the indicated buffer and writes the length, Packet Checksum, status, errors, and status fields. Length covers the data written to a receive buffer including CRC bytes (if any). Software must read multiple descriptors to determine the complete length for packets that span multiple receive buffers.

For standard 802.3 packets (non-VLAN) the Packet Checksum is by default computed over the entire packet from the first byte of the DA through the last byte of the CRC, including the Ethernet and IP headers. Software may modify the starting offset for the packet checksum calculation by means of the Receive Control Register. This register is described in Section 13.4.22. To verify the TCP checksum using the Packet Checksum, software must adjust the Packet Checksum value to back out the bytes that are not part of the true TCP Checksum.

3.2.3.1 Receive Descriptor Status Field

Status information indicates whether the descriptor has been used and whether the referenced buffer is the last one for the packet. Refer to Table 3-2 for the layout of the status field. Error status information is shown in Table 3-3.

For multi-descriptor packets, packet status is provided in the final descriptor of the packet (EOP set). If EOP is not set for a descriptor, only the Address, Length, and DD bits are valid.

Table 3-2. Receive Status (RDESC.STATUS) Layout

7	6	5	4	3	2	1	0
PIF	IPCS	TCPCS	RSV	VP	IXSM	EOP	DD

Receive Descriptor Status Bits	Description
PIF (bit 7)	Passed in-exact filter Hardware supplies the PIF field to expedite software processing of packets. Software must examine any packet with PIF set to determine whether to accept the packet. If PIF is clear, then the packet is known to be for this station, so software need not look at the packet contents. Packets passing only the Multicast Vector has PIF set.
IPCS (bit 6)	IP Checksum Calculated on Packet When Ignore Checksum Indication is deasserted (IXSM = 0b), IPCS bit indicates whether the hardware performed the IP checksum on the received packet. 0b = Do not perform IP checksum 1b = Perform IP checksum Pass/Fail information regarding the checksum is indicated in the error bit (IPE) of the descriptor receive errors (RDESC.ERRORS) IPv6 packets do not have the IPCS bit set. Reads as 0b.

Receive Descriptor Status Bits	Description
TCPCS (bit 5)	TCP Checksum Calculated on Packet When Ignore Checksum Indication is deasserted (IXSM = 0b), TCPCS bit indicates whether the hardware performed the TCP/UDP checksum on the received packet. 0b = Do not perform TCP/UDP checksum; 1b = Perform TCP/UDP checksum Pass/Fail information regarding the checksum is indicated in the error bit (TCPE) of the descriptor receive errors (RDESC.ERRORS). IPv6 packets may have this bit set if the TCP/UDP packet was recognized. Reads as 0b.
RSV (bit 4)	Reserved Reads as 0b.
VP (bit 3)	Packet is 802.1Q (matched VET) Indicates whether the incoming packet's type matches VET (i.e., if the packet is a VLAN (802.1q) type). It is set if the packet type matches VET and CTRL.VME is set. For a further description of 802.1q VLANs, see Chapter 9. Reads as 0b.
IXSM (bit 2)	Ignore Checksum Indication When IXSM = 1b, the checksum indication results (IPCS, TCPCS bits) should be ignored. When IXSM = 0b the IPCS and TCPCS bits indicate whether the hardware performed the IP or TCP/UDP checksum(s) on the received packet. Pass/Fail information regarding the checksum is indicated in the status bits as described below for IPE and TCPE. Reads as 1b.
EOP (bit 1)	End of Packet EOP indicates whether this is the last descriptor for an incoming packet.
DD (bit 0)	Descriptor Done Indicates whether hardware is done with the descriptor. When set along with EOP, the received packet is complete in main memory.

Note: See Table 3-5 for a description of supported packet types for receive checksum offloading. Unsupported packet types either have the IXSM bit set, or they don't have the TCPCS bit set.

3.2.3.2 Receive Descriptor Errors Field

Most error information appears only when the Store Bad Packets bit (RCTL.SBP) is set and a bad packet is received. Refer to Table 3-3 for a definition of the possible errors and their bit positions.

The error bits are valid only when the EOP and DD bits are set in the descriptor status field (RDESC.STATUS)

Table 3-3. Receive Errors (RDESC.ERRORS) Layout

7	6	5	4	3	2	1	0
RXE	IPE	TCPE	RSV CXE[a]	RSV	SEQ RSV[b]	SE RSV[b]	CE

a. **82544GC/EI** only.
b. **82541xx**, **82547GI/EI**, and **82540EP/EM** only.

Receive Descriptor Error bits	Description
RXE (bit 7)	RX Data Error Indicates that a data error occurred during the packet reception. A data error in TBI[a] mode (**82544GC/EI**)/internal SerDes (**82546GB/EB** and **82545GM/EM**) refers to the reception of a /V/ code (see Section 8.2.1.3). In GMII or MII mode, the assertion of I_RX_ER during data reception indicates a data error. This bit is valid only when the EOP and DD bits are set; it is not set in descriptors unless RCTL.SBP (Store Bad Packets) control bit is set.
IPE (bit 6)	IP Checksum Error When set, indicates that IP checksum error is detected in the received packet. Valid only when the IP checksum is performed on the receive packet as indicated via the IPCS bit in the RDESC.STATUS field. If receive IP checksum offloading is disabled (RXCSUM.IPOFL), the IPE bit is set to 0b. It has no effect on the packet filtering mechanism. Reads as 0b.
TCPE (bit 5)	TCP/UDP Checksum Error When set, indicates that TCP/UDP checksum error is detected in the received packet. Valid only when the TCP/UDP checksum is performed on the receive packet as indicated via TCPCS bit in RDESC.STATUS field. If receive TCP/UDP checksum offloading is disabled (RXCSUM.TUOFL), the TCPE bit is set to 0b. It has no effect on the packet filtering mechanism. Reads as 0b.
CXE RSV (bit 4)	Carrier Extension Error When set, indicates a packet was received in which the carrier extension error was signaled across the GMII interface. A carrier extension error is signaled by the PHY by the encoding of 1Fh on the receive data inputs while I_RX_ER is asserted. Valid only while working in 1000 Mb/s half-duplex mode of operation. This bit is reserved for all Ethernet controllers except the **82544GC/EI**.
RSV (Bit 3)	Reserved Reads as 0b.

Receive Descriptor Error bits	Description
SEQ (bit 2)	Sequence Error When set, indicates a received packet with a bad delimiter sequence (in TBI mode/ internal SerDes). In other 802.3 implementations, this would be classified as a framing error. A valid delimiter sequence consists of: idle →start-of-frame (SOF) → data, →pad (optional) → end-of-frame (EOF) → fill (optional) → idle.
SE (bit 1)	Symbol Error When set, indicates a packet received with bad symbol. Applicable only in TBI mode/ internal SerDes.
CE (bit 0)	CRC Error or Alignment Error CRC errors and alignment errors are both indicated via the CE bit. Software may distinguish between these errors by monitoring the respective statistics registers.

a. Not applicable to the **82540EP/EM**, **82541xx**, or **82547GI/EI**.

3.2.3.3 Receive Descriptor Special Field

Hardware stores additional information in the receive descriptor for 802.1q packets. If the packet type is 802.1q, determined when a packet type field matches the VLAN[1] Ethernet Register (VET) and RCTL.VME = 1b, then the special field records the VLAN information and the four byte VLAN information is stripped from the packet data storage. The Ethernet controller stores the Tag Control Information (TCI) of the 802.1q tag in the Special field. Otherwise, the special field contains 0000h.

Table 3-4. Special Descriptor Field Layout

802.1q Packets

15 13	12	11 0
PRI	CFI	VLAN

All Other Packets

15 8	7 0
00	00

Receive Descriptor Special Field	Description
VLAN	VLAN Identifier 12 bits that records the packet VLAN ID number
CFI	Canonical Form Indicator 1 bit that records the packet's CFI VLAN field
PRI	User Priority 3 bits that records the packet's user priority field.

1. Not applicable to the **82541ER**.

3.2.4 Receive Descriptor Fetching

The descriptor fetching strategy is designed to support large bursts across the PCI bus. This is made possible by using 64 on-chip receive descriptors and an optimized fetching algorithm. The fetching algorithm attempts to make the best use of PCI bandwidth by fetching a cache line (or more) descriptors with each burst. The following paragraphs briefly describe the descriptor fetch algorithm and the software control provided.

When the on-chip buffer is empty, a fetch happens as soon as any descriptors are made available (software writes to the tail pointer). When the on-chip buffer is nearly empty (RXDCTL.PTHRESH), a prefetch is performed whenever enough valid descriptors (RXDCTL.HTHRESH) are available in host memory and no other PCI activity of greater priority is pending (descriptor fetches and write-backs or packet data transfers).

When the number of descriptors in host memory is greater than the available on-chip descriptor storage, the chip may elect to perform a fetch which is not a multiple of cache line size. The hardware performs this non-aligned fetch if doing so results in the next descriptor fetch being aligned on a cache line boundary. This mechanism provides the highest efficiency in cases where fetches fall behind software.

Note: The Ethernet controller **never** fetches descriptors beyond the descriptor TAIL pointer.

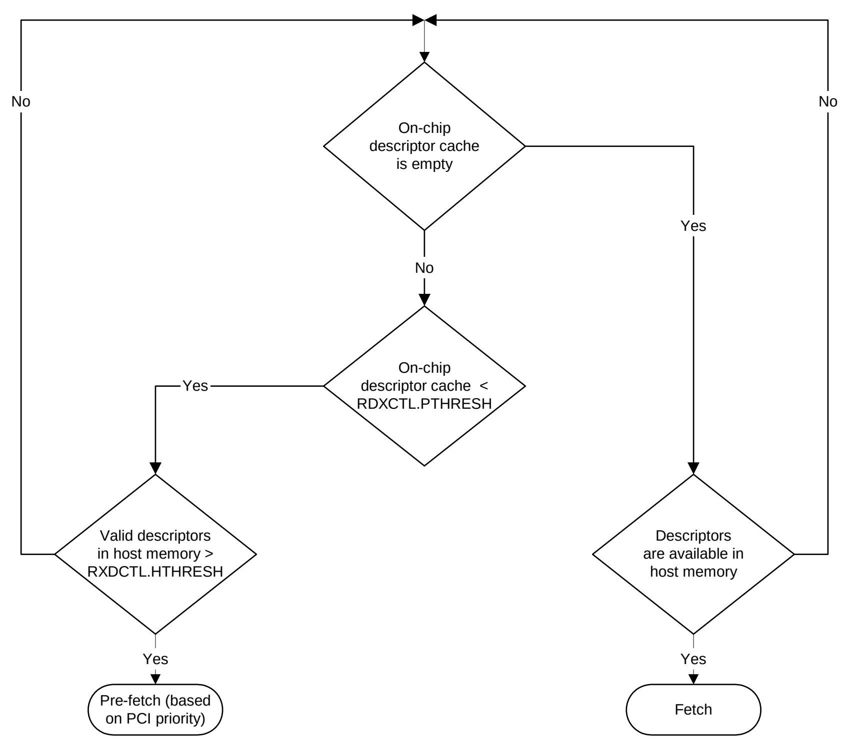


Figure 3-1. Receive Descriptor Fetching Algorithm

3.2.5 Receive Descriptor Write-Back

Processors have cache line sizes that are larger than the receive descriptor size (16 bytes). Consequently, writing back descriptor information for each received packet would cause expensive partial cache line updates. Two mechanisms minimize the occurrence of partial line write backs:

- Receive descriptor packing
- Null descriptor padding

The following sections explain these mechanisms.

3.2.5.1 Receive Descriptor Packing

To maximize memory efficiency, receive descriptors are “packed” together and written as a cache line whenever possible. Descriptors accumulate and are written out in one of three conditions:

- RXDCTL.WTHRESH descriptors have been used (the specified max threshold of unwritten used descriptors has been reached)
- The receive timer expires (RADV or RDTR)
- Explicit software flush (RDTR.FPD)

For the first condition, if the number of descriptors specified by RXDCTL.WTHRESH are used, they are written back, regardless of cacheline alignment. It is therefore recommended that WTHRESH be a multiple of cacheline sizes.

In the second condition, a timer (RDTR or RADV) expiration causes all used descriptors to be written back prior to initiating an interrupt.

In the second condition for the **82544GC/EI**, a timer (RDTR) is included to force timely write–back of descriptors. The first packet after timer initialization starts the timer. Timer expiration flushes any accumulated descriptors and sets an interrupt event (receiver timer interrupt). In general, the arrival rate is sufficiently fast enough that packing is the common case under load.

For the final condition, software may explicitly flush accumulated descriptors by writing the timer register with the high order bit set.

3.2.5.2 Null Descriptor Padding

Hardware stores no data in descriptors with a null data address. Software can make use of this property to cause the first condition under receive descriptor packing to occur early. Hardware writes back null descriptors with the DD bit set in the status byte and all other bits unchanged.

3.2.6 Receive Descriptor Queue Structure

Figure 3-2 shows the structure of the receive descriptor ring. Hardware maintains a circular ring of descriptors and writes back used descriptors just prior to advancing the head pointer. Head and tail pointers wrap back to base when “size” descriptors have been processed.

Software adds receive descriptors by writing the tail pointer with the index of the entry beyond the last valid descriptor. As packets arrive, they are stored in memory and the head pointer is incremented by hardware. When the head pointer is equal to the tail pointer, the ring is empty. Hardware stops storing packets in system memory until software advances the tail pointer, making more receive buffers available.

The receive descriptor head and tail pointers reference 16-byte blocks of memory. Shaded boxes in the figure represent descriptors that have stored incoming packets but have not yet been recognized by software. Software can determine if a receive buffer is valid by reading descriptors in memory rather than by I/O reads. Any descriptor with a non-zero status byte has been processed by the hardware, and is ready to be handled by the software.

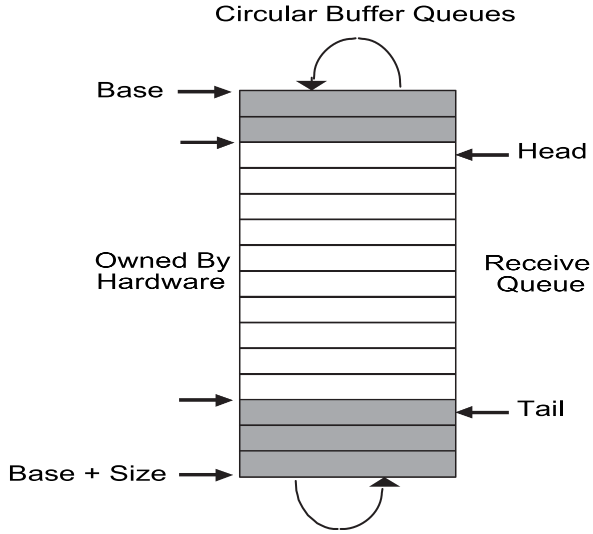


Figure 3-2. Receive Descriptor Ring Structure

Note: The head pointer points to the next descriptor that is written back. At the completion of the descriptor write-back operation, this pointer is incremented by the number of descriptors written back. **HARDWARE OWNS ALL DESCRIPTORS BETWEEN [HEAD AND TAIL]**. Any descriptor not in this range is owned by software.

The receive descriptor ring is described by the following registers:

- Receive Descriptor Base Address registers (RDBAL and RDBAH)

  These registers indicate the start of the descriptor ring buffer. This 64-bit address is aligned on a 16-byte boundary and is stored in two consecutive 32-bit registers. RDBAL contains the lower 32-bits; RDBAH contains the upper 32 bits. Hardware ignores the lower 4 bits in RDBAL.

- Receive Descriptor Length register (RDLEN)

  This register determines the number of bytes allocated to the circular buffer. This value must be a multiple of 128 (the maximum cache line size). Since each descriptor is 16 bytes in length, the total number of receive descriptors is always a multiple of 8.

- Receive Descriptor Head register (RDH)

  This register holds a value that is an offset from the base, and indicates the in–progress descriptor. There can be up to 64K descriptors in the circular buffer. Hardware maintains a shadow copy that includes those descriptors completed but not yet stored in memory.

- Receive Descriptor Tail register (RDT)

  This register holds a value that is an offset from the base, and identifies the location beyond the last descriptor hardware can process. Note that tail should still point to an area in the descriptor ring (somewhere between RDBA and RDBA + RDLEN). This is because tail points to the location where software writes the first new descriptor.

If software statically allocates buffers, and uses memory read to check for completed descriptors, it simply has to zero the status byte in the descriptor to make it ready for reuse by hardware. This is not a hardware requirement (moving the hardware tail pointer is), but is necessary for performing an in–memory scan.

3.2.7 Receive Interrupts

The Ethernet controller can generate four receive-related interrupts:

- Receiver Timer Interrupt (ICR.RXT0)
- Small Receive Packet Detect (ICR.SRPD)
- Receive Descriptor Minimum Threshold (ICR.RXDMT0)
- Receiver FIFO Overrun (ICR.RX0)

3.2.7.1 Receive Timer Interrupt

The Receive Timer Interrupt is used to signal most packet reception events (the Small Receive Packet Detect interrupt is also used in some cases as described later in this section). In order to minimize the interrupts per work accomplished, the Ethernet controller provides two timers to control how often interrupts are generated.

3.2.7.1.1 Receive Interrupt Delay Timer / Packet Timer (RDTR)

The Packet Timer minimizes the number of interrupts generated when many packets are received in a short period of time. The packet timer is started once a packet is received and transferred to host memory (specifically, after the last packet data byte is written to memory) and is reinitialized (to the value defined in RDTR) and started EACH TIME a new packet is received and transferred to the host memory. When the Packet Timer expires (e.g. no new packets have been received and transferred to host memory for the amount of time defined in RDTR) the Receive Timer Interrupt is generated.

Setting the Packet Timer to 0b disables both the Packet Timer and the Absolute Timer (described below) and causes the Receive Timer Interrupt to be generated whenever a new packet has been stored in memory.

Writing to RDTR with its high order bit (FPD) set forces an explicit writeback of consumed descriptors (potentially a partial cache lines amount of descriptors), causes an immediate expiration of the Packet Timer and generates a Receive Timer Interrupt.

The Packet Timer is reinitialized (but not started) when the Receive Timer Interrupt is generated due to an Absolute timer expiration or Small Receive Packet Detect Interrupt.

See section Section 13.4.30 for more details on the Packet Timer.

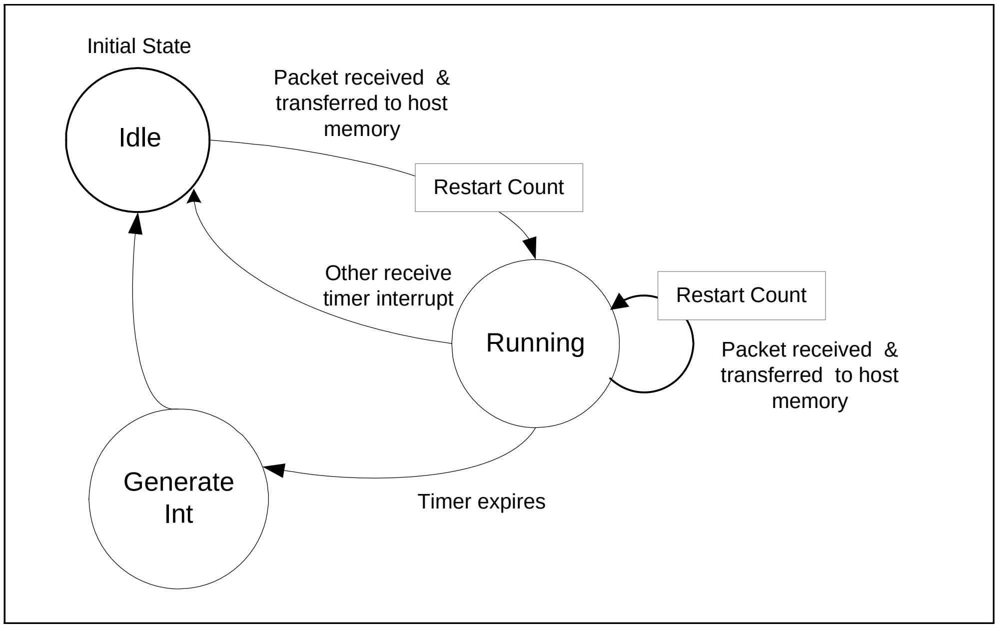


Figure 3-3. Packet Delay Timer Operation (State Diagram)

3.2.7.1.2 Receive Interrupt Absolute Delay Timer (RADV)

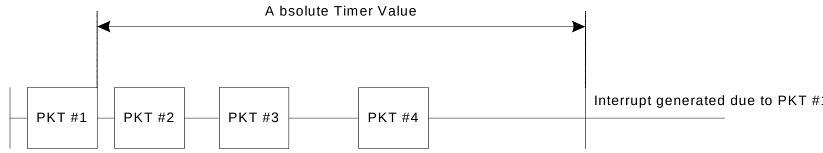
The Absolute Timer ensures that a receive interrupt is generated at some predefined interval after the first packet is received. The absolute timer is started once a packet is received and transferred to host memory (specifically, after the last packet data byte is written to memory) but is NOT reinitialized / restarted each time a new packet is received. When the Absolute Timer expires (no receive interrupt has been generated for the amount of time defined in RADV) the Receive Timer Interrupt is generated.

Setting RADV to 0b or RDTR to 0b disables the Absolute Timer. To disable the Packet Timer only, RDTR should be set to RADV + 1b.

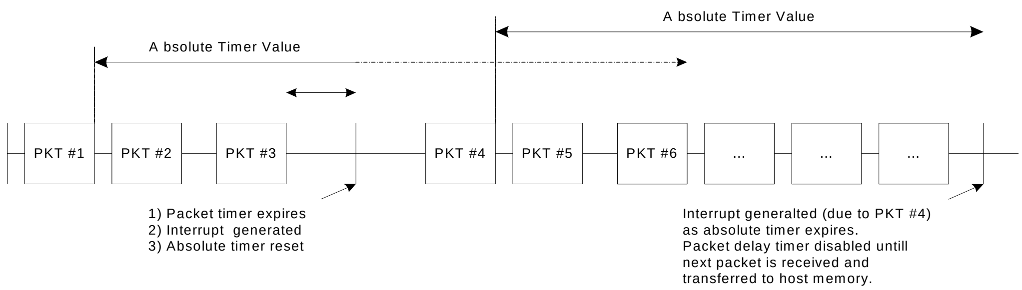
The Absolute Timer is reinitialized (but not started) when the Receive Timer Interrupt is generated due to a Packet Timer expiration or Small Receive Packet Detect Interrupt.

The diagrams below show how the Packet Timer and Absolute Timer can be used together:

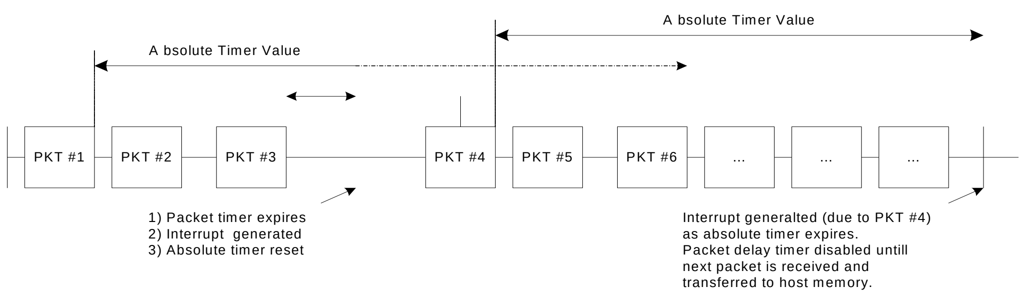
Case A: Using only an absolute timer



Case B: Using an absolute time in conjunction with the Packet timer



Case C: Packet timer expiring while a packet is transferred to host memory.
Illustrates that packet timer is re-started only after a packet is transferred to host memory.



3.2.7.2 Small Receive Packet Detect

A Small Receive Packet Detect interrupt (ICR.SRPD) is asserted when small-packet detection is enabled (RSRPD is set with a non-zero value) and a packet of (size ≤ RSRPD.SIZE) has been transferred into the host memory. When comparing the size the headers and CRC are included (if CRC stripping is not enabled). CRC and VLAN headers are not included if they have been stripped. A receive timer interrupt cause (ICR.RXT0) is also noted when the Small Packet Detect interrupt occurs.

For the **82541xx** and **82547GI/EI**, receiving a small packet does not clear the absolute or packet delay timers, so one packet might generate two interrupts, one due to small packet reception and one due to timer expiration.

3.2.7.3 Receive Descriptor Minimum Threshold (ICR.RXDMT)

The minimum descriptor threshold helps avoid descriptor under-run by generating an interrupt when the number of free descriptors becomes equal to the minimum amount defined in RCTL.RDMTS (measured as a fraction of the receive descriptor ring size).

3.2.7.4 Receiver FIFO Overrun

FIFO overrun occurs when hardware attempts to write a byte to a full FIFO. An overrun could indicate that software has not updated the tail pointer to provide enough descriptors/buffers, or that the PCI bus is too slow draining the receive FIFO. Incoming packets that overrun the FIFO are dropped and do not affect future packet reception.

3.2.8 82544GC/EI Receive Interrupts

The presence of new packets is indicated by the following:

- Absolute timer (RDTR) — A predetermined amount of time has elapsed since the first packet received after the hardware timer was written (specifically, after the last packet data byte was written to memory); this also flushes any accumulated descriptors to memory. Software can set the timer value to 0b if it wants to be notified each time a new packet has been stored in memory.

  Writing the absolute timer with its high order bit 1 forces an explicit flush of any partial cache lines. Hardware writes all used descriptors to memory and updates the globally visible value of the head pointer.

In addition, hardware provides the following interrupts:

- Receive Descriptor Minimum Threshold (ICR.RXDMT)

  The minimum descriptor threshold helps avoid descriptor underrun by generating an interrupt when the number of free descriptors becomes equal to the minimum. It is measured as a fraction of the receive descriptor ring size.

- Receiver FIFO Overrun (ICR.RXO)

  FIFO overrun occurs when hardware attempts to write a byte to a full FIFO. An overrun could indicate that software has not updated the tail pointer to provide enough descriptors/buffers, or that the PCI bus is too slow draining the receive FIFO. Incoming packets that overrun the FIFO are dropped and do not affect future packet reception.

3.2.9 Receive Packet Checksum Offloading

The Ethernet controller supports the offloading of three receive checksum calculations: the Packet Checksum, the IP Header Checksum, and the TCP/UDP Checksum.

Note: IPv6 packets do not have IP checksums.

The Packet checksum is the one's complement over the receive packet, starting from the byte indicated by RXCSUM.PCSS (0b corresponds to the first byte of the packet), after stripping. For example, for an Ethernet II frame encapsulated as an 802.3ac VLAN packet and with RXCSUM.PCSS set to 14 decimal, the Packet Checksum would include the entire encapsulated frame, excluding the 14-byte Ethernet header (DA,SA,Type/Length) and the 4-byte q-tag. The Packet checksum does not include the Ethernet CRC if the RCTL.SECRC bit is set.

Software must make the required offsetting computation (to back out the bytes that should not have been included and to include the pseudo-header) prior to comparing the Packet Checksum against the TCP checksum stored in the packet.

For supported packet/frame types, the entire checksum calculation may be offloaded to the Ethernet controller. If RXCSUM.IPOFLD is set to 1b, the controller calculates the IP checksum and indicates a pass/fail condition to software by means of the IP Checksum Error bit (RDESC.IPE) in the ERROR field of the receive descriptor. Similarly, if the RXCSUM.TUOFLD is set to 1b, the Ethernet controller calculates the TCP or UDP checksum and indicates a pass/fail condition to software by means of the TCP/UDP Checksum Error bit (RDESC.TCPE). These error bits are valid when the respective status bits indicate the checksum was calculated for the packet (RDESC.IPCS and RDESC.TCPCS).

If neither RXCSUM.IPOFLD nor RXCSUM.TUOFLD is set, the Checksum Error bits (IPE and TCPE) is 0b for all packets.

Supported Frame Types include:

- Ethernet II
- Ethernet SNAP

Note: See Table 3-6 for the **82544GC/EI** supported receive checksum capabilities.

Table 3-5. Supported Receive Checksum Capabilities

Packet Type	HW IP Checksum Calculation	HW TCP/UDP Checksum Calculation
IPv4 packets	Yes	Yes
IPv6 packets	No (n/a)	Yes
IPv6 packet with next header options: Hop-by-Hop options Destinations options Routing Fragment	 No (n/a) No (n/a) No (n/a) No (n/a)	 Yes Yes Yes No
IPv4 tunnels: IPv4 packet in an IPv4 tunnel IPv6 packet in an IPv4 tunnel	 No Yes (IPv4)	 No Yes[a]
IPv6 tunnels: IPv4 packet in an IPv6 tunnel IPv6 packet in an IPv6 tunnel	 No No	 No No
Packet is an IPv4 fragment	Yes	No
Packet is greater than 1552 bytes; (LPE=1b)[b]	Yes	Yes
Packet has 802.3ac tag	Yes	Yes

Table 3-5. Supported Receive Checksum Capabilities

Packet Type	HW IP Checksum Calculation	HW TCP/UDP Checksum Calculation
IPv4 Packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet has TCP or UDP options	Yes	Yes
IP header's protocol field contains a protocol # other than TCP or UDP.	Yes	No

a. The IPv6 header portion can include supported extension headers as described in the IPv6 Filter section.
b.For the **82541xx** and **82547GI/EI**, frame sizes greater than 2 KB require full-duplex operation.

Table 3-6. 82544GC/EI Supported Receive Checksum Capabilities

Packet Type	HW IP Checksum Calculation	HW TCP/UDP Checksum Calculation
IP v4 packets	Yes	Yes
IP v6 packets (no IP checksum in IPv6)	No	No
Packet is an IP fragment	Yes	No
Packet is greater than 1552 bytes; (LPE=1)	Yes	Yes
Packet has 802.3ac tag	Yes	Yes
Packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet has TCP or UDP options	Yes	Yes
IP header's protocol field contains a protocol other than TCP or UDP.	Yes	No

Table 3-5 lists the general details about what packets are processed. In more detail, the packets are passed through a series of filters (Section 3.2.9.1 through Section 3.2.9.5) to determine if a receive checksum is calculated.

Note: (Section 3.2.9.1 through Section 3.2.9.5) does not apply to the **82544GC/EI**.

3.2.9.1 MAC Address Filter

This filter checks the MAC destination address to be sure it is valid (IA match, broadcast, multicast, etc.). The receive configuration settings determine which MAC addresses are accepted. See the various receive control configuration registers such as RCTL (RTCL.UPE, RCTL.MPE, RCTL.BAM), MTA, RAL, and RAH.

3.2.9.2 SNAP/VLAN Filter

This filter checks the next headers looking for an IP header. It is capable of decoding Ethernet II, Ethernet SNAP, and IEEE 802.3ac headers. It skips past any of these intermediate headers and looks for the IP header. The receive configuration settings determine which next headers are accepted. See the various receive control configuration registers such as RCTL (RCTL.VFE), VET, and VFTA.

3.2.9.3 IPv4 Filter

This filter checks for valid IPv4 headers. The version field is checked for a correct value (4). IPv4 headers are accepted if they are any size greater than or equal to 5 (dwords). If the IPv4 header is properly decoded, the IP checksum is checked for validity. The RXCSUM.IPOFL bit must be set for this filter to pass.

3.2.9.4 IPv6 Filter

This filter checks for valid IPv6 headers, which are a fixed size and have no checksum. The IPv6 extension headers accepted are: Hop-by-Hop, Destination Options, and Routing. The maximum size next header accepted is 16 dwords (64 bytes).

All of the IPv6 extension headers supported by the Ethernet controller have the same header structure:

Byte 0	Byte 1	Byte 2	Byte 3
Next Header	Hdr Ext Len		

- NEXT HEADER is a value that identifies the header type. The supported IPv6 next headers values are:
  - Hop-by-Hop = 00h
  - Destination Options = 3Ch
  - Routing = 2Bh
- HDR EXT LEN is the 8 byte count of the header length, not including the first 8 bytes. For example, a value of 3 means that the total header size including the NEXT HEADER and HDR EXT LEN fields is 32 bytes (8 + 3*8).
  - The RXCSUM.IPV6OFL bit must be set for this filter to pass.

3.2.9.5 UDP/TCP Filter

This filter checks for a valid UDP or TCP header. The prototype next header values are 11h and 06h, respectively. The RXCSUM.TUOFL bit must be set for this filter to pass.

3.3 Packet Transmission

The transmission process for regular (non-TCP Segmentation packets) involves:

- The protocol stack receives from an application a block of data that is to be transmitted.

- The protocol stack calculates the number of packets required to transmit this block based on the MTU size of the media and required packet headers.
- For each packet of the data block:
  - — Ethernet, IP and TCP/UDP headers are prepared by the stack.
  - — The stack interfaces with the software device driver and commands the driver to send the individual packet.
  - — The driver gets the frame and interfaces with the hardware.
  - — The hardware reads the packet from host memory (via DMA transfers).
  - — The driver returns ownership of the packet to the Network Operating System (NOS) when the hardware has completed the DMA transfer of the frame (indicated by an interrupt).

Output packets are made up of pointer–length pairs constituting a descriptor chain (so called descriptor based transmission). Software forms transmit packets by assembling the list of pointer–length pairs, storing this information in the transmit descriptor, and then updating the on–chip transmit tail pointer to the descriptor. The transmit descriptor and buffers are stored in host memory. Hardware typically transmits the packet only after it has completely fetched all packet data from host memory and deposited it into the on-chip transmit FIFO. This permits TCP or UDP checksum computation, and avoids problems with PCI underruns.

3.3.1 Transmit Data Storage

Data are stored in buffers pointed to by the descriptors. Alignment of data is on an arbitrary byte boundary with the maximum size per descriptor limited only to the maximum allowed packet size (16288 bytes). A packet typically consists of two (or more) descriptors, one (or more) for the header and one for the actual data. Some software implementations copy the header(s) and packet data into one buffer and use only one descriptor per transmitted packet.

3.3.2 Transmit Descriptors

The Ethernet controller provides three types of transmit descriptor formats.

The original descriptor is referred to as the “legacy” descriptor format. The two other descriptor types are collectively referred to as extended descriptors. One of them is similar to the legacy descriptor in that it points to a block of packet data. This descriptor type is called the TCP/IP Data Descriptor and is a replacement for the legacy descriptor since it offers access to new offloading capabilities. The other descriptor type is fundamentally different as it does not point to packet data. It merely contains control information which is loaded into registers of the controller and affect the processing of future packets. The following sections describe the three descriptor formats.

The extended descriptor types are accessed by setting the TDESC.DEXT bit to 1b. If this bit is set, the TDESC.DTYP field is examined to control the interpretation of the remaining bits of the descriptor. Table 3-7 shows the generic layout for all extended descriptors. Fields marked as NR are not reserved for any particular function and are defined on a per-descriptor type basis. Notice that the DEXT and DTYP fields are non-contiguous in order to accommodate legacy mode operation. For legacy mode operation, bit 29 is set to 0b and the descriptor is defined in Section 3.3.3.

Table 3-7. Transmit Descriptor (TDESC) Layout

	63 30	29	28 24	23 20	19 0
0	Buffer Address [63:0]				
8	NR	DEXT	NR	DTYP	NR

3.3.3 Legacy Transmit Descriptor Format

To select legacy mode operation, bit 29 (TDESC.DEXT) should be set to 0b. In this case, the descriptor format is defined as shown in Table 3-8. The address and length must be supplied by software. Bits in the command byte are optional, as are the Checksum Offset (CSO), and Checksum Start (CSS) fields.

Table 3-8. Transmit Descriptor (TDESC) Layout – Legacy Mode

	63 48	47 40	39 36	35 32	31 24	23 16	15 0
0	Buffer Address [63:0]						
8	Special	CSS	RSV	STA	CMD	CSO	Length

Table 3-9. Transmit Descriptor Legacy Descriptions

Transmit Descriptor Legacy	Description
Buffer Address	Buffer Address Address of the transmit descriptor in the host memory. Descriptors with a null address transfer no data. If they have the RS bit in the command byte set (TDESC.CMD), then the DD field in the status word (TDESC.STATUS) is written when the hardware processes them.
Length	Length is per segment. The maximum length associated with any single legacy descriptor is 16288 bytes. Although a buffer as short as one byte is allowed, the total length of the packet, before padding and CRC insertion must be at least 48 bytes. Length can be up to a default value of 16288 bytes per descriptor, and 16288 bytes total. In other words, the length of the buffer pointed to by one descriptor, or the sum of the lengths of the buffers pointed to by the descriptors can be as large as the maximum allowed transmit packet. Descriptors with zero length transfer no data. If they have the RS bit in the command byte set (TDESC.CMD), then the DD field in the status word (TDESC.STATUS) is written when the hardware processes them.
CSO	Checksum Offset The Checksum offset field indicates where, relative to the start of the packet, to insert a TCP checksum if this mode is enabled. (Insert Checksum bit (IC) is set in TDESC.CMD). Hardware ignores CSO unless EOP is set in TDESC.CMD. CSO is provided in unit of bytes and must be in the range of the data provided to the Ethernet controller in the descriptor. (CSO < length - 1). Should be written with 0b for future compatibility.

Transmit Descriptor Legacy	Description
CMD	Command field See Section 3.3.3.1 for a detailed field description.
STA	Status field See Section 3.3.3.2 for a detailed field description.
RSV	Reserved Should be written with 0b for future compatibility.
CSS	Checksum Start Field The Checksum start field (TDESC.CSS) indicates where to begin computing the checksum. The software must compute this offset to back out the bytes that should not be included in the TCP checksum. CSS is provided in units of bytes and must be in the range of data provided to the Ethernet controller in the descriptor (CSS < length). For short packets that ar padded by the software, CSS must be in the range of the unpadded data length. A value of 0b corresponds to the first byte in the packet. CSS must be set in the first descriptor of the packet.
Special	Special Field See the notes that follow this table for a detailed field description.

Notes:

1. Even though CSO and CSS are in units of bytes, the checksum calculation typically works on 16-bit words. Hardware does not enforce even byte alignment.
2. Hardware does not add the 802.1Q EtherType or the VLAN field following the 802.1Q EtherType to the checksum. So for VLAN packets, software can compute the values to back out only on the encapsulated packet rather than on the added fields.
3. Although the Ethernet controller can be programmed to calculate and insert TCP checksum using the legacy descriptor format as described above, it is recommended that software use the newer TCP/IP Context Transmit Descriptor Format. This newer descriptor format allows the hardware to calculate both the IP and TCP checksums for outgoing packets. See Section 3.3.5 for more information about how the new descriptor format can be used to accomplish this task.

3.3.3.1 Transmit Descriptor Command Field Format

The CMD byte stores the applicable command and has fields shown in Table 3-10.

Table 3-10. Transmit Command (TDESC.CMD) Layout

7	6	5	4	3	2	1	0
IDE	VLE	DEXT	RSV RPS[a]	RS	IC	IFCS	EOP

a. **82544GC/EI** only.

TDESC.CMD	Description
IDE (bit 7)	Interrupt Delay Enable When set, activates the transmit interrupt delay timer. The Ethernet controller loads a countdown register when it writes back a transmit descriptor that has RS and IDE set. The value loaded comes from the IDV field of the Interrupt Delay (TIDV) register. When the count reaches 0, a transmit interrupt occurs if transmit descriptor write-back interrupts (IMS.TXDW) are enabled. Hardware always loads the transmit interrupt counter whenever it processes a descriptor with IDE set even if it is already counting down due to a previous descriptor. If hardware encounters a descriptor that has RS set, but not IDE, it generates an interrupt immediately after writing back the descriptor. The interrupt delay timer is cleared.
VLE (bit 6)	VLAN Packet Enable When set, indicates that the packet is a VLAN packet and the Ethernet controller should add the VLAN Ethertype and an 802.1q VLAN tag to the packet. The Ethertype field comes from the VET register and the VLAN tag comes from the special field of the TX descriptor. The hardware inserts the FCS/CRC field in that case. When cleared, the Ethernet controller sends a generic Ethernet packet. The IFCS controls the insertion of the FCS field in that case. In order to have this capability CTRL.VME bit should also be set, otherwise VLE capability is ignored. VLE is valid only when EOP is set.
DEXT (bit 5)	Extension (0b for legacy mode). Should be written with 0b for future compatibility.
RPS RSV (bit 4)	Report Packet Sent When set, the **82544GC/EI** defers writing the DD bit in the status byte (DESC.STATUS) until the packet has been sent, or transmission results in an error such as excessive collisions. It is used is cases where the software must know that the packet has been sent, and not just loaded to the transmit FIFO. The **82544GC/EI** might continue to prefetch data from descriptors logically after the one with RPS set, but does not advance the descriptor head pointer or write back any other descriptor until it sent the packet with the RPS set. RPS is valid only when EOP is set. This bit is reserved and should be programmed to 0b for all Ethernet controllers except the **82544GC/EI**.
RS (bit 3)	Report Status When set, the Ethernet controller needs to report the status information. This ability may be used by software that does in-memory checks of the transmit descriptors to determine which ones are done and packets have been buffered in the transmit FIFO. Software does it by looking at the descriptor status byte and checking the Descriptor Done (DD) bit.

TDESC.CMD	Description
IC (bit 2)	Insert Checksum When set, the Ethernet controller needs to insert a checksum at the offset indicated by the CSO field. The checksum calculations are performed for the entire packet starting at the byte indicated by the CCS field. IC is ignored if CSO and CCS are out of the packet range. This occurs when (CSS ≥ length) OR (CSO ≥ length - 1). IC is valid only when EOP is set.
IFCS (bit 1)	Insert FCS Controls the insertion of the FCS/CRC field in normal Ethernet packets. IFCS is valid only when EOP is set.
EOP (bit 0)	End Of Packet When set, indicates the last descriptor making up the packet. One or many descriptors can be used to form a packet.

Notes:

1. VLE, IFCS, and IC are qualified by EOP. That is, hardware interprets these bits ONLY when EOP is set.
2. Hardware only sets the DD bit for descriptors with RS set.
3. Descriptors with the null address (0b) or zero length transfer no data. If they have the RS bit set then the DD field in the status word is written when hardware processes them.
4. Although the transmit interrupt may be delayed, the descriptor write-back requested by setting the RS bit is performed without delay unless descriptor write-back bursting is enabled.

3.3.3.2 Transmit Descriptor Status Field Format

The STATUS field stores the applicable transmit descriptor status and has the fields shown in Table 3-11.

The transmit descriptor status field is only present in cases where RS (or RPS for the **82544GC/EI** only) is set in the command field.

Table 3-11. Transmit Status Layout

3	2	1	0
RSV TU[a]	LC	EC	DD

a. **82544GC/EI** only.

TDESC.STATUS	Description
TU RSV (bit 3)	Transmit Underrun Indicates a transmit underrun event occurred. Transmit Underrun might occur if Early Transmits are enabled (based on ETT.Txthreshold value) and the **82544GC/EI** was not able to complete the early transmission of the packet due to lack of data in the packet buffer. This does not necessarily mean the packet failed to be eventually transmitted. The packet is successfully re-transmitted if the TCTL.NRTU bit is cleared (and excessive collisions do not occur). This bit is reserved and should be programmed to 0b for all Ethernet controllers except the **82544GC/EI**.
LC (bit 2)	Late Collision Indicates that late collision occurred while working in half-duplex mode. It has no meaning while working in full-duplex mode. Note that the collision window is speed dependent: 64 bytes for 10/100 Mb/s and 512 bytes for 1000 Mb/s operation.
EC (bit 1)	Excess Collisions Indicates that the packet has experienced more than the maximum excessive collisions as defined by TCTL.CT control field and was not transmitted. It has no meaning while working in full-duplex mode.
DD (bit 0)	Descriptor Done Indicates that the descriptor is finished and is written back either after the descriptor has been processed (with RS set) or for the **82544GC/EI**, after the packet has been transmitted on the wire (with RPS set).

Note: The DD bit reflects status of all descriptors up to and including the one with the RS bit set (or RPS for the **82544GC/EI**).

3.3.4 Transmit Descriptor Special Field Format

The SPECIAL field is used to provide the 802.1q/802.1ac tagging information.

When CTRL.VME is set to 1b, all packets transmitted from the Ethernet controller that have VLE set in the TDESC.CMD are sent with an 802.1Q header added to the packet. The contents of the header come from the transmit descriptor special field and from the VLAN type register. The special field is ignored if the VLE bit in the transmit descriptor command field is 0b. The special field is valid only for descriptors with EOP set to 1b in TDESC.CMD.

Table 3-12. Special Field (TDESC.SPECIAL) Layout

15 13	12	11 0
PRI	CFI	VLAN

TDESC.SPECIAL	Description
PRI	User Priority 3 bits that provide the VLAN user priority field to be inserted in the 802.1Q tag.
CFI	Canonical Form Indicator.
VLAN	VLAN Identifier 12 bits that provide the VLAN identifier field to be inserted in the 802.1Q tag.

3.3.5 TCP/IP Context Transmit Descriptor Format

The TCP/IP context transmit descriptor provides access to the enhanced checksum offload facility available in the Ethernet controller. This feature allows TCP and UDP packet types to be handled more efficiently by performing additional work in hardware, thus reducing the software overhead associated with preparing these packets for transmission.

The TCP/IP context transmit descriptor does not point to packet data as a data descriptor does. Instead, this descriptor provides access to an on-chip context that supports the transmit checksum offloading feature of the controller. A “context” refers to a set of registers loaded or unloaded as a group to provide a particular function.

The context is explicit and directly accessible via the TCP/IP context transmit descriptor. The context is used to control the checksum offloading feature for normal packet transmission.

The Ethernet controller automatically selects the appropriate legacy or normal context to use based on the current packet transmission.

While the architecture supports arbitrary ordering rules for the various descriptors, there are restrictions including:

- Context descriptors should not occur in the middle of a packet.
- Data descriptors of different packet types (legacy or normal) should not be intermingled except at the packet level.

All contexts control calculation and insertion of up to two checksums. This portion of the context is referred to as the checksum context.

In addition to checksum context, the segmentation context adds information specific to the segmentation capability. This additional information includes the total payload for the message (TDESC.PAYLEN), the total size of the header (TDESC.HDRLEN), the amount of payload data that should be included in each packet (TDESC.MSS), and information about what type of protocol (TCP, IPv4, IPv6, etc.) is used. This information is specific to the segmentation capability and is therefore ignored for context descriptors that do not have the TSE bit set.

Because there are dedicated resources on-chip for the normal context, the context remains constant until it is modified by another context descriptor. This means that a context can be used for multiple packets (or multiple segmentation blocks) unless a new context is loaded prior to each new packet. Depending on the environment, it may be completely unnecessary to load a new context for each packet. For example, if most traffic generated from a given node is standard TCP frames, this context could be set up once and used for many frames. Only when some other frame type is required would a new context need to be loaded by software. After the “non-standard” frame is transmitted, the “standard” context would be setup once more by software. This method avoids the “extra descriptor per packet” penalty for most frames. The penalty can be eliminated altogether if software elects to use TCP/IP checksum offloading only for a single frame type, and thus performs those operations in software for other frame types.

This same logic can also be applied to the segmentation context, though the environment is a more restrictive one. In this scenario, the host is commonly asked to send a message of the same type, TCP/IP for instance, and these messages also have the same total length and same maximum segment size (MSS). In this instance, the same segmentation context could be used for multiple TCP messages that require hardware segmentation. The limitations of this scenario and the relatively small performance advantage make this approach unlikely; however, it is useful in understanding the underlying mechanism.

3.3.6 TCP/IP Context Descriptor Layout

The following section describes the layout of the TCP/IP context transmit descriptor.

To select this descriptor format, bit 29 (TDESC.DEXT) must be set to 1b and TDESC.DTYP must be set to 0000b. In this case, the descriptor format is defined as shown in Table 3-13.

Note that the TCP/IP context descriptor does not transfer any packet data. It merely prepares the checksum hardware for the TCP/IP Data descriptors that follow.

Table 3-13. Transmit Descriptor (TDESC) Layout – (Type = 0000b)

	63 48	47 40	39 32	31 16	15 8	7 0
0	TUCSE	TUCSO	TUCSS	IPCSE	IPCSO	IPCSS

	63 48	47 40	39 36	35 32	31 24	23 20	19 0
8	MSS	HDRLEN	RSV	STA	TUCMD	DTYP	PAYLEN

Note: The first quadword of this descriptor type contains parameters used to calculate the two checksums which may be offloaded.

Table 3-14. Transmit Descriptor (TDESC) Layout

Transmit Descriptor Offload	Description
TUCSE	TCP/UDP Checksum Ending Defines the ending byte for the TCP/UDP checksum offload feature. Setting TUCSE field to 0b indicates that the checksum covers from TUCCS to the end of the packet.
TUCSO	TCP/UDP Checksum Offset Defines the offset where to insert the TCP/UDP checksum field in the packet data buffer. This is used in situations where the software needs to calculate partial checksums (TCP pseudo-header, for example) to include bytes which are not contained within the range of start and end. If no partial checksum is required, software must write a value of 0b.
TUCSS	TCP/UDP Checksum Start Defines the starting byte for the TCP/UDP checksum offload feature. It must be defined even if checksum insertion is not desired for some reason. When setting the TCP segmentation context, TUCSS is used to indicate the start of the TCP header.
IPCSE	IP Checksum Ending Defines the ending byte for the IP checksum offload feature. It specifies where the checksum should stop. A 16-bit value supports checksum offloading of packets as large as 64KB. Setting IPCSE field to 0b indicates that the checksum covers from IPCCS to the end of the packet. In this way, the length of the packet does not need to be calculated.
IPCSO	IP Checksum Offset The IPCSO field specifies where the resulting IP checksum should be placed. It is limited to the first 256 bytes of the packet and must be less than or equal to the total length of a given packet. If this is not the case, the checksum is not inserted.
IPCSS	IP Checksum Start IPCSS specifies the byte offset from the start of the transferred data to the first byte in be included in the checksum. Setting this value to 0b means the first byte of the data would be included in the checksum. Note that the maximum value for this field is 255. This is adequate for typical applications. The IPCSS value needs to be less than the total transferred length of the packet. If this is not the case, the results are unpredictable. IPCSS must be defined even if checksum insertion is not desired for some reason. When setting the TCP segmentation context, IPCSS is used to indicate the start of the IP header.
MSS	Maximum Segment Size Controls the Maximum Segment Size. This specifies the maximum TCP or UDP payload “segment” sent per frame, not including any header. The total length of each frame (or “section”) sent by the TCP Segmentation mechanism (excluding 802.3ac tagging and Ethernet CRC) is MSS bytes + HRDLEN. The one exception is the last packet of a TCP segmentation context which is (typically) shorter than “MSS+HDRLEN”. This field is ignored if TDESC.TSE is not set.
HDRLEN	Header Length Specifies the length (in bytes) of the header to be used for each frame (or “section”) of a TCP Segmentation operation. The first HDRLEN bytes fetched from data descriptor(s) are stored internally and used as a prototype header for each section, and are pre-pended to each payload segment to form individual frames. For UDP packets this is normally equal to “UDP checksum offset + 2”. For TCP packets it is normally equal to “TCP checksum offset + 4 + TCP header option bytes”. This field is ignored if TDESC.TSE is not set.

Transmit Descriptor Offload	Description
RSV	Reserved Should be programmed to 0b for future compatibility.
STA	TCP/UDP Status field Provides transmit status indication. Section 3.3.6.2 provides the bit definition for the TDESC.STA field.
TUCMD	TCP/UDP command field The command field provides options that control the checksum offloading, along with some of the generic descriptor processing functions. Section 3.3.6.1 provides the bit definitions for the TDESC.TUCMD field.
DTYP	Descriptor Type Set to 0000b for TCP/IP context transmit descriptor type.
PAYLEN	The packet length field (TDESC.PAYLEN) is the total number of payload bytes for this TCP Segmentation offload context (i.e., the total number of payload bytes that could be distributed across multiply frames after TCP segmentation is performed). Following the fetch of the prototype header, PAYLEN specifies the length of data that is fetched next from data descriptor(s). This field is also used to determine when “last-frame” processing needs to be performed. Typically, a new data descriptor is used to denote the start of the payload data buffer(s), but this is not required. PAYLEN specification should not include any header bytes. There is no restriction on the overall PAYLEN specification with respect to the transmit FIFO size, once the MSS and HDRLEN specifications are legal. This field is ignored if TDESC.TSE is not set. Refer to Section 3.5 for details on the TCP Segmentation off-loading feature.

Notes:

1. A number of the fields are ignored if the TCP Segmentation enable bit (TDESC.TSE) is cleared, denoting that the descriptor does not refer to the TCP segmentation context.
2. Maximum limits for the HDRLEN and MSS fields are dictated by the lengths variables. However, there is a further restriction that for any TCP Segmentation operation, the hardware must be capable of storing a complete section (completely-built frame) in the transmit FIFO prior to transmission. Therefore, the sum of MSS + HDRLEN must be at least 80 bytes less than the allocated size of the transmit FIFO.

3.3.6.1 TCP/UDP Offload Transmit Descriptor Command Field

The command field (TDESC.TUCMD) provides options to control the TCP segmentation, along with some of the generic descriptor processing functions.

Table 3-15. Command Field (TDESC.TUCMD) Layout

7	6	5	4	3	2	1	0
IDE	RSV	DEXT	RSV	RS	TSE	IP	TCP

TDESC.TUCMD	Description
IDE (bit 7)	Interrupt Delay Enable IDE activates the transmit interrupt delay timer. Hardware loads a countdown register when it writes back a transmit descriptor that has the RS bit and the IDE bit set. The value loaded comes from the IDV field of the Interrupt Delay (TIDV) register. When the count reaches 0, a transmit interrupt occurs. Hardware always loads the transmit interrupt counter whenever it processes a descriptor with IDE set even if it is already counting down due to a previous descriptor. If hardware encounters a descriptor that has RS set, but not IDE, it generates an interrupt immediately after writing back the descriptor. The interrupt delay timer is cleared.
RSV (Bit 6)	Reserved. Set to 0b for future compatibility.
DEXT(Bit 5)	Descriptor Extension Must be 1b for this descriptor type.
RSV (Bit 4)	Reserved. Set to 0b for future compatibility.
RS (Bit 3)	Report Status RS tells the hardware to report the status information for this descriptor. Because this descriptor does not transmit data, only the DD bit in the status word is valid. Refer to Section 3.3.6.2 for the layout of the status field.
TSE (Bit 2)	TCP Segmentation Enable TSE indicates that this descriptor is setting the TCP segmentation context. If this bit is not set, the checksum offloading context for normal (non-"TCP Segmentation") packets is written. When a descriptor of this type is processed the Ethernet controller immediately updates the context in question (TCP Segmentation or checksum offloading) with values from the descriptor. This means that if any normal packets or TCP Segmentation packets are in progress (a descriptor with EOP set has not been received for the given context), the results are likely to be undesirable.
IP (Bit 1)	Packet Type (IPv4 = 1b, IPv6 = 0b) Identifies what type of IP packet is used in the segmentation process. This is necessary for hardware to know where the IP Payload Length field is located. This does not override the checksum insertion bit, IXSM.
IP (Bit 1) **82544GC/EI** only	Packet Type (IP = 1b) Identifies the packet as an IP packet. The purpose of this bit is to enable/disable the updating of the IP header during the segmentation process. This does not override the checksum insertion bit, IXSM.
TCP (bit 0)	Packet Type (TCP = 1b) Identifies the packet as either TCP or UDP (non-TCP). This affects the processing of the header information.

Note:

1. The IDE, DEXT, and RS bits are valid regardless of the state of TSE. All other bits are ignored if TSE = 0b.
2. The TCP Segmentation feature also provides access to a generic block send function and may be useful for performing "segmentation offload" in which the header information is constant. By clearing both the TCP and IP bits, a block of data may be broken down into frames of a given size, a constant, arbitrary length header may be pre-pended to each frame, and two checksums optionally added.

3.3.6.2 TCP/UDP Offload Transmit Descriptor Status Field

Four bits are reserved to provide transmit status, although only one is currently assigned for this specific descriptor type. The status word is only written back to host memory in cases where the RS is set in the command.

Table 3-16. Transmit Status Layout

3	2	1	0
RSV			DD

TDESC.STA	Description
RSV	Reserved Reserved for future use. Reads as 0b.
DD (bit 0)	Descriptor Done Indicates that the descriptor is finished and is written back after the descriptor has been processed.

3.3.7 TCP/IP Data Descriptor Format

The TCP/IP data descriptor is the companion to the TCP/IP context transmit descriptor described in the previous section. This descriptor type provides similar functionality to the legacy mode descriptor but also integrates the checksum offloading and TCP Segmentation feature.

To select this descriptor format, bit 29 in the command field (TDESC.DEXT) must be set to 1b and TDESC.DTYP must be set to 0001b. In this case, the descriptor format is defined as shown in Table 3-17.

Table 3-17. Transmit Descriptor (TDESC) Layout – (Type = 0001b)

	63 – 48	47 – 40	39 – 36	35 – 32	31 – 24	23 – 20	19 – 0
0	Address [63:0]						
8	Special	POPTS	RSV	STA	DCMD	DTYP	DTALEN

Transmit Descriptor	Description
Address	Data buffer address Address of the data buffer in the host memory which contains a portion of the transmit packet.
DTALEN	Data Length Field Total length of the data pointed to by this descriptor, in bytes. For data descriptors not associated with a TCP Segmentation operation (TDESC.TSE not set), the descriptor lengths are subject to the same restrictions specified for legacy descriptors (the sum of the lengths of the data descriptors comprising a single packet must be at least 80 bytes less than the allocated size of the transmit FIFO.)
DTYP	Data Type Set to 0001b to identify this descriptor as a TCP/IP data descriptor.
DCMD	Descriptor Command Field Provides options that control some of the generic descriptor processing features. Refer to Section 3.3.7.1 for bit definitions of the DCMD field.
STA	TCP/IP Status field Provides transmit status indication. Section 3.3.7.2 provides the bit definition for the TDESC.STA field.
RSV	Reserved Set to 0b for future compatibility.
POPTS	Packet Option Field Provides a number of options which control the handling of this packet. This field is ignored except on the first data descriptor of a packet. Section 3.3.7.3 provides the bit definition for the TDESC.POPTS field.
Special	Special field The Special field is used to provide 802.1q tagging information. This field is only valid in the last descriptor of the given packet (qualified by the EOP bit).

3.3.7.1 TCP/IP Data Descriptor Command Field

The Command field provides options that control checksum offloading and TCP segmentation features along with some of the generic descriptor processing features.

Table 3-18. Command Field (TDESC.DCMD) Layout

7	6	5	4	3	2	1	0
IDE	VLE	DEXT	RSV RPS[a]	RS	TSE	IFCS	EOP

a. **82544GC/EI** only.

TDESC.DCMD	Description
IDE (bit 7)	Interrupt Delay Enable When set, activates the transmit interrupt delay timer. Hardware loads a countdown register when it writes back a transmit descriptor that has RS and IDE set. The value loaded comes from the IDV field of the Interrupt Delay (TIDV) register. When the count reaches 0, a transmit interrupt occurs if enabled. Hardware always loads the transmit interrupt counter whenever it processes a descriptor with IDE set even if it is already counting down due to a previous descriptor. If hardware encounters a descriptor that has RS set, but not IDE, it generates an interrupt immediately after writing back the descriptor. The interrupt delay timer is cleared.
VLE (bit 6)	VLAN Enable When set, indicates that the packet is a VLAN packet and the hardware should add the VLAN Ethertype and an 802.1q VLAN tag to the packet. The Ethertype should come from the VET register and the VLAN data comes from the special field of the TX descriptor. The hardware in that case appends the FCS/CRC. Note that the CTRL.VME bit should also be set. If the CTRL.VME bit is not set, the Ethernet controller does not insert VLAN tags on outgoing packets and it sends generic Ethernet packets. The IFCS controls the insertion of the FCS/CRC in that case. VLE is only valid in the last descriptor of the given packet (qualified by the EOP bit).
DEXT (Bit 5)	Descriptor Extension Must be 1b for this descriptor type
RPS RSV (bit 4)	Report Packet Sent RPS is used in cases where software must know that a packet has been sent on the wire, not just that it has been loaded into the **82544GC/EI** controller's internal packet buffer. When set, hardware defers writing the DD bit in the status byte until the packet has been sent, or transmission results in an error such as excess collisions. Hardware can continue to pre-fetch data from descriptors logically after the one with RPS set, but does not advance the head pointer or write back any other descriptors until it has sent the packet with RPS set. For a TCP Segmentation context, the RPS bit indicates to the **82544GC/EI** that the descriptor status should only be written back once all packets that make up the given TCP Segmentation context had been sent. This bit is reserved and should be programmed to 0b for all Ethernet controllers except the **82544GC/EI**.
RS (bit 3)	Report Status When set, tells the hardware to report the status information for this descriptor as soon as the corresponding data buffer has been fetched and stored in the controller's internal packet buffer.

TDESC.DCMD	Description
TSE (bit 2)	TCP Segmentation Enable TSE indicates that this descriptor is part of the current TCP Segmentation context. If this bit is not set, the descriptor is part of the "normal" context.
IFCS (Bit 1)	Insert IFCS Controls the insertion of the FCS/CRC field in normal Ethernet packets. IFCS is only valid in the last descriptor of the given packet (qualified by the EOP bit).
EOP (Bit 0)	End Of Packet The EOP bit indicates that the buffer associated with this descriptor contains the last data for the packet or for the given TCP Segmentation context. In the case of a TCP Segmentation context, the DTALEN length of this descriptor should match the amount remaining of the original PAYLEN. If it does not, the TCP Segmentation context is terminated but the end of packet processing may be incorrectly performed. These abnormal termination events are counted in the TSCTFC statistics register.

Note: The VLE, IFCS, and VLAN fields are only valid in certain descriptors. If TSE is enabled, the VLE, IFCS, and VLAN fields are only valid in the first data descriptor of the TCP segmentation context. If TSE is not enabled, then these fields are only valid in the last descriptor of the given packet (qualified by the EOP bit).

3.3.7.2 TCP/IP Data Descriptor Status Field

Four bits are reserved to provide transmit status, although only the DD is valid[1]. The status word is only written back to host memory in cases where the RS bit is set in the command field. The DD bit indicates that the descriptor is finished and is written back after the descriptor has been processed.

Table 3-19. Transmit Status Layout

3	2	1	0
RSV TU[a]	LC	EC	DD

a. **82544GC/EI** only.

TDESC.STA	Description
Reserved	Reserved

1. Unless the RPS bit is set in the descriptor (**82544GC/EI** only).

TDESC.STA	Description
LC (bit2)	Late Collision Indicates that late collision occurred while working in half-duplex mode. It has no meaning while working in full-duplex mode. Note that the collision window is speed dependent: 64 bytes for 10/100 Mb/s and 512 bytes for 1000 Mb/s operation.
EC (bit 1)	Excess Collision Indicates that the packet has experienced more than the maximum excessive collisions as defined by TCTL.CT control field and was not transmitted. Is has no meaning while working in full-duplex mode.
DD (bit 0)	Descriptor Done Indicates that the descriptor is done and is written back either after the descriptor has been processed (with RS set), or for the **82554GC/EI** only, after the packet has been transmitted on the wire (with RPS set).

3.3.7.3 TCP/IP Data Descriptor Option Field

The POPTS field provides a number of options which control the handling of this packet. This field is ignored except on the first data descriptor of a packet.

Table 3-20. Packet Options Field (TDESC.POPTS) Layout

7	6	5	4	3	2	1	0
RSV	RSV	RSV	RSV	RSV	RSV	TXSM	IXSM

TDESC.POPTS	Description
RSV (bit 2-7)	Reserved Should be written with 0b for future compatibility.
TXSM (bit1)	Insert TCP/UDP Checksum Controls the insertion of the TCP/UDP checksum. If not set, the value placed into the checksum field of the packet data is not modified, and is placed on the wire. When set, TCP/UDP checksum field is modified by the hardware. Valid only in the first data descriptor for a given packet or TCP Segmentation context.
IXSM (bit 0)	Insert IP Checksum Controls the insertion of the IP checksum. If not set, the value placed into the checksum field of the packet data is not modified and is placed on the wire. When set, the IP checksum field is modified by the hardware. Valid only in the first data descriptor for a given packet or TCP Segmentation context.

3.3.7.4 TCP/IP Data Descriptor Special Field

The SPECIAL field is used to provide the 802.1q/802.3ac tagging information.

When CTRL.VME is set to 1b, all packets transmitted from the Ethernet controller that has VLE set in the DCMD field is sent with an 802.1Q header added to the packet. The contents of the header come from the transmit descriptor special field and from the VLAN type register. The special field is ignored if the VLE bit in the transmit descriptor command field is 0b. The special field is valid only when EOP is set.

Table 3-21. Special Field (TDESC.SPECIAL) Layout

15 … 13	12	11 … 0
PRI	CFI	VLAN

TDESC.SPECIAL	Description
PRI	User Priority Three bits that provide the VLAN user priority field to be inserted in the 802.1Q tag.
CFI	Canonical Form Indicator
VLAN	VLAN Identifier 12 bits that provide the VLAN identifier field to be inserted in the 802.1Q tag.

3.4 Transmit Descriptor Ring Structure

The transmit descriptor ring structure is shown in Figure 3-4. A pair of hardware registers maintains the transmit queue. New descriptors are added to the ring by writing descriptors into the circular buffer memory region and moving the ring's tail pointer. The tail pointer points one entry beyond the last hardware owned descriptor (but at a point still within the descriptor ring). Transmission continues up to the descriptor where head equals tail at which point the queue is empty.

Descriptors passed to hardware should not be manipulated by software until the head pointer has advanced past them.

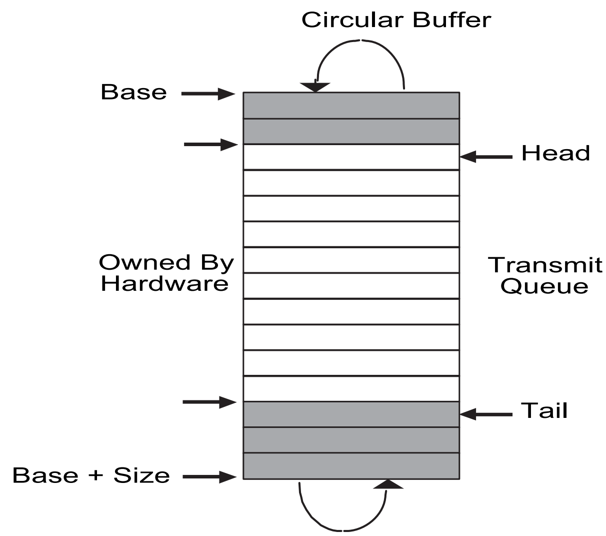


Figure 3-4. Transmit Descriptor Ring Structure

Shaded boxes in Figure 3-4 represent descriptors that have been transmitted but not yet reclaimed by software. Reclaiming involves freeing up buffers associated with the descriptors.

The transmit descriptor ring is described by the following registers:

- Transmit Descriptor Base Address registers (TDBAL and TDBAH)

  These registers indicate the start of the descriptor ring buffer. This 64-bit address is aligned on a 16-byte boundary and is stored in two consecutive 32-bit registers. TDBAL contains the lower 32-bits; TDBAH contains the upper 32 bits. Hardware ignores the lower 4 bits in TDBAL.

- Transmit Descriptor Length register (TDLEN)

  This register determines the number of bytes allocated to the circular buffer. This value must be 128 byte aligned.

- Transmit Descriptor Head register (TDH)

  This register holds a value which is an offset from the base, and indicates the in–progress descriptor. There can be up to 64K descriptors in the circular buffer. Reading this register returns the value of "head" corresponding to descriptors already loaded in the output FIFO.

- Transmit Descriptor Tail register (TDT)

  This register holds a value which is an offset from the base, and indicates the location beyond the last descriptor hardware can process. This is the location where software writes the first new descriptor.

The base register indicates the start of the circular descriptor queue and the length register indicates the maximum size of the descriptor ring. The lower seven bits of length are hard–wired to 0b. Byte addresses within the descriptor buffer are computed as follows:

*address = base + (ptr * 16)*, where *ptr* is the value in the hardware head or tail register.

The size chosen for the head and tail registers permit a maximum of 64 K descriptors, or approximately 16 K packets for the transmit queue given an average of four descriptors per packet.

Once activated, hardware fetches the descriptor indicated by the hardware head register. The hardware tail register points one beyond the last valid descriptor.

Software can determine if a packet has been sent by setting the RS bit (or the RPS bit for the **82544GC/EI** only) in the transmit descriptor command field. Checking the transmit descriptor DD bit in memory eliminates a potential race condition. All descriptor data is written to the IO bus prior to incrementing the head register, but a read of the head register could “pass” the data write in systems performing IO write buffering. Updates to transmit descriptors use the same IO write path and follow all data writes. Consequently, they are not subject to the race condition. Other potential conditions also prohibit software reading the head pointer.

In general, hardware prefetches packet data prior to transmission. Hardware typically updates the value of the head pointer after storing data in the transmit FIFO[1].

The process of checking for completed packets consists of one of the following:

- Scan memory for descriptor status write-backs.
- Take an interrupt. An interrupt condition can be generated whenever a transmit queue goes empty (ICR.TXQE). Interrupts can also be triggered in other ways.

3.4.1 Transmit Descriptor Fetching

The descriptor processing strategy for transmit descriptors is essentially the same as for receive descriptors except that a different set of thresholds are used. As for receives, the number of on-chip transmit descriptors buffer space is 64 descriptors.

When the on-chip buffer is empty, a fetch happens as soon as any descriptors are made available (software writes to the tail pointer). When the on-chip buffer is nearly empty (TXDCTL.PTHRESH), a prefetch is performed whenever enough valid descriptors (TXDCTL.HTHRESH) are available in host memory and no other DMA activity of greater priority is pending (descriptor fetches and write-backs or packet data transfers).

The descriptor prefetch policy is aggressive to maximize performance. If descriptors reside in an external cache, the system must ensure cache coherency before changing the tail pointer.

When the number of descriptors in host memory is greater than the available on-chip descriptor storage, the chip may elect to perform a fetch which is not a multiple of cache line size. The hardware performs this non-aligned fetch if doing so results in the next descriptor fetch being aligned on a cache line boundary. This allows the descriptor fetch mechanism to be most efficient in the cases where it has fallen behind software.

3.4.2 Transmit Descriptor Write-back

The descriptor write-back policy for transmit descriptors is similar to that for receive descriptors with a few additional factors. First, since transmit descriptor write-backs are optional (controlled by RS[2] in the transmit descriptor), only descriptors which have one (or both) of these bits set starts the accumulation of write-back descriptors. Secondly, to preserve backward compatibility with the 82542, if the TXDCTL.WTHRESH value is 0b, the Ethernet controller writes back a single byte of the descriptor (TDESCR.STA) and all other bytes of the descriptor are left unchanged.

1. With the RPS bit set, the head is not advanced until after the packet is transmitted or rejected due to excess collisions **(82544GC/EI** only).
2. And RPS for the **82544GC/EI** only.

Since the benefit of delaying and then bursting transmit descriptor write-backs is small at best, it is likely that the threshold are left at the default value (0b) to force immediate write-back of transmit descriptors and to preserve backward compatibility.

Descriptors are written back in one of three conditions:

- TXDCTL.WTHRESH = 0b and a descriptor which has RS[1] set is ready to be written back
- Transmit Interrupt Delay timer expires
- TXDCTL.WTHRESH > 0b and TXDCTL.WTHRESH descriptors have accumulated

For the first condition, write-backs are immediate. This is the default operation and is backward compatible. For this case, the Transmit Interrupt delay function works as described in Section 3.4.3.1.

The other two conditions are only valid if descriptor bursting is enabled (see Section 13.4.44). In the second condition, the Transmit Interrupt Delay timer (TIDV) is used to force timely write–back of descriptors. The first packet after timer initialization starts the timer. Timer expiration flushes any accumulated descriptors and sets an interrupt event (TXDW).

For the final condition, if TXDCTL.WTHRESH descriptors are ready for write-back, the write-back is performed.

3.4.3 Transmit Interrupts

Hardware supplies three transmit interrupts. These interrupts are initiated through the following conditions:

- Transmit queue empty (TXQE) — All descriptors have been processed. The head pointer is equal to the tail pointer.
- Descriptor done [Transmit Descriptor Write-back (TXDW)] — Set when hardware writes back a descriptor with RS[1] set. This is only expected to be used in cases where, for example, the streams interface has run out of descriptors and wants to be interrupted whenever progress is made.
- Transmit Delayed Interrupt (TXDW) — In conjunction with IDE (Interrupt Delay Enable), the TXDW indication is delayed by a specific time per the TIDV register. This interrupt is set when the transmit interrupt countdown register expires. The countdown register is loaded with the value of the IDV field of the TIDV register, when a transmit descriptor with its RS[1] bit and the IDE bit are set, is written back. When a Transmit Delayed Interrupt occurs, the TXDW interrupt cause bit is set (just as when a Transmit Descriptor Write-back interrupt occurs). This interrupt may be masked in the same manner as the TXDW interrupt. This interrupt is used frequently by software that performs dynamic transmit chaining, by adding packets one at a time to the transmit chain.

Note: The transmit delay interrupt is indicated with the same interrupt bit as the transmit write-back interrupt, TXDW. The transmit delay interrupt is only delayed in time as discussed above.

1. Or RPS for the **82544GC/EI** only.

- Link status change (LSC) - Set when the link status changes. When using the internal PHY, link status changes are determined and indicated by the PHY via a change in its LINK indication.

  When using an external TBI device (**82544GC/EI** only), the device might indicate a link status change using its LOS (loss of sync) indication. In this TBI mode, if HW Auto-Negotiation is enabled, the MAC can also detect and signal a link status change if the Configuration Base Page register is received (0b), or if either the LRST or ANE bits are changed by software.

- Transmit Descriptor Ring Low Threshold Hit (TXD_LOW) (not applicable to the **82544GC/EI**) - Set when the total number of transmit descriptors available (as measured by the difference between the Tx descriptor ring Head and Tail pointer) hits the low threshold specified in the TXDCTL.LWTHRESH field.

3.4.3.1 Delayed Transmit Interrupts

This mechanism allows software the flexibility of delaying transmit interrupts until no more descriptors are added to a transmit chain for a certain amount of time, rather than when the Ethernet controller's head pointer catches the tail pointer. This occurs if the Ethernet controller is processing packets slightly faster than the software, a likely scenario for gigabit operations.

A software driver usually has no knowledge of when it is going to be asked to send another frame. For performance reasons, it is best to generate only one transmit interrupt after a burst of packets have been sent.

Refer to Section 3.3.3.1 for specific details.

3.5 TCP Segmentation

Hardware TCP Segmentation is one of the off-loading options of most modern TCP/IP stacks. This is often referred to as "Large Send" offloading. This feature enables the TCP/IP stack to pass to the Ethernet controller software driver a message to be transmitted that is bigger than the Maximum Transmission Unit (MTU) of the medium. It is then the responsibility of the software driver and hardware to carve the TCP message into MTU size frames that have appropriate layer 2 (Ethernet), 3 (IP), and 4 (TCP) headers. These headers must include sequence number, checksum fields, options and flag values as required. Note that some of these values (such as the checksum values) are unique for each packet of the TCP message, and other fields such as the source IP address is constant for all packets associated with the TCP message.

The offloading of these processes from the software driver to the Ethernet controller saves significant CPU cycles. The software driver shares the additional tasks to support these options with the Ethernet controller.

Although the Ethernet controller's TCP segmentation offload implementation was specifically designed to take advantage of new "TCP Segmentation offload" features, the hardware implementation was made generic enough so that it could also be used to "segment" traffic from other protocols. For instance this feature could be used any time it is desirable for hardware to segment a large block of data for transmission into multiple packets that contain the same generic header.

3.5.1 Assumptions

The following assumption applies to the TCP Segmentation implementation in the Ethernet controller:

- The RS bit operation is not changed. Interrupts are set after data in buffers pointed to by individual descriptors is transferred to hardware.
- Checksums are not accurate above a 12 K frame size.
- The function of the RPS[1] bit in the Transmit Descriptor is applicable to all of the packets that make up the "TCP Segmentation" context, not the individual packets segmented by hardware.

3.5.2 Transmission Process

The transmission process for regular (non-TCP Segmentation packets) involves:

- The protocol stack receives from an application a block of data that is to be transmitted.
- The protocol stack calculates the number of packets required to transmit this block based on the MTU size of the media and required packet headers.
- For each packet of the data block:
- Ethernet, IP and TCP/UDP headers are prepared by the stack.
- The stack interfaces with the software device driver and commands the driver to send the individual packet.
- The driver gets the frame and interfaces with the hardware.
- The hardware reads the packet from host memory (via DMA transfers).
- The driver returns ownership of the packet to the operating system when the hardware has completed the DMA transfer of the frame (indicated by an interrupt).

The transmission process for the Ethernet controller TCP segmentation offload implementation involves:

- The protocol stack receives from an application a block of data that is to be transmitted.
- The stack interfaces to the software device driver and passes the block down with the appropriate header information.
- The software device driver sets up the interface to the hardware (via descriptors) for the TCP Segmentation context.
- The hardware transfers the packet data and performs the Ethernet packet segmentation and transmission based on offset and payload length parameters in the TCP/IP context descriptor including:
  — Packet encapsulation
  — Header generation & field updates including IP and TCP/UDP checksum generation
  — The driver returns ownership of the block of data to the operating system when the hardware has completed the DMA transfer of the entire data block (indicated by an interrupt).

1. **82544GC/EI** only.

3.5.2.1 TCP Segmentation Data Fetch Control

To perform TCP Segmentation in the Ethernet controller, the DMA unit must ensure that the entire payload of the segmented packet fits into the available space in the on-chip Packet Buffer. The segmentation process is performed without interruption. The DMA performs various comparisons between the payload and the Packet Buffer to ensure that no interruptions occur. The TCP Segmentation Pad & Minimum Threshold (TSPMT) register is used to allow software to program the minimum threshold required for a TCP Segmentation payload. Consideration should be made for the MTU value when writing this field. The TSPMT register is also used to program the threshold padding overhead. This padding is necessary due to the indeterminate nature of the MTU and the associated headers.

3.5.3 TCP Segmentation Performance

Performance improvements for a hardware implementation of TCP Segmentation offload mean:

- The operating system stack does not need to partition the block to fit the MTU size, saving CPU cycles.
- The operating system stack only computes one Ethernet, IP, and TCP header per segment, saving CPU cycles.
- The operating system stack interfaces with the software device driver only once per block transfer, instead of once per frame.
- Larger PCI bursts are used which improves bus efficiency.
- Interrupts are easily reduced to one per TCP message instead of one per packet.
- Fewer I/O accesses are required to command the hardware.

3.5.4 Packet Format

Typical TCP/IP transmit window size is 8760 bytes (about 6 full size frames). A TCP message can be as large as 64 KB and is generally fragmented across multiple pages in host memory. The Ethernet controller partitions the data packet into standard Ethernet frames prior to transmission. The Ethernet controller supports calculating the Ethernet, IP, TCP, and even UDP headers, including checksum, on a frame by frame basis.

Ethernet	IPv4	TCP/UDP	DATA	FCS

Figure 3-5. TCP/IP Packet Format

Frame formats supported by the Ethernet controller's TCP segmentation include:

- Ethernet 802.3
- IEEE 802.1q VLAN (Ethernet 802.3ac)
- Ethernet Type 2
- Ethernet SNAP
- IPv4 headers with options
- IPv6 headers with IP option next headers
- IPv6 packet tunneled in IPv4

- TCP with options
- UDP with limitations.

UDP (unlike TCP) is not a “reliable protocol”, and fragmentation is not supported at the UDP level. UDP messages that are larger than the MTU size of the given network medium are normally fragmented at the IP layer. This is different from TCP, where large TCP messages can be fragmented at either the IP or TCP layers depending on the software implementation. The Ethernet controller has the ability to segment UDP traffic (in addition to TCP traffic). This process has limited usefulness.

IP tunneled packets are not supported for TCP Segmentation operation[1].

3.5.5 TCP Segmentation Indication

Software indicates a TCP Segmentation transmission context to the hardware by setting up a TCP/IP Context Transmit Descriptor. The purpose of this descriptor is to provide information to the hardware to be used during the TCP segmentation offload process. The layout of this descriptor is reproduced in Section 3.3.6.

	63 48	47 40	39 32	31 16	15 8	7 0
0	TUCSE	TUCSO	TUCSS	IPCSE	IPCSO	IPCSS

	63 48	47 40	39 36	35 32	31 24	23 20	19 0
8	MSS	HDRLEN	RSV	STA	TUCMD	DTYP	PAYLEN

7	6	5	4	3	2	1	0
IDE	RSV	DEXT	RSV	RS	TSE	IP	TCP

Figure 3-6. TCP/IP Context Transmit Descriptor & Command Layout

Setting the TSE bit in the Command field to 1b indicates that this descriptor refers to the TCP Segmentation context (as opposed to the normal checksum offloading context). This causes the checksum offloading, packet length, header length, and maximum segment size parameters to be loaded from the descriptor into the Ethernet controller.

The TCP Segmentation prototype header is taken from the packet data itself. Software must identity the type of packet that is being sent (IP/TCP, IP/UDP, other), calculate appropriate checksum offloading values for the desired checksums, and calculate the length of the header which is pre-pended. The header may be up to 240 bytes in length.

Once the TCP Segmentation context has been set, the next descriptor provides the initial data to transfer. This first descriptor(s) must point to a packet of the type indicated. Furthermore, the data it points to may need to be modified by software as it serves as the prototype header for all packets within the TCP Segmentation context. The following sections describe the supported packet types and the various updates which are performed by hardware. This should be used as a guide to determine what must be modified in the original packet header to make it a suitable prototype header.

The following summarizes the fields considered by the driver for modification in constructing the prototype header:

- IPv4 Header
  - — Length should be set to zero
  - — Identification Field should be set as appropriate for first packet of send (if not already)
  - — Header Checksum should be zeroed out unless some adjustment is needed by the driver
- IPv6 Header
  - — Length should be set to zero
- TCP Header
  - — Sequence Number should be set as appropriate for first packet of send (if not already)
  - — PSH, and FIN flags should be set as appropriate for <u>last</u> packet of send
  - — TCP Checksum should be set to the partial pseudo-header checksum as follows:

<table>
<tr><td colspan="3">IP Source Address</td></tr>
<tr><td colspan="3">IP Destination Address</td></tr>
<tr><td colspan="3">Zero</td></tr>
<tr><td colspan="2">Zero</td><td>Next Header</td></tr>
<tr><td>Zero[a]</td><td>Layer 4 Protocol[a]</td><td>Zero[a]</td></tr>
</table>

a. **82544GC/EI** only

Figure 3-7. TCP Partial Pseudo-Header Checksum

- UDP Header
  - — Checksum should be set as in TCP header, above

The Ethernet controller's DMA function fetches the ethernet, IP, and TCP/UDP prototype header information from the initial descriptor(s) and save them on-chip for individual packet header generation. The following sections describe the updating process performed by the hardware for each frame sent using the TCP Segmentation capability.

3.5.6 TCP Segmentation Use of Multiple Data Descriptors

TCP Segmentation enables a packet to be segmented to describe more than one data descriptor. A large packet contained in a single virtual-address buffer is better described as a series of data descriptors, each referencing a single physical address page.

The only requirement for this use is if multiple data descriptors for TCP segmentation follows this guideline:

- If multiple data descriptors are used to describe the IP/TCP/UDP header section, each descriptor must describe one or more complete headers; descriptors referencing only parts of headers are not supported.

Note: It is recommended that the entire header section, as described by the TCP Context Descriptor HDRLEN field, be coalesced into a single buffer and described using a single data descriptor.

3.5.7 IP and TCP/UDP Headers

This section outlines the format and content for the IP, TCP and UDP headers. The Ethernet controller requires baseline information from the software device driver in order to construct the appropriate header information during the segmentation process.

Header fields that are modified by the Ethernet controller are highlighted in the figures that follow.

The IPv4 header is first shown in the traditional (RFC 791) representation, and because byte and bit ordering is confusing in that representation, the IP header is also shown in little-endian format. The actual data is fetched from memory in little-endian format.

<table>
<tr><td colspan="5">1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td></tr>
<tr><td>Version</td><td>IP Hdr Length</td><td>TYPE of service</td><td colspan="2">Total length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td>Layer 4 Protocol ID</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
<tr><td colspan="5">Options</td></tr>
</table>

Figure 3-8. IPv4 Header (Traditional Representation)

<table>
<tr><th>Byte 3</th><th colspan="4">Byte 2</th><th>Byte 1</th><th colspan="2">Byte 0</th></tr>
<tr><td>7 6 5 4 3 2 1 0</td><td colspan="4">7 6 5 4 3 2 1 0</td><td>7 6 5 4 3 2 1 0</td><td colspan="2">7 6 5 4 3 2 1 0</td></tr>
<tr><td colspan="5">LSB Total length MSB</td><td>TYPE of service</td><td>Version</td><td>IP Hdr Length</td></tr>
<tr><td>Fragment Offset Low</td><td>RES</td><td>NF</td><td>MF</td><td>Fragment Offset High</td><td colspan="3">LSB Identification MSB</td></tr>
<tr><td colspan="5">Header Checksum</td><td>Layer 4 Protocol ID</td><td colspan="2">Time to Live</td></tr>
<tr><td colspan="8">Source Address</td></tr>
<tr><td colspan="8">Destination Address</td></tr>
<tr><td colspan="8">Options</td></tr>
</table>

Figure 3-9. IPv4 Header (Little-Endian Order)

Flags Field Definition:

The Flags field is defined below. Note that hardware does not evaluate or change these bits.

- MF More Fragments
- NF No Fragments
- Reserved

Note: The IPv6 header is first shown in the traditional (RFC 2460), big-endian representation. The actual data is fetched from memory in little-endian format.

<table>
<tr><th colspan="4">0 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 20 1 2 3 4 5 6 7 8 9 30 1</th></tr>
<tr><td>Version</td><td>Traffic Class</td><td colspan="2">Flow Label</td></tr>
<tr><td colspan="2">Payload Length</td><td>Next Header</td><td>Hop Limit</td></tr>
<tr><td colspan="4">Source Address</td></tr>
<tr><td colspan="4">Destination Address</td></tr>
</table>

Figure 3-10. IPv6 TCP Header (Traditional Representation)

A TCP or UDP frame uses a 16 bit wide one's complement checksum. The checksum word is computed on the outgoing TCP or UDP header and payload, and on the Pseudo Header. Details on checksum computations are provided in Section 3.5. TCP requires the use of checksum, where it is optional for UDP.

The TCP header is first shown in the traditional (RFC 793) representation. Because byte and bit ordering is confusing in that representation, the TCP header is also shown in little-endian format. The actual data is fetched from memory in little-endian format.

<table>
<tr><th colspan="9">1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th></tr>
<tr><td colspan="4">Source Port</td><td colspan="5">Destination Port</td></tr>
<tr><td colspan="9">Sequence Number</td></tr>
<tr><td colspan="9">Acknowledgement Number</td></tr>
<tr><td>TCP Header Length</td><td>Reserved</td><td>U R G</td><td>A C K</td><td>P S H</td><td>R S T</td><td>S Y N</td><td>F I N</td><td>Window</td></tr>
<tr><td colspan="4">Checksum</td><td colspan="5">Urgent Pointer</td></tr>
<tr><td colspan="9">Options</td></tr>
</table>

Figure 3-11. TCP Header (Traditional Representation)

<table>
<tr><th>Byte3</th><th>Byte2</th><th colspan="7">Byte1</th><th colspan="2">Byte0</th></tr>
<tr><td>7 6 5 4 3 2 1 0</td><td>7 6 5 4 3 2 1 0</td><td colspan="7">7 6 5 4 3 2 1 0</td><td colspan="2">7 6 5 4 3 2 1 0</td></tr>
<tr><td colspan="2">Destination Port</td><td colspan="9">Source Port</td></tr>
<tr><td colspan="11">LSB Sequence Number MSB</td></tr>
<tr><td colspan="11">Acknowledgement Number</td></tr>
<tr><td colspan="2">Window</td><td>RE S</td><td>U R G</td><td>A C K</td><td>P S H</td><td>R S T</td><td>S Y N</td><td>FI N</td><td>TCP Header Length</td><td>Reserved</td></tr>
<tr><td colspan="2">Urgent Pointer</td><td colspan="9">Checksum</td></tr>
<tr><td colspan="11">Options</td></tr>
</table>

Figure 3-12. TCP Header (Little-Endian)

The TCP header is always a multiple of 32 bit words. TCP options may occupy space at the end of the TCP header and are a multiple of 8 bits in length. All options are included in the checksum.

The checksum also covers a 96-bit pseudo header conceptually prefixed to the TCP Header (see Figure 3-13 and Figure 3-14). The IPv4 pseudo header contains the IPv4 Source Address, the IPv4 Destination Address, the IPv4 Protocol field, and TCP Length. The IPv6 pseudo header contains the IPv6 Source Address, the IPv6 Destination Address, the IPv6 Payload Length, and the IPv6 Next Header field. Software pre-calculates the partial pseudo header sum, which includes IPv4 SA, DA and protocol types, but not the TCP length, and stores this value into the TCP checksum field of the packet.

The Protocol ID field should always be added the least significant byte (LSB) of the 16 bit pseudo header sum, where the most significant byte (MSB) of the 16 bit sum is the byte that corresponds to the first checksum byte out on the wire.

The TCP Length field is the TCP Header Length including option fields plus the data length in bytes, which is calculated by hardware on a frame by frame basis. The TCP Length does not count the 12 bytes of the pseudo header. The TCP length of the packet is determined by hardware as:

TCP Length = Payload + HDRLEN - TUCSS

“Payload” is normally MSS except for the last packet where it represents the remainder of the payload.

0		31
IP Source Address		
IP Destination Address		
Zero	Layer 4 Protocol ID	TCP Length

Figure 3-13. TCP Pseudo Header Content (Traditional Representation)

IP Source Address	
IP Destination Address	
Upper Layer Packet Length	
Zero	Next Header

Figure 3-14. TCP PseudoHeader Content for IPv6

Note: The IP Destination address is the final destination of the packet. Therefore, if a routing header is used, the last address in the route list is used in this calculation. The upper-layer packet length is the length of the TCP header and the TCP payload.

The UDP header is always 8 bytes in size with no options.

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

Figure 3-15. UDP Header (Traditional Representation)

Byte3	Byte2	Byte1	Byte0
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Destination Port		Source Port	
Checksum		Length	

Figure 3-16. UDP Header (Little-Endian Order)

UDP pseudo header has the same format as the TCP pseudo header. The IPv4 pseudo header conceptually prefixed to the UDP header contains the IPv4 source address, the IPv4 destination address, the IPv4 protocol field, and the UDP length (same as the TCP Length discussed above). The IPv6 pseudo header for UDP is the same as the IPv6 pseudo header for TCP. This checksum procedure is the same as is used in TCP.

IP Source Address		
IP Destination Address		
Zero	Layer 4 Protocol ID	UDP Length

Figure 3-17. UDP Pseudo Header Diagram for IPv4

IP Source Address	
IP Destination Address	
Upper Layer Packet Length	
Zero	Next Header

Figure 3-18. UDP PseudoHeader Diagram for IPv6

Note: The IP Destination Address is the final destination of the packet. Therefore, if a routing header is used, the last address in the route list is used in this calculation. The upper-layer packet length is the length of the UDP header and UDP payload.

Unlike the TCP checksum, the UDP checksum is optional. Software must set the TXSM bit in the TCP/IP Context Transmit Descriptor to indicate that a UDP checksum should be inserted. Hardware does not overwrite the UDP checksum unless the TXSM bit is set.

3.5.8 Transmit Checksum Offloading with TCP Segmentation

The Ethernet controller supports checksum off-loading as a component of the TCP Segmentation offload feature and as a standalone capability. Section 3.5.8 describes the interface for controlling the checksum off-loading feature. This section describes the feature as it relates to TCP Segmentation.

The Ethernet controller supports IP and TCP/UDP header options in the checksum computation for packets that are derived from the TCP Segmentation feature. The Ethernet controller is capable of computing one level of IP header checksum and one TCP/UDP header and payload checksum. In case of multiple IP headers, the driver has to compute all but one IP header checksum. The Ethernet controller calculates checksums on the fly on a frame by frame basis and inserts the result in the IP/TCP/UDP headers of each frame. TCP and UDP checksum are a result of performing the checksum on all bytes of the payload and the pseudo header.

Three specific types of checksum are supported by the hardware in the context of the TCP Segmentation offload feature:

- IPv4 checksum (IPv6 does not have a checksum)
- TCP checksum
- UDP checksum

Each packet that is sent via the TCP segmentation offload feature optionally includes the IPv4 checksum and either the TCP or UDP checksum.

All checksum calculations use a 16-bit wide one's complement checksum. The checksum word is calculated on the outgoing data. The checksum field is written with the 16 bit one's complement of the one's complement sum of all 16-bit words in the range of CSS to CSE, including the checksum field itself.

3.5.9 IP/TCP/UDP Header Updating

IP/TCP/UDP header is updated for each outgoing frame based on the IP/TCP header prototype which hardware transfers from the first descriptor(s) and stores on chip. The IP/TCP/UDP headers are fetched from host memory into an on-chip 240 byte header buffer once for each TCP segmentation context (for performance reasons, this header is not fetched again for each additional packet that is derived from the TCP segmentation process). The checksum fields and other header information are later updated on a frame by frame basis. The updating process is performed concurrently with the packet data fetch.

The following sections define which fields are modified by hardware during the TCP Segmentation process by the Ethernet controller. Figure 3-19 illustrates the overall data flow.

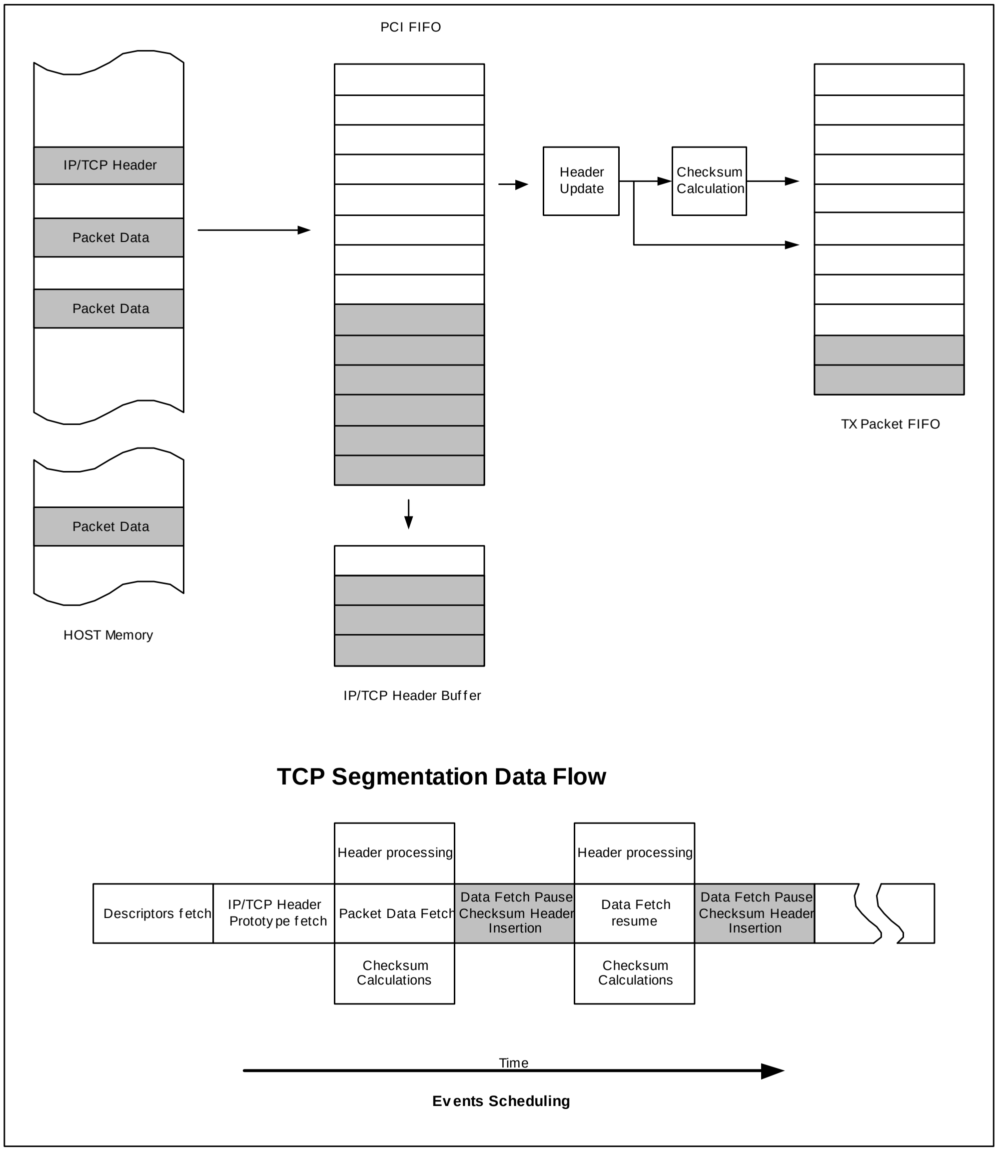


Figure 3-19. Overall Data Flow

3.5.9.1 TCP/IP/UDP Header for the First Frame

The hardware makes the following changes to the headers of the first packet that is derived from each TCP segmentation context.

- IPv4 Header
  - IP Total Length = MSS + HDRLEN – IPCSS
  - IP Checksum
  - IPv6 Header
  - Payload Length = MSS + HDRLEN - IPCSS
- TCP Header
  - Sequence Number: The value is the Sequence Number of the first TCP byte in this frame.
  - If FIN flag = 1b, it is cleared in the first frame.
  - If PSH flag =1b, it is cleared in the first frame.
  - TCP Checksum
- UDP Header
  - UDP length: MSS + HDRLEN - TUCSS
  - UDP Checksum

3.5.9.2 TCP/IP/UDP Header for the Subsequent Frames

The hardware makes the following changes to the headers for subsequent packets that are derived as part of a TCP segmentation context:

Note: Number of bytes left for transmission = PAYLEN – (N * MSS). Where N is the number of frames that have been transmitted.

- IPv4 Header
  - IP Identification: incremented from last value (wrap around)
  - IP Total Length = MSS + HDRLEN – IPCSS
  - IP Checksum
- IPv6 Header
- Payload Length = MSS + HRDLEN - IPCSS
- TCP Header
  - Sequence Number update: Add previous TCP payload size to the previous sequence number value. This is equivalent to adding the MSS to the previous sequence number.
  - If FIN flag = 1b, it is cleared in these frames.
  - If PSH flag =1b, it is cleared in these frames.
  - TCP Checksum
- UDP Header
  - UDP Length: MSS + HDRLEN – TUCSS
  - UDP Checksum

3.5.9.3 TCP/IP/UDP Header for the Last Frame

The controller makes the following changes to the headers for the last frame of a TCP segmentation context:

Note: Last frame payload bytes = PAYLEN – (N * MSS)

- IPv4 Header
  - IP Total Length = (last frame payload bytes + HDRLEN) – IPCSS
  - IP Identification: incremented from last value (wrap around)
  - IP Checksum
- IPv6 Header
- Payload Length = MSS + HDRLEN - IPCSS
- TCP Header
  - Sequence Number update: Add previous TCP payload size to the previous sequence number value. This is equivalent to adding the MSS to the previous sequence number.
  - If FIN flag = 1b, set it in this last frame
  - If PSH flag =1b, set it in this last frame
  - TCP Checksum
- UDP Header
  - UDP length: (last frame payload bytes + HDRLEN) - TUCSS
  - UDP Checksum

3.6 IP/TCP/UDP Transmit Checksum Offloading

The previous section on TCP Segmentation offload describes the IP/TCP/UDP checksum offloading mechanism used in conjunction with TCP Segmentation. The same underlying mechanism can also be applied as a standalone feature. The main difference in normal packet mode (non-TCP Segmentation) is that only the checksum fields in the IP/TCP/UDP headers need to be updated.

Before taking advantage of the Ethernet controller's enhanced checksum offload capability, a checksum context must be initialized. For the normal transmit checksum offload feature, this task is performed by providing the Ethernet controller with a TCP/IP Context Descriptor with TSE = 0b to denote a non-segmentation context. For additional details on contexts, refer to Section 3.3.5. Enabling the checksum offloading capability without first initializing the appropriate checksum context leads to unpredictable results. Once the checksum context has been set, that context, is used for all normal packet transmissions until a new context is loaded. Also, since checksum insertion is controlled on a per packet basis, there is no need to clear/reset the context.

The Ethernet controller is capable of performing two transmit checksum calculations. Typically, these would be used for TCP/IP and UDP/IP packet types, however, the mechanism is general enough to support other checksums as well. Each checksum operates independently and provides identical functionality. Only the IP checksum case is discussed as follows.

Three fields in the TCP/IP Context Descriptor set the context of the IP checksum offloading feature:

- IPCSS

  This field specifies the byte offset form the start of the transferred data to the first byte to be included in the checksum. Setting this value to 0b means that the first byte of the data is included in the checksum. The maximum value for this field is 255. This is adequate for typical applications.

Note: The IPCSS value needs to be less than the total DMA length to a packet. If this is not the case, the result will be unpredictable.

- IPCSO

  This field specifies where the resulting checksum should be placed. Again, this is limited to the first 256 bytes of the packet and must be less than or equal to the total length of a given packet. If this is not the case, the checksum is not inserted.

- IPCSE

  This field specifies where the checksum should stop. A 16-bit value supports checksum offloading of packets as large as 64KB. Setting the IPCSE field to all zeros means End-of-Packet. In this way, the length of the packet does not need to be calculated.

As mentioned above, it is not necessary to set a new context for each new packet. In many cases, the same checksum context can be used for a majority of the packet stream. In this case, some of the offload feature only for a particular traffic type, thereby avoiding all context descriptors except for the initial one.

Note: This page intentionally left blank.

PCI Local Bus Interface 4

The PCI/PCI-X Family of Gigabit Ethernet Controllers are PCI 2.2 or 2.3 compliant devices and implement the PCI-X Addendum to the PCI Local Bus Specification, Revision 1.0.

Note: The **82540EP/EM**, **82541xx**, and **82547GI/EI** do not support PCI-X mode.

4.1 PCI Configuration

The PCI Specification requires implementation of PCI Configuration registers. After a system reset, these registers are initially configured by the BIOS, and/or a "Plug and Play" aware Operating System (OS). Device drivers read these registers to determine what resources (interrupt number, memory mapping location, etc.) the BIOS and/or OS assigned to the Ethernet controller.

The **82547GI/EI** uses a dedicated CSA port for its system bus connection. Logically, it still follows PCI configuration. However, some configuration parameters, such as cache line, are irrelevant. Additionally, the **82547GI/EI** requires special interrupt configuration in the BIOS (see Section 4.5).

Note: The **82547GI/EI** does not support 64-bit addressing.

Four different regions of the PCI configuration space are used.

Address	Item	Description
00h-3Ch	PCI	Section 2.3.1
DCh-E0h	PCI Power Management	Section 6.3.3
E4h-E8h	PCI-X	Section 4.1.1
F0h-FCh	Message Signaled Interrupt[a]	Section 4.1.3.1

a. Not applicable to the **82541xx** and **82547GI/EI**.

These spaces are linked into a linked list using the Capabilities Pointer field (Cap_Ptr) in the PCI Configuration section.

The implementation of the PCI registers for the PCI/PCI-X Family of Gigabit Ethernet Controllers are listed in Table 4-1:

Table 4-1. Mandatory PCI Registers

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0h	Device ID		Vendor ID	
4h	Status Register		Command Register	
8h	Class Code (020000h)			Revision ID
Ch	BIST (00h)	Header Type (00h)	Latency Timer	Cache Line Size
10h	Base Address 0[a]			
4h	Base Address 1			
18h	Base Address 2			

1Ch	Base Address 3 (unused)			
20h	Base Address 4 (unused)			
2h4	Base Address 5 (unused)			
28h	Cardbus CIS Pointer (not used)			
2Ch	Subsystem ID		Subsystem Vendor ID	
30h	Expansion ROM Base Address			
34h	Reserved			Cap_Ptr
38h	Reserved			
3Ch	Max_Latency (00h)	Min_Grant (FFh)	Interrupt Pin (01h)	Interrupt Line

a. Refer to Table 4-2.

The following list provides explanations of the various PCI registers and their bit fields:

Vendor ID — This uniquely identifies all Intel PCI products. This field may be auto-loaded from the EEPROM at power on or upon the assertion of PCI_RST#. A value of 8086h is the default for this field upon power up if the EEPROM does not respond or is not programmed.

Device ID — This uniquely identifies the Ethernet controller. This field may be autoloaded from the EEPROM at power on or upon the assertion of RST#. The default value for this field is used upon power up if the EEPROM does not respond or is not programmed.

Command Reg. — The layout is listed in Table 4-3. Shaded bits are not used by this implementation and are hard wired to 0b.

Status Register — The layout is listed in Table 4-4. Shaded bits are not used by this implementation and are hard wired to 0b.

Revision — Sequential stepping number starting with 00h for the A0 revision of the Ethernet controller. Refer to the *PCI/PCI-X Family of Gigabit Ethernet Controllers Specification Update* for the latest stepping information.

Class Code — The class code, 020000h identifies the Ethernet controller as an Ethernet adapter.

Cache Line Size[1] Used to store the cache line size. The value is in units of 4 bytes. A system with a cache line size of 64 bytes sets the value of this register to 10h. The only sizes that are supported are 16, 32, 64, and 128 bytes. All other sizes are treated as 0b. See the information about exceptions in Section 4.4.

Unsupported values affect PCI cache line support. All writes default to using the memory write (MW) command, and memory read command determination uses a cache line size of 32 bytes.

Latency Timer The lower two bits are not implemented and return 0b. The upper six bits are Read/Write.

Header Type This is for a normal single function Ethernet controller and reads 00h.

BIST Built in Self-test is not implemented as supportable from PCI configuration space in this version of the Ethernet controller.

Base Address Registers

The Base Address Registers (or BARs) are used to map the Ethernet controller's register space and flash to system memory space. In PCI-X mode or in PCI mode when the BAR32 bit of the EEPROM is 0b, two registers are used for each of the register space and the flash memory in order to map 64-bit addresses. In PCI mode, if the BAR32 bit in the EEPROM is 1b, one register is used for each to map 32-bit addresses.

64-bit BARs PCI-X mode with BAR32 bit in the EEPROM set to 0b.

Table 4-2. Base Address Registers

BAR	Addr.	31 4	3	2 1	0
0	10h	***Memory*** Register Base Address (bits 31:4)	pref.	type	mem
1	14h	***Memory*** Register Base Address (bits 63:32)			
2	18h	***Memory*** Flash Base Address (bits 31:4)	pref.	type	mem
3	1Ch	***Memory*** Flash Base Address (bits 63:32)			
4	20h	IO Register Base Address (bits 31:2)		0b	mem
5	24h	Reserved (read as all 0b's)			

32-bit BARs Conventional PCI mode with BAR32 bit in the EEPROM set to 1b

BAR	Addr.	31 4	3	2 1	0
0	10h	***Memory*** Register Base Address	pref.	type	mem
1	14h	***Memory*** Flash Base Address	pref.	type	mem
2	18h	IO Register Base Address (bits 31:2)		0b	mem
3	1Ch	Reserved (read as all 0b's)			
4	20h	Reserved (read as all 0b's)			
5	24h	Reserved (read as all 0b's)			

1. Not applicable to the **82547GI/EI**.

All base address registers have the following fields:

Field	Bit(s)	Read/ Write	Initial Value	Description
Mem	0	R	0b ***for mem*** 1b for I/O	0b indicates memory space. 1b indicates I/O.
Type	2:1	R	00b for 32-bit 10b for 64-bit	Indicates the address space size. 00b = 32-bit 10b = 64-bit
Prefetch	3	R	0b	0b = non-prefetchable space 1b = prefetchable space Ethernet controller implements non-prefetchable space since it has read side-effects.
Address	31:0	R/W	0b	The lower bits of the address are hard-wired to 0b. The upper bits can be written by the system software to set the base address of the register or flash address space. The ***memory*** register space is 128K bytes. ***The Memory Register BAR has:*** • Bits 16:4 are hard-wired to 0b. • Bits 63:17 or 31:17 are read/write. The size of the flash space can very between 64 KB and 512 KB depending on the FLASH size read from the EEPROM. ***The Memory Flash BAR has these characteristics:*** Flash Size Valid Bits Zero Bits (R/W) (RO) • 64 KB 63/31:16 15:4 • 128 KB 63/31:17 16:4 • 256 KB 63/31:18 17:4 • 512 KB 63/31:19 18:4 The size of the IO register space is 8 bytes. The I/O Register BAR has: • Bit 2 hard-wired to 0b • Bits 31:3 as read/write

Expansion ROM Base Address

This register is used to define the address and size information for boot-time access to the optional Flash memory.

31 ... 11	10 ... 1	0
Expansion Rom Base Address	Reserved	En

Field	Bit(s)	Read/ Write	Initial Value	Description
En	0	R/W	0b	1b = Enables expansion ROM access. 0b = Disables expansion ROM access.
Reserved	10:1	R	0b	Always read as 0b. Writes are ignored.
Address	31:11	R/W	0b	The lower bits of the address are hard-wired to 0b. The upper bits can be written by the system software to set the base address of the register or flash address space. Since the flash is used as the expansion ROM, the size of the expansion ROM can very between 64 KB and 512 KB, depending on the FLASH size read from the EEPROM. Flash Size Valid Bits Zero Bits: • 64 KB 63/31:16 15:11 • 128 KB 63/31:17 16:11 • 256 KB 63/31:18 17:11 • 512 KB 63/31:19 18:11

CardBus CIS Pointer (82541PI/GI/EI and 82540EP Only)

When the Enable CLK_RUN# bit of the EEPROM's Initialization Control Word 2 and the 64/32 BAR bit of the EEPROM Initialization Control Word 1 (indicating a 32-bit BAR) are both set to 1b, the Cardbus CIS Pointer contains a value of 00000022h. Otherwise, it contains a value of 00000000h.

31 ... 3	2 ... 0
Offset	Space

Field	Bit(s)	Read/ Write	Initial Value	Description
Space	2:0	R/W	0 or 2	Indicates the address space where the CIS is located. 0 = Configuration Space 1 = BAR0 2 = BAR1 3 = BAR2 4 = BAR3 5 = BAR4 6 = BAR5 7 = Expansion ROM
Offset	31:3	R	0 or 4	Offset within the specified address space, multiplied by eight. When enabled, the value indicates that the CIS (Card Information Structure) is at an offset of 4*8, or 32 bytes into the Flash memory.

Subsystem ID This value can be loaded automatically from the EEPROM upon power-up or PCI reset. A value of 1008h is the default for this field upon power-up if the EEPROM does not respond or is not programmed.

Subsystem Vendor ID

This value can be loaded automatically from the EEPROM upon power-up or PCI reset. A value of 8086h is the default for this field upon power-up if the EEPROM does not respond or is not programmed.

Cap_Ptr The Capabilities Pointer field (Cap_Ptr) is an 8-bit field that provides an offset in the Ethernet controller's PCI Configuration Space for the location of the first item in the Capabilities Linked List. The Ethernet controller sets this bit and then implements a capabilities list to indicate that it supports PCI Power Management, PCI-X, and Message Signaled Interrupts[1]. Its value is DCh which is the address of the first entry: ACPI[2] Power Management.

Address	Item	Next Pointer
DCh-E0h	ACPI Power Management	E4h
E4h-E8h	PCI-X	F0h
F0h-FCh	Message Signaled Interrupt	00h

Figure 4-1. Capabilities Linked List

In conventional PCI mode, Message Signaled interrupts can be disabled in the EEPROM. If disabled, the message signaled interrupts won't appear on the linked list and PCI-X's "Next Pointer" is 0b.

1. Not applicable to the **82541xx** or **82547GI/EI**.
2. Not applicable to the **82541ER**.

Max_Lat/Min_Gnt[1]

The Ethernet controller places a very high load on the PCI bus during peak transmit and receive traffic. In full duplex mode, it has a peak throughput demand of 250 MB/sec. The peak delivered bandwidth on a 64-bit PCI bus at 33 MHz is 264 MB/sec, so the bus is fully saturated when transmit and receive are operating simultaneously. In half duplex operation, the Ethernet controller has a peak throughput demand of 125 MB/sec, which still puts an enormous load on the PCI bus. Consequently, the Max_Lat should be small and is set to 00h, and Min_Gnt is set to FFh indicating that the Ethernet controller requires a very high priority and time slice.

Interrupt Pin

Read only register indicating which interrupt line (INTA# vs. INTB#) the **82546GB/EB** uses. A value of 1b indicates that the **82546GB/EB** uses INTA# (as with all single-port Ethernet controllers). A value of 10b indicates that the **82546GB/EB** uses INTB#.

For each separate device/function within the Ethernet controller, the value reported here is based on the EEPROM Initialization Control Word 3 associated with this controller, as well as whether both device/functions are enabled. Provided both functions are enabled, then the value reported for each specific function is based on the *Interrupt Pin* field of each Ethernet controller's Initialization Control Word 3.

If only a single internal device/function is enabled, then the value reported here is 1b regardless of EEPROM configuration.

Interrupt Line

Read write register programmed by software to indicate which of the system interrupt request lines this Ethernet controller's interrupt pin is bound to. See the PCI definition for more details.

Table 4-3. Command Register Layout

15 … 10	9 … 0
Reserved	Command Bits

Bit(s)	Initial Value	Description
0	0b	I/O Access Enable.
1	0b	Memory Access Enable.
2	0b	Enable Mastering. Ethernet controller in PCI-X mode is permitted to initiate a split completion transaction regardless of the state of this bit.
3	0b	Special Cycle Monitoring.

1. This bit is a don't care for the **82547GI/EI**.

Bit(s)	Initial Value	Description
4	0b	Memory Write and Invalidate Enable (not applicable to the **82547GI/EI**).
5	0b	Palette Snoop Enable.
6	0b	Parity Error Response (not applicable to the **82547GI/EI**).
7	0b	Wait Cycle Enable.
8	0b	SERR# Enable (not applicable to the **82547GI/EI**).
9	0b	Fast Back-to-Back Enable.
10[a]	0b	Interrupt Disable (INTA# or CSA signaled).
15:10 15:11[a]	0b	Reserved.

a. **82541xx** and **82547GI/EI** only.

Table 4-4. Status Register Layout

15 ... 4	3 ... 0
Status Bits	Reserved

Bit(s)	Initial Value	Description
3:0 2:0[a]	0b	Reserved.
3[a]	0b	Interrupt Status. This bit is 1b when the Ethernet controller is generating an interrupt internally. When Interrupt Disable in the Command Register is also cleared, the Ethernet controller asserts INTA# or signal an interrupt over CSA.
4	1b	New Capabilities: Indicates that an Ethernet controller implements Extended Capabilities. The Ethernet controller sets this bit and implements a capabilities list to indicate that it supports PCI Power Management, PCI-X Bus, and message signaled interrupts.
5	1b	66 MHz Capable (don’t care for the **82547GI/EI**).
6	0b	UDF Supported. Hardwired to 0b for PCI 2.3[a].
7	0b	Fast Back-to-Back CapableThis bit must be cleared to 0b in PCI-X mode (not applicable to the **82547GI/EI**).
8	0b	Data Parity Reported.
10:9	01b	DEVSEL Timing (indicates medium device). Not applicable to the **82547GI/EI**.
11	0b	Signaled Target Abort.

Bit(s)	Initial Value	Description
12	0b	Received Target Abort.
13	0b	Received Master Abort.
14	0b	Signaled System Error (not applicable to the **82547GI/EI**).
15	0b	Detected Parity Error (not applicable to the **82547GI/EI**).

a. **82541xx** and **82547GI/EI** only.

4.1.1 PCI-X Configuration Registers

The Ethernet controller supports additional configuration registers that are specific to PCI-X. These registers are visible in conventional PCI and PCI-X modes, although they only affect the operation of PCI-X mode. The PCI-X registers are linked into the Capabilities linked list.

Note: The **82540EP/EM**, **82541xx**, and **82547GI/EI** do not support PCI-X mode.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
E4h	PCI-X Command		Next Capability	PCI-X Capability ID
E8h	PCI-X Status			

Figure 4-2. PCI-X Capability Registers

4.1.1.1 PCI-X Capability ID

Bits	Read/ Write	Initial Value	Description
7:0	R	7	**Capability ID** - Identifies the PCI-X register set in the capabilities linked list.

4.1.1.2 Next Capability

Bits	Read/ Write	Initial Value	Description
7:0	R	F0[a]	**Next Capability** – points to the next capability in the capabilities linked list.

a. In conventional PCI mode, Message Signaled Interrupts can also be disabled in the EEPROM. If disabled, the Message Signaled Interrupt registers are not visible, and PCI-X's "Next Capability" pointer is 0b.

4.1.1.3 PCI-X Command

15 ... 7	6 ... 4	3 ... 2	1	0
Reserved	Max. Split Transactions	Read Count	RO	DP

Bits	Read Write	Initial Value	Description
0	RW	0b	**Data Parity Error Recovery Enable.** If this bit is 1b, the Ethernet controller attempts to recover from Parity errors. If this bit is 0b, the Ethernet controller asserts SERR# (if enabled) whenever the Master Data Parity Error bit (Status Register, bit 8) is set.
1	RW	1b	**Enable Relaxed Ordering.** If this bit is set, the Ethernet controller sets the Relaxed Ordering attribute bit in some transactions.
3:2	RW	0b	**Maximum Memory Read Byte Count.** This register sets the maximum byte count the Ethernet controller uses for a Memory Read Sequence. The allowable values are: Register — Maximum Byte Count 0 — 512 1 — 1024 2 — 2048 3 — 4096
6:4	RW	0b	**Maximum Outstanding Split Transactions.** This register sets the maximum number of outstanding split transactions that the Ethernet controller uses. The Ethernet controller is only allowed to have one outstanding split transaction at any time. Register — Maximum Outstanding Transactions 0 — 1 1 — 2 2 — 3 3 — 4 4 — 8 5 — 12 6 — 16 7 — 32
15:7	R	0b	Reserved. Reads as 0b

4.1.1.4 PCI-X Status

31 29	28 26	25 23	22 21	20	19	18	17	16	15 8	7 3	2 0
Res.	Read Size	Max. Split	Rd Byte	Cplx	USC	SCD	133	64b	Bus Number	Device Number	Func. Num.

Bits	Read/ Write	Intial Value	Description
2:0	R	0b	**Function Number.** This number forms part of the Requester and Completer IDs for PCI-X transactions.
7:3	R	1Fh	**Device Number.** The system assigns a device number (other than 0b) to the Ethernet controller. It forms part of the Requester and Completer IDs for PCI-X transactions. The Ethernet controller updates this register with the contents of AD[15:11] on any Type 0 Configuration Write cycle.
15:8	R	FFh	**Bus Number.** This indicates the bus the Ethernet controller is placed on. It forms part of the Requester and Completer IDs for PCI-X transactions. The Ethernet controller updates this register with the contents of AD[7:0] on any Type 0 Configuration Write cycle.
16	R	1b[a]	**64-bit Device.** This indicates the Ethernet controller is a 64-bit device. It does not indicate the current bus width. It is loaded from the EEPROM Initialization Control Word 2 (see Section 5.6.12).
17	R	1b[a]	**133 MHz Capable.** A 1b indicates that the Ethernet controller is capable of operating at 133 MHz in PCI-X mode. A 0b indicates 66 MHz capability. This bit is loaded from the EEPROM Initialization Control Word 2 (see Section 5.6.12).
18	read, write 1b to clear	0b	**Split Completion Discarded.** (Write 1b to clear) This bit is set if the Ethernet controller discards a Split Completion because the requester would not accept it.
19	read, write 1b to clear	0b	**Unexpected Split Completion.** (Write 1b to clear) This bit indicates whether the Ethernet controller received an unexpected Split Completion with its requestor ID.
20	R	0b	**Device Complexity.** A 0b indicates the Ethernet controller is a simple device. A 1b indicates that the Ethernet controller is a bridge.
22:21	R	2b[a]	**Designed Maximum Memory Read Byte Count.** Indicates the maximum memory read byte count the Ethernet controller is designed to generate. Register — Maximum Byte Count 0 — 512 1 — 1024 2 — 2048 3 — 4096 The value of this register depends on the *Max_Read* bit in the EEPROM's *Initialization Control Word 2* (see Section 5.6.12). • *Max_Read* = 0b then value = 2 (2 KB) • *Max_Read* = 1b then value = 3 (4 KB)

<table>
<tr><th>Bits</th><th>Read/ Write</th><th>Intial Value</th><th>Description</th></tr>
<tr><td>25:23</td><td>R</td><td>0b</td><td>Designed Maximum Outstanding Split Transactions. A 0b indicates that the Ethernet controller is designed to have at the most one outstanding transaction.
<u>Register</u> — <u>Maximum Outstanding Transactions</u>
0 — 1
1 — 2
2 — 3
3 — 4
4 — 8
5 — 12
6 — 16
7 — 32</td></tr>
<tr><td>28:26</td><td>R</td><td>0b[a] (see Description)</td><td>Designed Maximum Cumulative Read Size. Indicates a number that is greater or equal maximum cumulative outstanding bytes to be read at one time.
<u>Register</u> — <u>Maximum Outstanding Bytes</u>
0 — 1 KB
1 — 2 KB
2 — 4 KB
3 — 8 KB
4 — 16 KB
5 — 32 KB
6 — 64 KB
7 — 128 KB
The value of this register depends on the DMCR_Map and Max_Read bits in the EEPROM’s Initialization Control Word 2 (see Section 5.6.12).
• DMCR_Map = 0b:
The value of this register reflects the number of bytes programmed in the Maximum Memory Read Byte Count (MMRBC) field of the PCI-X Command Register as follows:
• MMRBC = 0 (512) - DMCRS = 0 (1KB)
• MMRBC = 1 (1K) - DMCRS = 0 (1KB)
• MMRBC = 2 (2K) - DMCRS = 1 (2KB)
• MMRBC = 3 (4K) - DMCRS = 2 (4KB)
• DMCR_Map = 1b and Max_Read = 0b: DMCRS = 1 (2KB)
• DMCR_Map = 1b and Max_Read = 1b: DMCRS = 2 (4KB)</td></tr>
<tr><td>29</td><td>Read, write 1b to clear</td><td>0b</td><td>Received Split Completion Error Message. This bit is set if the Ethernet controller receives a Split Completion Message with the Split Completion Error attribute bit set.</td></tr>
<tr><td>31:30</td><td>R</td><td>0b</td><td>Reserved. Reads as 0b</td></tr>
</table>

a. Loaded from EEPROM.

4.1.2 Reserved and Undefined Addresses

Any PCI or PCI-X register address space not explicitly declared in this specification should be considered to be reserved, and should not be written. Writing to reserved or undefined configuration register addresses can cause indeterminate behavior. Reads from reserved or undefined configuration register addresses can return indeterminate values.

4.1.3 Message Signaled Interrupts[1]

Message Signaled Interrupt (MSI) capability is optional for PCI 2.2 or 2.3, but required for PCI-X. When Message Signaled Interrupts are enabled, instead of asserting an interrupt pin, the Ethernet controller generates an interrupt using a memory write command. The address and most of the data of the command are determined by the system and programmed in configuration registers. This permits the system to program a different message for each function so it can speed up interrupt delivery.

To enable Message Signaled Interrupts, the system software writes to the "MSI Enable" bit in the MSI "Message Control" register. When Message Signaled Interrupts are enabled, the Ethernet controller no longer asserts its INTA# pin to signal interrupts.

MSI systems allow a function to request up to 32 messages, but does not guarantee that all of them are allocated. The Ethernet controller supports only a single message. When Message Signaled Interrupts are enabled, the Ethernet controller generates a message when any of the unmasked bits in the Interrupt Cause Read register (ICR) are set to 1b. The Ethernet controller does not generate the message again until the ICR is read and a subsequent interrupt event occurs.

In conventional PCI mode, Message Signaled Interrupts can also be disabled in the EEPROM. If MSI is disabled, the Message Signaled Interrupt registers is not visible.

4.1.3.1 Message Signaled Interrupt Configuration Registers

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
F0h	Message Control		Next Capability	MSI Capability ID
F4h	Message Address			
F8h	Message Upper Address			
FCh	Reserved		Message Data	

Figure 4-3. Message Signaled Interrupt Configuration Registers

4.1.3.1.1 MSI Capability ID

Bits	Read/ Write	Initial Value	Description
7:0	R	05h	**Capability ID** - Identifies the Message Signaled Interrupt register set in the capabilities linked list.

4.1.3.1.2 Next Capability

Bits	Read/ Write	Initial Value	Description
7:0	R	00h	**Next Capability** – points to the next capability in the capabilities linked list. Its value is 0b since the Message Signaled Interrupt is the last item in the list.

1. Not applicable to the **82541xx** or **82547GI/EI**.

4.1.3.1.3 Message Control

15 — 8	7	6 — 4	3 — 1	0
Reserved	64b	Multiple Enable	Multiple Capable	En

Bits	Read/ Write	Initial Value	Description
0	R	0b	**MSI Enable.** If 1b, Message Signaled Interrupts[a] are enabled and the Ethernet controller generates Message Signaled Interrupts instead of asserting INTA#.
3:1	R	0b	**Multiple Message Capable.** Indicates the number of messages requested. The Ethernet controller only requests one message. Register — Number of messages 0 — 1 1 — 2 2 — 4 3 — 8 4 — 16 5 — 32 6 — Reserved 7 — Reserved
6:4	RW	0b	**Multiple Message Enable.** Written by the system to indicate the number of messages allocated. Since the Ethernet controller only supports one message, the system should never write a value other than 0b.
7	R	1b	**64-bit capable.** A value of 1b indicates that the Ethernet controller is capable of generating 64-bit message addresses.
15:8	R	0b	Reserved. Reads as 0b.

a. Not applicable to the **82541xx** or **82547GI/EI**.

4.1.3.1.4 Message Address

Bits	Read/ Write	Initial Value	Description
31:0	RW	0b	**Message Address** – Written by the system to indicate the lower 32-bits of the address to use for the MSI memory write transaction. The lower two bits are always written as 0b.

4.1.3.1.5 Message Upper Address

Bits	Read/ Write	Initial Value	Description
31:0	RW	0b	**Message Upper Address** – Written by the system to indicate the upper 32-bits of the address to use for the MSI memory write transaction.

4.1.3.1.6 Message Data

Bits	Read/ Write	Initial Value	Description
15:0	RW	0b	**Message Data** – Written by the system to indicate the lower 16 bits of the data written in the MSI memory write DWORD transaction. The upper 16 bits of the transaction are written as 0b.

4.2 Commands

The Ethernet controller is capable of decoding and encoding commands for both PCI and PCI-X modes. The difference between PCI and PCI-X commands is noted in Table 4-5.

Table 4-5. PCI and PCI-X Encoding Difference

C/BE Encoding	PCI Commands	Abr.	PCI-X Commands	Abr.
0h	Interrupt Acknowledge		Interrupt Acknowledge	
1h	Special Cycle		Special Cycle	
2h	I/O Read	IOR	I/O Read	IOR
3h	I/O Write	IOW	I/O Write	IOW
4h	Reserved		Reserved	
5h	Reserved		Reserved	
6h	Memory Read	MR	Memory Read DWORD	MRD
7h	Memory Write	MW		
8h	Reserved		Alias to MRB	AMR
9h	Reserved		Alias to MWB	AMW
Ah	Configuration Read	CFR	Configuration Read	CFR
Bh	Configuration Write	CFW	Configuration Write	CFW
Ch	Memory Read Multiple	MRM	Split Completion	SC

Table 4-5. PCI and PCI-X Encoding Difference

C/BE Encoding	PCI Commands	Abr.	PCI-X Commands	Abr.
Dh	Dual Address Cycle	DAC	Dual Address Cycle	DAC
Eh	Memory Read Line	MRL	Memory Read Block	MRB
Fh	Memory Write & Invalidate	MWI	Memory Write Block	MWB

As a target, the Ethernet controller only accepts transactions that address its BARs or a configuration transaction in which its IDSEL input is asserted. In PCI-X mode, the Ethernet controller also accepts split completion for an outstanding memory read command that it has requested. The Ethernet controller does not respond to Interrupt Acknowledge or Special Cycle in either mode.

Table 4-6. Accepted PCI/PCI-X Command as a Target

Transaction Target	PCI Commands	PCI-X Commands
Register or Flash Read	MR,MRL,MRM,IOR	MRD, MRB, AMR,IOR
Register or Flash Write	MW, MWI,IOW	MW, MWB, AMW,IOW
Configuration Read	CFR	CFR
Configuration Write	CFW	CFW
Memory Read Completion	N/A	SC

As a master, the Ethernet controller generates Read and Write commands for different causes as listed in Table 4-7. The addresses of these transactions are programmed either by system software or the software driver. The Ethernet controller always expects that they are claimed by one of the devices on the bus segment. The Ethernet controller never generates Interrupt Acknowledge, Special Cycle, I/O commands, or Configuration Commands.

Table 4-7. Generated PCI/PCI-X as a Master

Transaction Cause	PCI Commands	PCI-X Commands	
		CMD	RO
Tx Descriptor Read	MR,MRL,MRM	MRB	1
Tx Descriptor Write back	MW,MWI	MWB	0
Tx Data Read	MR, MRL,MRM	MRB	1
Rx Descriptor Read	MR,MRL,MRM	MRB	1
Rx Descriptor Write back	MW,MWI	MWB	0
Rx Data Write	MW,MWI	MWB	1
Message Signaled Interrupt[a]	MW	MWB	0
Split Completion	N/A	SC	N/A

a. Not applicable to the **82541xx** or **82547GI/EI**.

Transaction burst length on PCI is determined by several factors, including the PCI latency timer expiration, the type of bus transfer (descriptor read/write or data read/write) made, the size of the data transfer (for data transfers), and whether the cycle is initiated by the receive or transmit logic.

Following are a few specific rules:

- For descriptor fetches, the burst length is always equal to the multiple of cache line sizes set by the transmit and receive descriptor fetch threshold fields. (See Section 3.2.4 and Section 3.4.1) For descriptor writes, the transfer size ranges from 8 bytes to N cache line's worth of data. Cache line sizes are: 16, 32, 64, and 128 bytes.
- For transmit data fetches, the burst length is generally equal to the block of data being fetched, in other words, a descriptor's worth of data.
- For receive data writes, the burst size is typically equal to the packet length (rounded up to the next 8 bytes) or the buffer size, whichever is smaller.

4.3 PCI/PCI-X Command Usage

The Ethernet controller optimizes the use of PCI/PCI-X bus cycles to maximize throughput. The following sections describe this behavior.

4.3.1 Memory Write Operations

Memory write command usage has been implemented in the Ethernet controller to improve PCI performance. As noted below, cache line size has a significant impact on the usage of memory write commands. Specifically, cache line size entries which are unsupported causes hardware to default to the Memory Write (MW) command for all master write transactions. Also, all writes default to MW if the Memory Write and Invalidate (MWI) enable bit in the PCI configuration command register is 0b. MWI is the preferred write command and is used when the circumstances allow it.

Figure 4-4 depicts a behavioral state-machine representation of the command usage algorithm for master write operations.

Upon EACH master write access, the hardware evaluates the address alignment and the amount of data to be transferred. The following guidelines are used for command determination:

- If the address is cache line aligned and there is at least one cache line of data, then hardware uses the MWI command.
- If the address is aligned but there is not at least one cache line of data, or the address is not aligned, or if the MWI enable bit is set to 0b, then hardware uses the MW command.

During the burst, regardless of which command was originally issued, the hardware evaluates the remaining amount of data each time the write burst comes to a cache line boundary, or when the transaction is terminated due to a target disconnect or latency timer expiration.

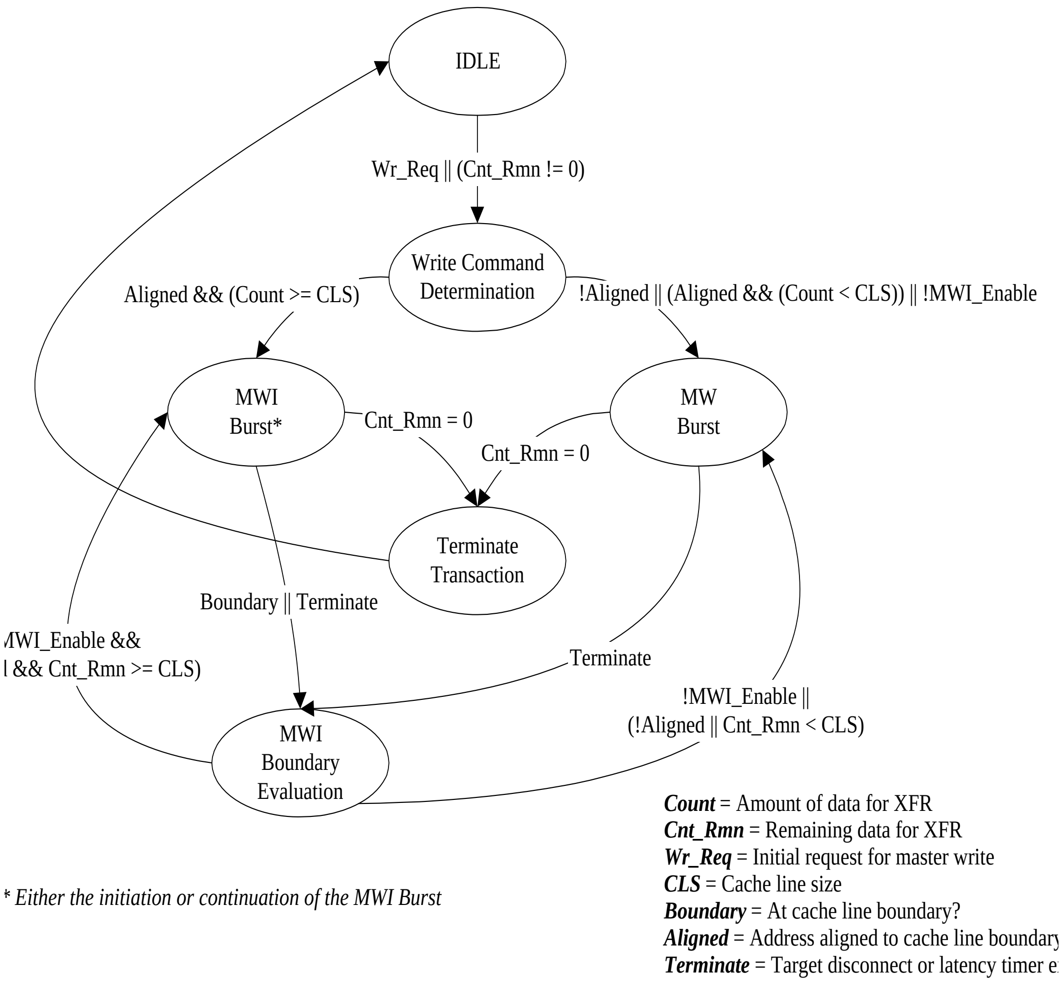


Figure 4-4. Master Write Command Usage Algorithm

4.3.1.1 MWI Bursts

- If there is at least one cache line of data remaining, then the Ethernet controller continues the MWI burst.
- If there is not at least one cache line of data remaining, then the Ethernet controller terminates the transaction on the boundary, re-acquires the bus, and issues a MW command for the remainder of data.
- If the transaction is terminated prematurely due to a target disconnect or latency time-out, the Ethernet controller re-evaluates command usage based on the new start address and the amount of remaining data.

4.3.1.2 MW Bursts

- The Ethernet controller always continues the burst until the end. If the system is concerned about MWI usage, it disconnects at the cache line boundary. The Ethernet controller then restarts the transaction and re-evaluates command usage.

Note: The algorithm described above defaults to the MW case when the MWI enable bit in the Configuration Register is set to 0b.

4.3.2 Memory Read Operations

For all read commands, the hardware evaluates the amount of data to be read with respect to the cache line size register and the read address alignment for command determination. The following rules apply:

Table 4-8. Rules for Memory Read Operations

Amount of Data Requested	Number of Cache Line Boundaries Crossed	Command Used by Hardware
> 2 Cache Lines	n/a	MRM
>= 1 Cache Line	>= 2	MRM
<= 1 Cache Line	0 or 1	MRL
< 1 Cache Line	0	MR

In other words, read command usage depends on the number of cache lines from which the data must be read from the target device.

As mentioned above, unsupported values in the cache line size field default to a size of 32 bytes for the memory read command usage algorithm.

Note: MRL should be used for a single cache line of data that is cache line aligned.

4.3.2.1 PCI-X Command Usage

In PCI-X mode, the Ethernet controller takes advantage of split transaction protocol to minimize retries and eliminate delayed read transactions.

Target Split Responses

When the Ethernet controller responds to a Memory Read or I/O Read command it determines if the data can be returned within 16 clock cycles. If not, it signals a split response and returns data later through the split completion protocol. If the Ethernet controller already has a command in its completion register it retries the requested read until the register is empty. Target posted writes and split completion are still accepted during that period. The internal register reads that cause splits are:

- General Registers: CTRL, STATUS, EECD, CTRL_EXT, FCAL, FCAH, FCT, VET, FCTTV, TXCW, RXCW, PBA,
- Interrupt Registers: ICR, ICS, IMS, IMC
- Transmit Registers: TCTL

Outstanding Memory Read

When the Ethernet controller masters a memory read and is responded to with a split response it waits for the completion of the data as a target. The Ethernet controller allows one outstanding memory read command at any time. The Ethernet controller continues to master posted memory writes and split completions if there are any.

Relaxed Ordering

- The Ethernet controller takes advantage of the relaxed ordering rules in PCI-X. By setting the RO bit for some of its master transactions, the Ethernet controller allows the system to optimize performance in the following cases:
  - Relaxed ordering for descriptor and data reads: When the Ethernet controller masters a read transaction its split completion has no relationship with the writes from the CPUs (same direction). It should be allowed to bypass the writes from the CPUs.
  - Relaxed ordering for receiving data writes: When the Ethernet controller masters receive data writes it also allows them to bypass each other in the path to system memory because the software does not process this data until their associated descriptor writes are done.
- The Ethernet controller cannot relax ordering for descriptor writes or an MSI write.

No Snoop Setting

The Ethernet controller always clears this bit in all of its master transactions because it cannot guarantee that the memory locations between transaction addresses are not cached in the system.

4.4 Cache Line Information[1]

The cache line size PCI configuration register is programmed by the BIOS and/or OS after a system reset. The value in the cache line size register corresponds to the cache line size that the system supports.

The value programmed into the cache line size register affects the DMA operations of the Ethernet controller. In general, the hardware attempts to fetch descriptors on a cache line basis. It also attempts to write back descriptors when a cache line of descriptors has been filled.

The size of the cache line register also has an effect on the Ethernet controller's usage of the MWI PCI command, because the use of this command requires that at least a whole cache line of data is written. The memory read commands are also affected as discussed in Section 4.3.2.

In PCI-X mode, the cache line size does not affect the commands used. However it does affect the descriptor transfer. If an unsupported cache line size larger than 128 bytes is programmed, the Ethernet controller acts as if a cache line size of 128 bytes was programmed.

1. Not applicable to the **82547GI/EI**.

4.4.1 Target Transaction Termination

When the Ethernet controller accepts a transaction as a target it always disconnects the transaction after a single data phase by following the “Master Completion Termination” in PCI 2.2, 2.3, or “Single data phase disconnect termination” in PCI-X. The “memory” in the Ethernet controller is actually a set of registers and is marked as “non-prefetchable”. This is also the case for FLASH memory.

4.5 Interrupt Assignment (82547GI/EI Only)

During a Power-On Self-Test (POST), the system BIOS must assign an Interrupt Request (IRQ) for the **82547GI(EI)**. The **82547GI(EI)** generates an interrupt by sending a hub interface message through the CSA port.

In a typical system, the **82547GI(EI)** component is Device 1 on the bus behind the CSA bridge component. When the GMCH component receives an interrupt message, it forwards the interrupt through a PIRQ programmed into the CSA Interface Interrupt Control Register (CSAINTC). Use the following information to program this register:

```
Address Offset - 48h
Default Value - 04h
Access - R/W
Size - 8 bits
Bit Field - [2:1]
Description - PCI Interrupt type for CSA generated Interrupt - R/W

00 - Reserved
01 - INT#B
10 - INT#C (default)
11 - INT#D
```

Interrupts are communicated with a special CSA bus cycle that causes the MCH to issue an interrupt to the interrupt controller in the ICH.

4.6 LAN Disable

For LAN designs, it is often desirable to program the BIOS setup to selectively enable or disable LOM devices. This capability gives the end user more control over system resource management and avoids conflicts with add-in boards.

Device presence or absence must be established early during BIOS execution to ensure that resource allocation (interrupts and memory) is performed correctly. This task is frequently accomplished using a BIOS CVDR (Configuration Values Driven on Reset) mechanism.

The **82541xx** and **82547GI/EI** LAN disable function resides on the FLSH_SO pin. This pin should be driven by a port on the system Super IO device so that BIOS can control it dependably.

4.7 CardBus Application (82541PI/GI/EI Only)

The **82541PI/GI/EI** has some features to facilitate its use in a CardBus application, following revision 7 of the PC Card specification.

To use the **82541PI/GI/EI** on CardBus, an external flash memory is required. Configure the Base Address Registers to 32-bit (required for CardBus) and enable CLKRUN in EEPROM. Setting these bits also enables the CardBus Information Space (CIS) pointer in the PCI Configuration Space. When enabled, the CIS pointer starts the first tuple of the CIS chain at byte address 20h in the Flash device. The PC Card specification requires that this tuple be a CISTPL_LINKTARGET tuple. The tuple chain can then continue within the memory.

When operating as a CardBus card, the **82541PI/GI/EI** supports full functionality with the exception of wake up.

EEPROM Interface 5

5.1 General Overview

The PCI/PCI-X Family of Gigabit Ethernet Controllers uses an EEPROM device for storing product configuration information. The EEPROM is divided into four general regions:

- **Hardware accessed** – loaded by the Ethernet controller after power-up, PCI Reset deassertion, D3->D0 transition, or software commanded EEPROM reset (CTRL_EXT.EE_RST).
- **ASF accessed –** loaded by the Ethernet controller in ASF mode after power-up, ASF Soft Reset (ASF FRC_RST), or software commanded ASF EEPROM read (ASF FRC_EELD).
- **Software accessed** – used by software only. The meaning of these registers as listed here is a convention for the software only and is ignored by the Ethernet controller.
- **External BMC (TCO) accessed** – loaded by an external BMC (TCO) from the SMBus after power up.

Note: The **82544GC/EI** and **82541ER** do not support ASF, SMBus, or an external BMC (TCO).

Several words of the EEPROM are accessed automatically by the Ethernet controller after reset to provide pre-boot configuration data before it is accessed by host software. The remainder of the stored information is available to software for storing the MAC address, serial numbers, and additional configuration information.

Intel has a software utility called EEUPDATE, which can be used to program EEPROM images in development or production line environments. To obtain a copy of this program, contact your Intel representative.

Note: Since the **82546GB/EB** is a dual port device, there are portions of the EEPROM and Flash that control one or both ports. Special considerations due to this feature are noted in this section.

5.2 Component Identification Via Programming Interface

Ethernet controller stepping is identified by the following register contents.

Table 5-1. Component Identification

Stepping	Vendor ID	Device ID	Description
82547EI-A0	8086h	1019h	Copper
82547EI-A1	8086h	1019h	Copper
82547EI-B0	8086h	1019h	Copper
82547EI-B0	8086h	101Ah	Mobile
82547GI-B0	8086h	1019h	Copper
82546EB-A1	8086h	1010h	Copper; Dual Port MAC Default
82546EB-A1	8086h	1012h	Fiber; Dual Port
82546EB-A1	8086h	101Dh	Copper; Quad Port
82546GB-B0	8086h	1079h	Copper; Dual Port
82546GB-B0	8086h	107Ah	Fiber; Dual Port
82546GB-B0	8086h	107Bh	SerDes; Dual Port
82545EM-A	8086h	100Fh	Copper
82545EM-A	8086h	1011h	Fiber
82545GM-B	8086h	1026h	Copper MAC Default
82545GM-B	8086h	1027h	Fiber
82545GM-B	8086h	1028h	SerDes
82544EI-A4	8086h	1107h	Copper MAC Default
82544GC-A4	8086h	1112h	Copper MAC Default
82541EI-A0	8086h	1013h	Cooper MAC Default
82541EI-A0	8086h	1013h	Cooper MAC Default
82541EI-B0	8086h	1013h	Cooper MAC Default
82541EI-B0	8086h	1018h	Mobile

Table 5-1. Component Identification

Stepping	Vendor ID	Device ID	Description
82541GI-B1	8086h	1076h	Cooper
82541GI-B1	8086h	1077h	Mobile
82541PI-C0	8086h	1076h	Cooper
82541ER-C0	8086h	1078h	Cooper
82540EP-A	8086h	1017	Desktop
82540EP-A	8086h	1016	Mobile
82540EM-A	8086h	100E	Desktop
82540EM-A	8086h	1015	Mobile

Note: These Ethernet controllers also provide identification data through the Test Access Port (TAP).

5.3 EEPROM Device and Interface

The EEPROM access algorithm, programmed into the Ethernet controller, is compatible with most, but not all, commerically available 3.3 V dc Microwire* interfaces and serial EEPROM devices with a 1 MHz speed rating. Ethernet controllers are compatible with two sizes of 4-wire serial EEPROM devices[1]. If ASF mode functionality is desired, a 4096-bit serial NM93C66 compatible EEPROM can be used. Otherwise, a 1024-bit serial NM93C46 compatible EEPROM can be used. Both EEPROMs are accessed in 16-bit words; the larger has 256 words while the smaller has 64 words. Refer to the appropriate Ethernet controller's design guide for recommended EEPROM manufacturers.

An Ethernet controller automatically determines which EEPROM it is connected to and sets the EEPROM SIZE field of the EEPROM/FLASH Control and Data Register (EEC.EE_SIZE) field appropriately. Software can use this field to determine how to access the EEPROM using direct access. Note that different EEPROM sizes have different numbers of address bits and therefore must be accessed with a slightly different serial protocol. Software must be aware of this if it accesses the EEPROM using direct access.

1. The **82544GC/EI** only supports one size of EEPROM. Refer to the *82544GC Gigabit Ethernet Controller Datasheet and Hardware Design Guide* (AP-427) for more information.

The EEPROM interface trace routing is not critical because the interface runs at a very slow speed.

Note: For the **82544GC/EI**, **82540EP/EM**, **82541xx**, and **82547GI/EI**, the EEPROM access algorithm drives extra pulses on the shift clock at the beginnings and ends of read and write cycles. the extra pulses might violate the timing specifications of some EEPROM devices. In selecting a serial EEPROM, choose a device that specifies “don’t care” shift clock states between accesses.

5.3.1 Software Access

The Ethernet controller provides two different methods for software access to the EEPROM. Software can either use the built-in controller to read the EEPROM, or access the EEPROM directly using the EEPROM’s 4-wire interface.

Software can use the EEPROM Read register (EERD) to cause the Ethernet controller to read a word from the EEPROM that the software can then use. To do this, software writes the address to read the Read Address (EERD.ADDR) field and then simultaneously writes a 1b to the Start Read bit (EERD.START). The Ethernet controller then reads the word from the EEPROM, sets the Read Done bit (EERD.DONE), and puts the data in the Read Data field (EERD.DATA). Software can poll the EEPROM Read register until it sees the EERD.DONE bit set, then use the data from the EERD.DATA field. Any words read this way are not written to hardware’s internal registers.

Software can also directly access the EEPROM’s 4-wire interface through the EEPROM/FLASH Control Register (EEC). It can use this for reads, writes, or other EEPROM operations.

To directly access the EEPROM, software should follow these steps:

1. Write a 1b to the EEPROM Request bit (EEC.EE_REQ).
2. Read the EEPROM Grant bit (EEC.EE_GNT) until it becomes 1b. It remains 0b as long as the hardware is accessing the EEPROM.
3. Write or read the EEPROM using the direct access to the 4-wire interface as defined in the EEPROM/FLASH Control & Data Register (EEC). The exact protocol used depends on the EEPROM placed on the board and can be found in the appropriate data sheet.
4. Write a 0b to the EEPROM Request bit (EEC.EE_REQ).

Software can cause the Ethernet controller to re-read the hardware accessed fields of the EEPROM (setting hardware’s internal registers appropriately) by writing a 1b to the EEPROM Reset bit of the Extended Device Control Register (CTRL_EXT.EE_RST). This action will also cause a reset.

5.4 Signature and CRC Fields

The Ethernet controller uses the Signature and CRC fields to determine if an EEPROM is present by attempting to read the EEPROM. The Ethernet controller first reads the Initialization Control Word 1 at address 0Ah and then checks the received value for bits 15 and 14. If bit 15 is 0b and bit 14 is 1b, the Ethernet controller considers the EEPROM to be present and valid. It then reads the additional EEPROM words and programs its internal registers based on the values read. Otherwise, it ignores the values it read from the Initialization Control Word 1 and does not read any other words.

In ASF Mode[1], the Ethernet controller's ASF function reads the ASF CRC word to determine if the EEPROM is valid. If the CRC is not valid, the ASF Configuration registers retain their default value. This CRC does not affect any of the remaining Ethernet controller's configuration, including the Management Control Register.

5.5 EEUPDATE Utility

The EEUPDATE utility meets the two basic requirements for an in-circuit programming utility. First, the utility can be used to update EEPROM images as part of an end-of-line production tool. Secondly, it can be used as a standalone development tool. The tool uses the two basic data files outlined in the following section (static data file and IA address file). To obtain a copy of this program, contact your Intel representative.

The EEUPDATE utility is flexible and can be used to update the entire EEPROM image or update only the IA address of the Ethernet controller.

5.5.1 Command Line Parameters

The DOS command format is a follows:

```
EEUPDATE Parameter_1 Parameter_2
```

where:

```
Parameter_1 = filename or /D

Parameter_2 = filename or /A
```

`Parameter 1`, above, is `file1.eep`, which contains the complete EEPROM image in a specific format that is used to update the complete EEPROM. All comments in the `.eep` file must be preceded by a semicolon (;).

`Parameter 1` can also be a switch /D. the switch /D implies: do not update the complete EEPROM image.

`Parameter 2`, above, is `file2.dat`, which contains a list of IA addresses. the EEUPDATE utility picks up the first unused address from this file and uses it to update the EEPROM. An address is marked as used by following the address with a date stamp. When the utility uses a specific address, it updates that address as used in a log file called eelog.dat. This file should then be used as the `.dat` file for the next update.

Note: Refer to the appropriate Ethernet controller's EEPROM map and programming information for sample EEPROM images.

1. The **82544GC/EI** and **82541ER** do not support ASF.

5.6 EEPROM Address Map[1]

Table 5-2 lists the EEPROM address map for the Ethernet controllers. Each word listed is described in the sections that follow.

Note: The “LAN A/B” column in Table 5-2 is only applicable to the **82546GB/EB**.

Table 5-2. Ethernet Controller Address Map

Word	Used By	Bit 15 - 8	Bit 7 - 0	Image Value	LAN A/B
00h 01h 02h	HW HW HW	Ethernet Address Byte 2 Ethernet Address Byte 4 Ethernet Address Byte 6[a]	Ethernet Address Byte 1 Ethernet Address Byte 3 Ethernet Address Byte 5	IA(2,1) IA(4,3) IA(6,5)	LAN A/B (both)
03h		Compatibility High	Compatibility Low	0000h	both
04h	SW	SerDes Configuration **Note:** Not applicable to the **82540EP/EM**, **82541xx**, and **82547GI/EI**		FFFFh	both
05h	SW	EEPROM Image Version **Note:** Applicable to the **82541xx** and **82547GI/EI** only		0000h	N/A
05h 06h 07h		Compatibility High (Words 06h and 07h reserved for the **82541xx** and **82547GI/EI**)	Compatibility Low (Words 06h and 07h reserved for the **82541xx** and **82547GI/EI**)	0000h 0000h 0000h	both
08h 09h		PBA, byte 1 PBA, byte 3	PBA, byte 2 PBA, byte 4		
0Ah	HW	Init Control 1		4408h 640Ah for the **82541xx** and **82547GI/EI**	both
0Bh	HW	Subsystem ID (Vendor)		see Table 5-1 for specific image values	both
0Ch	HW	Subsystem Vendor ID		8086h	both

1. Refer to Table 5-3 for the **82544GC/EI** and **82541ER** EEPROM address map.

Table 5-2. Ethernet Controller Address Map

Word	Used By	Bit 15 - 8	Bit 7 - 0	Image Value	LAN A/B
0Dh	HW	Device ID		see Table 5-1 for specific image values	LAN A
0Eh	HW	Vendor ID		8086h	both
0Fh	HW	Init Control 2		3040h for the **82545GM/EM** and **82540EP/EM** B080h for the **82541xx** and **82547GI/EI**	both
10h 11h	SW	PHY Registers **82541xx** and **82547GI/EI** only		00BAh 0000h	N/A
10h	HW	Software Defined Pins Control (**82546GB/EB** only) **Note:** Words 10h, 11h, and 13h through 1Fh are reserved for the **82545GM/EM** and **82540EP/EM**		XXXXh	LAN B
11h	HW	Device ID **82546GB/EB** only		see Table 5-1 for specific image values	LAN B
12h	HW	EEPROM Size **82541xx** and **82547GI/EI** only			N/A
12h	HW		Common Power		both
13h 1Eh	SW	PHY Registers **82541xx** and **82547GI/EI** only		00BAh 0000h	N/A
13h	HW	Management Control			LAN B
14h	HW	Init Control 3	SMBus Address[b]	XXXXh	LAN B
15h 16h	HW	IPv4 Address Byte 2 IPv4 Address Byte 4	IPv4 Address Byte 1 IPv4 Address Byte 3	IP(2,1) IP(4,3)	LAN B

Table 5-2. Ethernet Controller Address Map

Word	Used By	Bit 15 - 8	Bit 7 - 0	Image Value	LAN A/B
17h 18h 19h 1Ah 1Bh 1Ch 1Dh 1Eh	HW	IPv6 Address Byte 2 IPv6 Address Byte 4 IPv6 Address Byte 6 IPv6 Address Byte 8 IPv6 Address Byte 10 IPv6 Address Byte 12 IPv6 Address Byte 14 IPv6 Address Byte 16	IPv6 Address Byte 1 IPv6 Address Byte 3 IPv6 Address Byte 5 IPv6 Address Byte 7 IPv6 Address Byte 9 IPv6 Address Byte 11 IPv6 Address Byte 13 IPv6 Address Byte 15	IP(2,1) IP(4,3) IP(6,5) IP(8,7) IP(10,9) IP(12,11) IP(14,13) IP(16,15)	LAN B
1Fh		CSA Port Config 1 (**82547GI/EI** only) **Note:** This word is reserved for all remaining Ethernet controllers		0000h	N/A
20h	HW	Software Defined Pins Control		XXXXh 000Ch for the **82541xx** and **82547GI/EI**	LAN A
21h	HW	CSA Port Config 2 (**82547GI/EI** only)		0002h	N/A
21h	HW	Circuit Control		7863h 7061h for the **82540EP/EM**	both
22h	HW	D0 Power	D3 Power	280Ch 280Bh for the **82541xx** and **82547GI/EI**	both
23h	HW	Management Control		XXC8h XXXXh for the **82541xx** and **82547GI/EI**	LAN A
24h	HW	Init Control 3	SMBus Address[b]	XXXXh XXC8h for the **82545GM/EM** and **82540EP/EM** 001Ch for the **82541xx** and **82547GI/EI**	LAN A

Table 5-2. Ethernet Controller Address Map

Word	Used By	Bit 15 - 8	Bit 7 - 0	Image Value	LAN A/B
25h 26h	HW	IPv4 Address Byte 2 IPv4 Address Byte 4	IPv4 Address Byte 1 IPv4 Address Byte 3	IP(2,1) IP(4,3)	LAN A
27h 28h 29h 2Ah 2Bh 2Ch 2Dh 2Eh	HW	IPv6 Address Byte 2 IPv6 Address Byte 4 IPv6 Address Byte 6 IPv6 Address Byte 8 IPv6 Address Byte 10 IPv6 Address Byte 12 IPv6 Address Byte 14 IPv6 Address Byte 16	IPv6 Address Byte 1 IPv6 Address Byte 3 IPv6 Address Byte 5 IPv6 Address Byte 7 IPv6 Address Byte 9 IPv6 Address Byte 11 IPv6 Address Byte 13 IPv6 Address Byte 15	IP(2,1) IP(4,3) IP(6,5) IP(8,7) IP(10,9) IP(12,11) IP(14,13) IP(16,15)	LAN A
2Fh	HW	LEDCTL Default		0602h	both
30h 31h 32h 33h 34h ... 3Eh	Firm-ware	Intel Boot Agent Configuration … **Note:** Words 34h and 35h are not applicable to the **82545GM/EM** and **82540EP/EM**			
3Fh		Software Checksum, words 00h through 3Fh			
40h ... F7h	ASF	Controlled by the ASF Agent			
F8h ... FFh		Free for Software			

a. The lower bit of the last byte is complemented for LAN B.
b. The SMBus Address is a 7-bit value that is found in bits 7 through 1 of this byte. Bit 0 should be 0b.

Table 5-3. 82544GC/EI and 82541ER EEPROM Address Map

Word Address	HW Access	Description (Hi Byte)	Description (Low Byte)	Default Image Value (hex)
00h	Yes	IA Byte 2	IA Byte 1	IA(2,1)
01h	Yes	IA Byte 4	IA Byte 3	IA(4,3)
02h	Yes	IA Byte 6	IA Byte 5	IA(6,5)
03h	No	Compatibility high	Compatibility low	0000h
04h	No	Reserved		0000h
05h	No	EEPROM Image Version **Note:** Word 05h is reserved for the **82544GC/EI**		0000h
06h 07h	No	Reserved		0000h
08h	No	PBA, byte 1	PBA, byte 2	
09h	No	PBA, byte 3	PBA, byte 4	
0Ah	Yes	Init Control 1, high byte	Init Control 1, low byte	See Text
0Bh	Yes	Subsystem_ID, high byte	Subsystem_ID, low byte	1005h
0Ch	Yes	Subsystem_Vendor, high byte	Subsystem_Vendor, low byte	8086h
0Dh	Yes	Device ID, high	Device ID, low	1008h
0Eh	Yes	Vendor ID, high	Vendor ID, low	8086h
0Fh	Yes	Init Control 2, high byte	Init Control 2, low byte	See Text
10h - 1Fh	No	OEM Reserved	OEM Reserved	0000h
20h	Yes	Software Defined Pins Control, high byte	Software Defined Pins Control, low byte	See Text
21h	Yes	Circuit Control, high	Circuit Control, low	0021h
22h	Yes	D0 Power	D3 Power	See Text
23h - 2Eh	No	Reserved	Reserved	0000h
2Fh	Yes	LEDCTRL Default **Note:** Word 2Fh is reserved for the **82544GC/EI**		0602h
30h - 33h	Firmware	Intel Boot Agent **Note:** Words 30 - 33h are reserved for the **82541ER**		0000h
34h - 3Eh	Fixed	Reserved	Reserved	0000h
3Fh	No	Checksum, high byte	Checksum, low byte	Checksum of words 00h - 3Eh

5.6.1 Ethernet Address (Words 00h-02h)

The Ethernet Individual Address (IA) is a six-byte field that must be unique for each Ethernet port (and unique for each copy of the EEPROM image). The first three bytes are vendor specific. The value from this field is loaded into the Receive Address Register 0 (RAL0/RAH0). For a MAC address of 12-34-56-78-90-AB, words 2:0 load as follows (note that these words are byte-swapped):

Word 0 = 3412

Word 1 = 7856

Word 2 - AB90

Note: Since the **82546GB/EB** is a dual-port device, the Ethernet Address in these words are assigned to LAN A. The Ethernet Address for LAN B is the Ethernet Address for LAN A with its least significant bit inverted.

5.6.2 Software Compatibility Word (Word 03h)

This is the third word read by the Ethernet controller and contains additional initialization values that:

- Sets defaults for some internal registers
- Enables/disables specific features

Note: For the **82544GC/EI**, typical values are 0000h for a fiber-based design and 0400h for a copper-based design.

Table 5-4. Software Compatibility Word (Word 03h)

Bit	Name	Description
15:12	Reserved	Reserved for future use.
11	LOM	LAN on Motherboard (LOM#). Set this bit to 1b (default) to enable LOM#; set to 0b to disable LOM#.
10	SRV	Server card. Set this bit to 1b (default) to enable server card; set to 0b to disable server card. 0b is the default setting for the **82541xx** and **82547GI/EI**.
9	CLI	Client card. Set this bit to 0b (default) to disable client card; set to 1b to enable client card. 1b is the default setting for the **82541xx** and **82547GI/EI**.
8	OEM	OEM card. Set this bit to 1b (default) to enable OEM card; set to 0b to disable OEM card.
7:6	Reserved	Reserved for future use. Set these bits to 0b.
5	Reserved	Reserved for future use. Set this bit to 1b. Set this bit to 0b for the **82540EP/EM**, **82541xx** and **82547GI/EI**.
4	SMB[a]	SMBus. Set this bit to 1b (default) to enable SMBus; set to 0b to disable SMBus.

Table 5-4. Software Compatibility Word (Word 03h)

3	Reserved	Reserved for future use. Set this bit to 0b.
2	BOB	PCI bridge. Set this bit to 0b (default) to disable PCI bridge; set to 1b to enable PCI bridge. 1b is the default setting for the **82540EP/EM**.
1:0	Reserved	Reserved for future use. Set these bits to 0b.

a. Not applicable to the **82544GC/EI** or **82541ER**.

5.6.3 SerDes Configuration (Word 04h)

If this word has a value of other than FFFFh, software programs its value into the Extended PHY Specific Control Register 2, located at address 26d in the PHY register space (see Table 13-47).

Note: SerDes Configuration (Word 04h) is a reserved area for the **82544GC/EI, 82540EP/EM**, **82541xx**, and **82547GI/EI**.

5.6.4 EEPROM Image Version (Word 05h)

Word 05h determines the EEPROM image version for the **82541xx** and the **82547GI/EI**.

Bits	Name	Value
15:12	EEPROM major version.	0000h
11:8	EEPROM minor version.	0000h
7:0	EEPROM fix.	00000000h

5.6.5 Compatibility Fields (Word 05h - 07h)

These areas are reserved for compatibility information and are used by software drivers.

5.6.6 PBA Number (Word 08h, 09h)

A nine-digit Printed Board Assembly (PBA) number, used for Intel manufactured adapter cards, are stored in a four-byte field. Other hardware manufacturers can use these fields as they wish. Software device drivers should not rely on this field to identify the product or its capabilities.

5.6.7 Initialization Control Word 1 (Word 0Ah)

The first word read by the Ethernet controller contains initialization values that:

- Sets defaults for some internal registers
- Enables/disables specific features
- Determines which PCI configuration space values are loaded from the EEPROM

Table 5-5. Initialization Control Word 1 (Word 0Ah)

Bit	Name	Description
15:14	Signature	The Signature field represents a signature of 01b (default), indicating to the Ethernet controller that there is a valid EEPROM present. If the Signature field is not 01b, the other bits in this word are ignored, no further EEPROM read is performed, and default values are used for the configuration space IDs.
13	64/32 BAR	When set to 0b (default), enables 64-bit memory mapping. When set to 1b, disables 64-bit memory mapping **Note:** Set to 1b for the **82540EP/EM**.
12	IPS0	When set to 0b (default), does not invert the Power State Output bit 0 (CTRL_EXT[14]). When set to 1b, inverts the Power State Output invert bit 0 (CTRL_EXT[14]). **Note:** Reserved bit for the **82541xx** and **82547GI/EI** (set to 0b).
11	FRCSPD	Force Speed bit in the Device Control register (CTRL[11]). When set to 0b (default), does not force speed. When set to 1b, forces speed (default for the **82540EP/EM**). For 10/100/1000 Mb/s systems using TBI mode (**82544GC/EI**)/internal SerDes (**82546GB/EB** and **82545GM/EM**)[a], set this bit to 0b. **Note:** Reserved bit for the **82541xx** and **82547GI/EI** (set to 0b).
10	FD	Full Duplex (mapped to CTRL[0] and TXCW[5]). When set to 1b (default), enables full duplex (TBI mode/internal SerDes only). When set to 0b, disables full duplex (TBI mode only/internal SerDes). **Note:** Reserved bit for the **82541PI/GI/EI** and **82547GI/EI** (set to 1b). **Note:** Reserved bit for the **82541ER** (set to 0b).
9	LRST	Link Reset (mapped to CTRL[3]). When set to 0b, enables Auto-Negotiation at power up or when asserting RST# without driver intervention. When set to 1b, disables Auto-Negotiation at power up or when asserting RST# without driver intervention. **Note:** Reserved bit for the **82541xx** and **82547GI/EI** (set to 0b).
8	IPS1	When set to 0b (default), does not invert the Power State Output bit 1 (CTRL_EXT[16]). When set to 1b, inverts the Power State Output invert bit 1 (CTRL_EXT[16]). **Note:** Reserved bit for the **82541xx** and **82547GI/EI** (set to 0b).

Table 5-5. Initialization Control Word 1 (Word 0Ah)

Bit	Name	Description
7	Internal VREG Power Down Control	**82541xx** and **82547GI/EI** Only This bit is used to define the usage of the internal 1.2 V dc and 1.8 V dc regulators to supply power. 0b = Yes (default). 1b = No (external regulators used). **Note:** Reserved bit for all other Ethernet controllers.
6:5	Reserved	Reserved for future use. Set these bits to 0b.
4	Reserved	Reserved for copper PHY. Set this bit to 0b.
3	Power Management	When set to 1b (default), enables full support for power management. When set to 0b, the Power Management Registers set is read only. The Ethernet controller does not execute a hardware transition to D3. **Note:** Reserved bit for the **82541PI/GI/EI** and **82547GI/EI** (set to 1b). **Note:** Reserved bit for the **82541ER** (set to 0b).
2	PME Clock	When set to 0b (default), indicates that the PCICLK is not required for PME# output. When set to 1b, indicates that the PCICLK is required for PME# output. **Note:** Reserved bit for the **82541xx** and **82547GI/EI** (set to 0b).
1	Load Subsystem IDs	When set to 1b (default), indicates that the Ethernet controller is to load its PCI Subsystem ID and Subsystem Vendor ID from the EEPROM (words 0Bh, 0Ch). When set to 0b, indicates that the Ethernet controller is to load the default PCI Subsystem ID and Subsystem Vendor ID.
0	Load Vendor/Device IDs	When set to 0b (default), indicates that the Ethernet controller is to load the default values for PCI Vendor and Device IDs. When set to 1b (default for the **82541xx** and **82547GI/EI** only), indicates that the Ethernet controller is to load its PCI Vendor and Device IDs from the EEPROM (words 0Dh, 0Eh).

a. Not applicable to the **82541xx**, **82547GI/EI** or **82540EP/EM**.

5.6.8 Subsystem ID (Word 0Bh)

If the signature bits (15:14) and bit 1 (Load Subsystem IDs) of word 0Ah are valid, this word is read in to initialize the Subsystem ID.

5.6.9 Subsystem Vendor ID (Word 0Ch)

If the signature bits (15:14) and bit 1 (Load Subsystem IDs) of word 0Ah are valid, this word is read in to initialize the Subsystem Vendor ID.

5.6.10 Device ID (Word 0Dh, 11h[1])

If the signature bits (15:14) and bit 1 (Load Subsystem IDs) of word 0Ah are valid, this word is read in to initialize the Subsystem ID.

For the **82546GB**, the Device ID must be forced to 107Bh for SerDes-SerDes interface operation. For the **82545GM**, the Device ID should be 1028h. This ensures proper functionality with Intel drivers and boot agent.

Note: Since the **82546GB/EB** is a dual-port device, the Device ID in 0Dh corresponds to LAN A and the Device ID in 11h corresponds to LAN B.

5.6.11 Vendor ID (Word 0Eh)

If the signature bits (15:14) and bit 1 (Load Subsystem IDs) of word 0Ah are valid, this word is read in to initialize the Subsystem ID.

5.6.12 Initialization Control Word 2 (Word 0Fh)

This is the second word read by the Ethernet controller and contains additional initialization values that:

- Sets defaults for some internal registers
- Enables/disables specific features

Table 5-6. Initialization Control Word 2 (Word 0Fh)

Bit	Name	Description
15	APM PME# Enable	Initial value of the Assert PME On APM Wakeup bit in the Wakeup Control Register (WUC.APMPME). When set to 0b (default), deasserts PME# on wakeup. Set this bit to 1b for Intel LAN controller cards.
14	ASDE	When set to 0b (default), indicates the initial value of the Auto-Speed Detection Enable (ASDE) bit of the Device Control Register (CTRL). When set to 1b, enables 10/100/1000 Mb/s systems.
13:12	PAUSE Capability	Pause Capability - Mapped to TXCW[8:7]. When set to 1b (default), enables desired PAUSE capability for an advertised configuration base page. When set to 0b, disables desired PAUSE capability for an advertised configuration base page.
11	ANE Reserved bit for the **82541xx** and **82547GI/EI** (set to 0b).	Auto-Negotiation Enable. Mapped to TXCW[31]. Set this bit to 1b to automatically enable Auto-Negotiation. Set this bit to 0b (default) to automatically disable Auto-Negotiation. **Note:** Fiber implementations do not support this function.
10:9	FLASH Size Indication	Indicates FLASH size. 00b = 64 KB (default); 01b = 128 KB; 10b = 256 KB; 11b = 512 KB. These bits also impact the requested memory space for the FLASH and Expansion ROM BARs in PCI configuration space.

1. Word 11h only applicable to the **82546GB/EB**.

Table 5-6. Initialization Control Word 2 (Word 0Fh)

Bit	Name	Description
8	MAC Clock Speed	**82541PI/GI** Only. 0b = MAC runs at full speed. 1b = MAC runs at 1/4 speed on any drop from 1000 Mb/s. **Note:** Reserved bit for all other Ethernet controllers (set to 0b). Formally FLASH Disable, now located in Initialization Control Word 3, bit 3.
7	MSI Disable	When set to 0b (default), enables Message Signalled Interrupts (MSI) in standard PCI mode. When set to 1b, disables Message Signalled Interrupts (MSI) in standard PCI mode. **Note:** Reserved bit for the **82541xx** and **82547GI/EI** (set to 1b).
6	133 MHz Capable	When set to 1b (default), maps the 133 MHz Capable bit of the PCI-X Status Register (PCIXS). When set to 0b, does not map the 133 MHz Capable bit of the PCI-X Status Register (PCIXS). **Note:** Reserved bit for the **82541xx**, **82547GI/EI**, and **82540EP/EM** (set to 0b).
5	DMCR_Map	Indicates how the Designed Maximum Cumulative Read size bits in the PCI-X Status register are mapped. When set to 1b (default), the DMCR value reflects the hard-coded design capability as indicated by the Max_Read bit (bit 4). When set to 0b, the DMCR is mapped directly to the Maximum Memory Read Byte Count indicated in the PCI-X Command register. **Note:** Reserved bit for the **82541xx**, **82547GI/EI**, and **82540EP/EM** (set to 0b).
4	Max_Read	Indicates the maximum read value as advertised in the Designed Maximum Memory Read Byte Count field in the PCI-X Status Register. When set to 0b (default), or if there is no EEPROM, the advertised maximum read is 2 KB. When set to 1b, the advertised maximum read is 4 KB. Note that it is not recommended to set Max_Read to 1b because transmit FIFO overruns are possible under specific operating conditions. **Note:** Reserved bit for the **82541xx**, **82547GI/EI**, and **82540EP/EM** (set to 0b).
3	64-bit	When set to 1b (default), loads the 64-bit Device field of the PCI-X Status Register. When set to 0b, does not load the 64-bit Device field of the PCI-X Status Register. **Note:** Reserved bit for the **82541xx**, **82547GI/EI**, and **82540EP/EM** (set to 0b).
2	Reserved	Reserved for future use (set to 0b). Formerly APM Enable, now located in Initialization Control Word 3, bit 2. **Note:** Set to 1b for the **82541xx** and **82547GI/EI**.
1	Force CSR Read Split	When set to 0b (default), certain critical registers are decoded for non-split access. When set to 1b, forces all Ethernet controller control/status register-reads to be split when operating in a PCI-X environment. **Note:** Reserved bit for the **82541xx** and **82547GI/EI** (set to 0b).
0	Reserved	Reserved for future use (set to 0b).

5.6.13 PHY Register Address Data (Words 10h, 11h, and 13h - 1Eh)

These settings are specific to individual platform configurations for the **82541xx** and **82547GI/EI** and should not be altered from the reference design unless instructed to do so. Future Intel Ethernet controllers might use this space differently.

5.6.14 OEM Reserved Words (Words 10h, 11h, 13h - 1Fh)

Words 10h, 11h, and 13h through 1Fh of the EEPROM are reserved areas for general OEM use for all Ethernet controllers except the **82546GB/EB**.

5.6.15 EEPROM Size (Word 12h)

This word is only applicable to **82541xx** and **82547GI/EI** Ethernet controllers that use SPI EEPROMs. Unused bits are reserved and should be programmed to 0b. Bits 15:13 and 8:0 are reserved (see Table 5-7).

Table 5-7. SPI EEPROM Sizes

Bits 12:10	Bit 9	EEPROM Size (Bits)	EEPROM Size (Bytes)
000	0	1 Kb	128 Bytes
001	1	4 Kb	512 Bytes
010	1	8 Kb	1 KB
011	1	16 Kb	2 KB
100	1	32 Kb	4 KB
101	1	64 Kb	8 KB
110	1	128 Kb	16 KB
111	1	Reserved	Reserved

5.6.16 Common Power (Word 12h)

For all Ethernet controllers except the **82541xx** and **82547GI/EI**, if the signature bits are valid and Power Management is not disabled, the value in this field is used in the PCI Power Management Data Register when the *Data_Select* field of the Power Management Control/Status Register (PMCSR) is set to 8. This setting indicates the power usage and heat dissipation of the common logic that is shared by both functions in tenths of a watt.

5.6.17 Software Defined Pins Control (Word 10h[1], 20h)

This field contains initial settings for the Software Defined Pins (SPD). The default value for the upper byte (bits 15:8) is DFh; the default value for the lower byte (bits 7:0) is DEh.

1. Applicable to the **82546GB/EB** only.

Table 5-8. Software Defined Pins Control (Word 10h, 20h)

Bit	Name	Description
15	SDPDIR[7] SDPDIR[3] for the **82541xx** and **82547GI/EI**	SDP7(3) Pin - Initial Direction. Set this bit to 0b (default) to configure the initial hardware value of the SDP7(3)_IODIR bit in the Extended Device Control Register (CTRL_EXT) following power up. Set this bit to 1b if not connected on a board or if used as an output.
14	SDPDIR[6] SDPDIR[2] for the **82541xx** and **82547GI/EI**	SDP6(2) Pin - Initial Direction. Set this bit to 0b (default) to configure the initial hardware value of the SDP6(2)_IODIR bit in the Extended Device Control Register (CTRL_EXT) following power up. Set this bit to 1b if not connected on a board or if used as an output.
13:10	Reserved	Set these bits to 0b.
9	SDPDIR[1]	SDP1 Pin - Initial Direction. Set this bit to 0b (default) to configure the initial hardware value of the SDP1_IODIR bit in the Extended Device Control Register (CTRL_EXT) following power up. Set this bit to 1b if not connected on a board or if used as an output.
8	SDPDIR[0]	SDP0 Pin - Initial Direction. Set this bit to 0b (default) to configure the initial hardware value of the SDP0_IODIR bit in the Extended Device Control Register (CTRL_EXT) following power up. Set this bit to 1b if not connected on a board or if used as an output.
7	SDPVAL[7] SDPVAL[3] for the **82541xx** and **82547GI/EI**	SDP7(3) Pin - Initial Output Value. Set this bit to 0b (default) to configure the initial power-on value output on SDP7(3) (when configured as an output) by configuring the initial hardware value of the SDP7(3)_DATA bit in the Extended Device Control Register (CTRL_EXT) after power up. Set this bit to 1b if used as an output.
6	SDPVAL[6] SDPVAL[2] for the **82541xx** and **82547GI/EI**	SDP6(2) Pin - Initial Output Value. Set this bit to 0b (default) to configure the initial power-on value output on SDP6(2) (when configured as an output) by configuring the initial hardware value of the SDP6(2)_DATA bit in the Extended Device Control Register (CTRL_EXT) after power up. Set this bit to 1b if used as an output.
5:4	Reserved	Set these bits to 0b.
3	EN_PHY_PWR_MGMT	Set this bit to 1b (default) to configure the initial hardware default value of this bit in the Device Control Register (CTRL) following power up. Set this bit to 0b to not configure the initial hardware default value of this bit in the Device Control Register (CTRL) following power up.

Table 5-8. Software Defined Pins Control (Word 10h, 20h)

Bit	Name	Description
2	D3COLD_WAKEUP_ADV_EN	Set this bit to 1b (default) to configure the initial hardware default value of the ADVD3WUC bit in the Device Control Register (CTRL) following power up. Set this bit to 0b to not configure the initial hardware default value of the ADVD3WUC bit in the Device Control Register (CTRL) following power up.
1	SDPVAL[1]	SDP1 Pin - Initial Output Value. Set this bit to 0b (default) to configure the initial power-on value output on SDP1 (when configured as an output) by configuring the initial hardware value of the SDP1_DATA bit in the Extended Device Control Register (CTRL_EXT) after power up. Set this bit to 1b if used as an output.
0	SDPVAL[0]	SDP0 Pin - Initial Output Value. Set this bit to 0b (default) to configure the initial power-on value output on SDP0 (when configured as an output) by configuring the initial hardware value of the SDP0_DATA bit in the Extended Device Control Register (CTRL_EXT) after power up. Set this bit to 1b if used as an output.

Note: Since the **82546GB/EB** is a dual-port device, the SDP control in 10h corresponds to LAN B, and the SDP control in 20h corresponds to LAN A.

5.6.18 CSA Port Configuration 2 (Word 21h)

For the **82547GI/EI** only, this word controls the CSA port configuration and must be programmed to 93A7h for regular operation (see Table 5-9).

Table 5-9. CSA Port Configuration 2 (Word 21h)

Bit	Description	Default
15:13	Reserved.	Set to 100b.
12	Reserved.	Set to 1b.
11:2	Reserved.	Set to 0011101001b.
1	Dock/Undock Polarity.	1b = Indicates Docked (default). 0b = Indicates Undocked.
0	Reserved.	Set to 1b.

5.6.19 Circuit Control (Word 21h)

This word is loaded into the Circuit Control Register (CIRC) for setting PCI-X driver strength. See Table 5-2 and Table 5-3 for suggested values.

Note: PCI-X is not applicable to the **82540EP/EM**, **82541xx**, and **82547GI/EI**.

5.6.20 D0 Power (Word 22h high byte)

If the signature bits are valid and Power Management is not disabled, the value in this field is used in the PCI Power Management Data Register when the Data_Select field of the Power Management Control/Status Register (PMCSR) is set to 0 or 4. This indicates the power usage and heat dissipation of the networking function (including the Ethernet controller and any other devices managed by the Ethernet controller) in tenths of a watt. For example:

```
If word22h = 290E, POWER CONSUMPTION (in 1/10W, hex), then:

    bits 15:8 = 29h Power in D0a, 29h = 4.1W

    bits 7:0 = 0Eh Power in D3h, 0Eh = 1.4W
```

5.6.21 D3 Power (Word 22h low byte)

If the signature bits are valid and Power Management is not disabled, the value in this field is used in the PCI Power Management Data Register when the Data_Select field of the Power Management Control/Status Register (PMCSR) is set to 3 or 7. This indicates the power usage and heat dissipation of the networking function (including the Ethernet controller and any other devices managed by the Ethernet controller) in tenths of a watt as described in Section 5.6.20.

5.6.22 Reserved Words (23h - 2Eh)

Words 23h through 2Eh of the EEPROM are reserved areas for the **82541ER**.

5.6.23 Reserved Words (23h - 2Fh)

Words 23h through 2Fh of the EEPROM are reserved areas for the **82544GC/EI**.

5.6.24 Management Control (Word 13h[1], 23h[2])

The following table lists the initial settings for the Management Control Register as well as valid bits for the IPv4 Address and the IPv6 Address.

Table 5-10. Initial Management Control Register Settings

Bit	Name	Description
15	Enable ARP Response Filtering	This bit controls the initial value of the *MANC.RSP_EN* bit. 0b: ARP response packets are delivered to host memory. 1b: ARP response packets are delivered to the Ethernet controller for automatic ARP reply or forwarded to the BMC.
14	Reserved	Reserved. Set this bit to 0b.
13	Enable ARP Request Filtering	This bit controls the initial value of the *MANC.ARP_EN* bit. 0b: ARP request packets are delivered to host memory. 1b: ARP request packets are delivered to the Ethernet controller for automatic ARP reply or forwarded to the BMC.
12:10	Reserved	Set these bits to 0b.
9	Enable RMCP 0298h Filtering	This bit is controlled by the ASF agent. Manually set this bit for TCO mode. Set this bit to 0b (default) to control the initial value of the *MANC.0298_EN* bit, which enables classifying UDP packets of port 0298h as Management Packets for delivery to the SMBus or ASF controller. Set this bit to 0b if the SMBus is disabled.
8	Enable RMCP 026F Filtering	This bit is controlled by the ASF agent. Manually set this bit for TCO mode. Set this bit to 1b (default) to control the initial value of the *MANC.0298_EN* bit, which enables classifying UDP packets of port 026Fh as Management Packets for delivery to the SMBus or ASF controller. Set this bit to 0b if the SMBus is disabled.
7	IPv6 Address Valid	IPv6 Address in the IP Address EEPROM register is valid. This is written to bit 16 of the IP Address Valid (IPAV[16]) register.
6	IPv4 Address Valid	IPv4 Address in the IP Address EEPROM register is valid. This is written to bit 0b of the IP Address Valid (IPAV[0]) register.
5	Flex Filter Enable	**82541PI/GI/EI** and **82547GI/EI** Only. This bit enables the flexible filter loaded from the EEPROM. 0b = Disable flex filter. 1b = Enable flex filter. **Note:** Reserved bit for all other Ethernet controllers (set to 0b).
4:3	Reserved	Set these bits to 0b.

1. Applicable to the **82546GB/EB** only.
2. Not applicable to the **82544GC/EI** or **82541ER**.

Table 5-10. Initial Management Control Register Settings

Bit	Name	Description
2	Reset on Force TCO	This bit is controlled by the ASF agent. Manually set this bit for TCO mode. Reset the Ethernet controller on a ForceTCO SMBus Command with the “Force” bit set to 1b (default) in TCO mode, or on various conditions in ASF mode. Set this bit to 0b if the SMBus is disabled.
1	ASF Mode	This bit is controlled by the ASF agent. Set this bit to 1b to enable ASF mode. Set this bit to 0b (default) for TCO mode and Intel’s ASF Agent implementation.
0	SMBus Enable	This bit is controlled by the ASF agent. Manually set this bit for TCO mode. Set this bit to 1b (default) to enable SMBus functionality. Set this bit to 0b to enable ASF mode if its being routed by LAN A and if LAN A is the active interface for ASF to the BIOS.

Note: Since the **82546GB/EB** is a dual-port device, the Management Control in 13h corresponds to LAN B, and the Management Control in 23h corresponds to LAN A.

5.6.25 SMBus Slave Address (Word 14h[1] low byte, 24h low byte)

The following table lists the SMBus slave address for TCO mode.

Table 5-11. SMBus Slave Address

Bit	Name	Description
7:1	SMBus Slave Address	These bits are controlled by the ASF agent. Manually set these bits for TCO mode. Contains the SMBus slave address for TCO mode. This address must be 1100b 100b for ASF mode.
0	Reserved	Set this bit to 0b.

Note: This byte must be C8h for ASF mode. For example, to program an address of 0011_001b, the byte should be set to 0011_0010b. When this address is used on the SMBus, the address byte is 0011_0010b for writes and 0011_0011b for reads.

Note: Since the **82546GB/EB** is a dual-port device, the SMBus Slave Address in 14h corresponds to LAN B, and the SMBus Slave Address in 24h corresponds to LAN A.

1. Applicable to the **82546GB/EB** only.

5.6.26 Initialization Control 3 (Word 14h[1] high byte, 24h high byte)

This word controls the general initialization values.

Table 5-12. Initialization Control 3

Bit	Name	Description
7:5	Reserved	Reserved. Set these bits to 0b.
4	Interrupt Pin	Controls the value advertised in the Interrupt Pin field of the PCI Configuration header for this device/function. A value of 0b (default), reflected in the Interrupt Pin field, indicates that the **82546GB/EB** uses INTA#; a value of 1b indicates that the **82546GB/EB** uses INTB#. If only a single device/function of the Ethernet controller is enabled, this value is ignored and the Interrupt Pin field of the enabled device reports INTA# usage.
3	FLASH Disable	Set this bit to 0b (default) to enable the FLASH logic. Set this bit to 1b to disable the FLASH logic. Note that the Expansion ROM & secondary FLASH access BARs in PCI configuration space are also disabled.
2	APM Enable	Initial value of Advanced Power Management Wakeup Enable in the Wakeup Control Register (WUC.APME). The default for this bit is 0b.
1:0	Link Mode	Initial value of Link Mode bits of the Extended Device Control Register (CTRL_EXT.LINK_MODE), specifying which link interface and protocol is used by the MAC. **For Address 24h (High Byte) / LAN A** 00b = MAC operates in GMII/MII mode with internal copper PHY[a] 01b = External GMII/MII mode 10b = Internal SerDes mode (not applicable to the **82540EP/EM**) 11b = MAC operates in TBI mode using external TBI interface **For Address 14h (High Byte) / LAN B** 00b = MAC operates in GMII/MII mode with internal copper PHY 01b = Reserved 10b = Internal SerDes mode (not applicable to the **82540EP/EM**) 11b = MAC operates in TBI mode using external TBI interface

a. For the **82540EP/EM**, **82541PI/GI/EI**, and **82547GI/EI** to properly communicate with the internal copper PHY, this value must be set to 00b.

Note: Since the **82546GB/EB** is a dual-port device, the Initialization Control Word 3 bit assignments are port specific.

1. Applicable to the **82546GB/EB** only.

5.6.27 IPv4 Address (Words 15h - 16h[1] and 25h - 26h)

The following table lists the initial values for the IPv4 addresses.

Table 5-13. IPv4 Addresses

Bit	Name	Description
31:0	IPv4 Address	The initial value of IPv4 Address Table entry 0. (IP4AT[0]). Refer to the EEPROM Address Map listed in Table 5-2 for an indication of how the bytes are stored.

Note: Since the **82546GB/EB** is a dual-port device, the IPv4 Address in 15h-16h corresponds to LAN B, and the IPv4 Address in 25h-26h corresponds to LAN A.

5.6.28 IPv6 Address (words 17h - 1Eh[1] and 27h - 2Eh)

The following table lists the initial values for the IPv6 addresses.

Table 5-14. IPv6 Address

Bit	Name	Description
127:0	IPv6 Address	The initial value of IPv6 Address Table entry 0. (IP6AT[0]) Refer to the EEPROM Address Map listed in Table 5-2 for an indication of how the bytes are stored.

Note: Since the **82546GB/EB** is a dual-port device, the IPv6 Address in 17h-1Eh corresponds to LAN B, and the IPv6 Address in 27h-2Eh corresponds to LAN A.

1. Applicable to the **82546GB/EB** only.

5.6.29 LED Configuration Defaults (Word 2Fh)[1]

This EEPROM word specifies the hardware defaults for the LEDCTL register fields controlling the LED0/(LINK_UP#) and LED2/LINK100 output behaviors. Refer to Table 13-60 for the LED Control bit descriptions and Table 13-61 for the Mode Encodings.

Note: A value of 0602h is used to configure default hardware LED behavior equivalent to 82544-based Copper adapters (LED0/LINK_UP#, LED1/ACTIVITY# (blinking), LED2/LINK100#, and LED3/LINK1000#).

Table 5-15. LED Configuration Defaults

Bit	Name	Description
3:0	LED0 Mode	Initial value of the LED0_MODE field specifying what event/state/ pattern will be displayed on LED0 (LINK_UP) output. A value of 0010 (0x2) causes this to indicate LINK_UP state.
5:4	Reserved	Reserved. Set as 0b.
6	LED0 Invert	Initial value of LED0_IVRT field. 0b = Active-low output.
7	LED0 Blink	Initial value of LED0_BLINK field. 0b = Non-blinking.
11:8	LED2 Mode	Initial value of the LED2_MODE field specifying what event/state/ pattern will be displayed on LED2 (LINK_100) output. A value of 0110b (0x6) causes this to indicate 100 Mb/s operation.
13:12	Reserved	Reserved. Set as 0b.
14	LED2 Invert	Initial value of LED2_IVRT field. 0b = Active-low output.
15	LED2 Blink	Initial value of.LED2_BLINK field. 0b = Non-blinking

5.6.30 Boot Agent Main Setup Options (Word 30h)

The boot agent software configuration is controlled by the EEPROM with the main setup options stored in word 30h. These options are those that can be changed by using the Control-S setup menu or by using the IBA Intel Boot Agent utility.

Note: The **82541ER** does not support the Intel Boot Agent functionality.

1. Not applicable to the **82544GC/EI**.

Table 5-16. Boot Agent Main Setup Options

Bit	Name	Description
15	PPB	PXE Presence. Setting this bit to 0b Indicates that the image in the FLASH contains a PXE image. Setting this bit to 1b indicates that no PXE image is contained. The default for this bit is 0b in order to be backwards compatible with existing systems already in the field. If this bit is set to 0b, EEPROM word 32h (PXE Version) is valid. When EPB is set to 1b and this bit is set to 0b, indicates that both images are present in the FLASH.
14	EPB	EFI Presence. Setting this bit to 1b Indicates that the image in the FLASH contains an EFI image. Setting this bit to 0b indicates that no EFI image is contained. The default for this bit is 0b in order to be backwards compatible with existing systems already in the field. If this bit is set to 1b, EEPROM word 33h (EFI Version) is valid. When PPB is set to 0b and this bit is set to 1b, indicates that both images (PXE and EFI) are present in the FLASH.
13	Reserved	Reserved for future use. Set this bit to 0b.
12	FDP	Force Full Duplex. Set this bit to 0b for half duplex; set to 1b for full duplex. Note that this bit is a don’t care unless bits 10 and 11 are set.
11:10	FSP	These bits determine speed. 01b = 10 Mbs, 10b = 100 Mbs, 11b = Not allowed. All zeros indicate Auto-negotiate (the current bit state). Note that bit 12 is a don’t care unless these bits are set.
9	Reserved	Reserved for future use. Set this bit to 0b.
8	DSM	Display Setup Message. If this bit is set to 1b, the "Press Control-S" message appears after the title message. The default for this bit is 1b.
7:6	PT	Prompt Time. These bits control how long the "Press Control-S" setup prompt message appears, if enabled by DIM. 00b = 2 seconds (default) 01b = 3 seconds 10b = 5 seconds 11b = 0 seconds Note that the Ctrl-S message does not appear if 0 seconds prompt time is selected.
5	LBS	Local Boot Selection (OBSOLETE). In previous versions of the agent, this bit enables or disables local boot, if the DBS bit selects it. The default for this bit is 1b; enable local booting. The boot agent, at runtime, no longer uses this bit.

Table 5-16. Boot Agent Main Setup Options

Bit	Name	Description
4:3	DBS	Default Boot Selection. These bits select which device is the default boot device. These bits are only used if the agent detects that the BIOS does not support boot order selection or if the MODE field of word 31h is set to MODE_LEGACY. 00b = Network boot, then local boot 01b = Local boot, then network boot 10b = Network boot only 11b = Local boot only
2	BBS	BIOS Boot Specification (OBSOLETE). In previous versions of the agent, this bit enables or disables use of the BBS to determine boot order. If set to 1b, the BIOS boot order is used, and the DBS bits are ignored. The boot agent at runtime no longer uses this bit. The runtime checks for BBS/PnP and the setting in the MODE field of word 31h are used instead.
1:0	PS	Protocol Select. These bits select the boot protocol. 00b = PXE (default value) 01b = RPL protocol Other values are undefined.

5.6.31 Boot Agent Configuration Customization Options (Word 31h)

Word 31h contains settings that can be programmed by an OEM or network administrator to customize the operation of the software. These settings cannot be changed from within the Control-S setup menu or the IBA Intel Boot Agent utility. The lower byte contains settings that would typically be configured by a network administrator using the Intel Boot Agent utility; these settings generally control which setup menu options are changeable. The upper byte are generally settings that would be used by an OEM to control the operation of the agent in a LOM environment, although there is nothing in the agent to prevent their use on a NIC implementation

.

Table 5-17. Boot Agent Configuration Customization Options (Word 31h)

Bit	Name	Description
15:14	SIG	Signature. These bits must be set to 1b to indicate that this word has been programmed by the agent or other configuration software.
13:11	Reserved	Reserved for future use. Set these bits to 0b.
10:8	MODE	Selects the agent's boot order setup mode. This field changes the agent's default behavior in order to make it compatible with systems that do not completely support the BBS and PnP Expansion ROM standards. Valid values and their meanings are: 000b - Normal behavior. The agent attempts to detect BBS and PnP Expansion ROM support as it normally does. 001b - Force Legacy mode. The agent does not attempt to detect BBS or PnP Expansion ROM supports in the BIOS and assumes the BIOS is not compliant. The BIOS boot order can be changed in the Setup Menu. 010b - Force BBS mode. The agent assumes the BIOS is BBS-compliant, even though it may not be detected as such by the agent's detection code. The BIOS boot order CANNOT be changed in the Setup Menu. 011b - Force PnP Int18 mode. The agent assumes the BIOS allows boot order setup for PnP Expansion ROMs and hooks interrupt 18h (to inform the BIOS that the agent is a bootable device) in addition to registering as a BBS IPL device. The BIOS boot order CANNOT be changed in the Setup Menu. 100b - Force PnP Int19 mode. The agent assumes the BIOS allows boot order setup for PnP Expansion ROMs and hooks interrupt 19h (to inform the BIOS that the agent is a bootable device) in addition to registering as a BBS IPL device. The BIOS boot order CANNOT be changed in the Setup Menu. 101b - Reserved for future use. If specified, treated as value 000b. 110b - Reserved for future use. If specified, treated as value 000b. 111b - Reserved for future use. If specified, treated as value 000b.
7:6	Reserved	Reserved for future use. Set these bits to 0b.
5	DFU	Disable FLASH Update. If set to 1b, no updates to the FLASH image using PROSet is allowed. The default for this bit is 0b; allow FLASH image updates using PROSet.
4	DLWS	Disable Legacy Wakeup Support. If set to 1b, no changes to the Legacy OS Wakeup Support menu option is allowed. The default for this bit is 0b; allow Legacy OS Wakeup Support menu option changes.
3	DBS	Disable Boot Selection. If set to 1b, no changes to the boot order menu option is allowed. The default for this bit 0b; allow boot order menu option changes.

Table 5-17. Boot Agent Configuration Customization Options (Word 31h)

Bit	Name	Description
2	DPS	Disable Protocol Select. If set to 1b, no changes to the boot protocol is allowed. The default for this bit is 0b; allow changes to the boot protocol.
1	DTM	Disable Title Message. If set to 1b, the title message displaying the version of the boot agent is suppressed; the Control-S message is also suppressed. This is for OEMs who do not wish the boot agent to display any messages at system boot. The default for this bit is 0b; allow the title message that displays the version of the boot agent and the Control-S message.
0	DSM	Disable Setup Menu. If set to 1b, no invoking the setup menu by pressing Control-S is allowed. In this case, the EEPROM can only be changed via an external program. The default for this bit is 0b; allow invoking the setup menu by pressing Control-S.

5.6.32 Boot Agent Configuration Customization Options (Word 32h)

Word 32h is used to store the version of the boot agent that is stored in the FLASH image. When the Boot Agent loads, it can check this value to determine if any first-time configuration needs to be performed. The agent then updates this word with its version. Some diagnostic tools to report the version of the Boot Agent in the FLASH also read this word. This word is only valid if the PPB is set to 0b. Otherwise the contents might be undefined.

Table 5-18. Boot Agent Configuration Customization Options (Word 32h)

Bit	Name	Description
15:12	MAJOR	PXE boot agent major version. The default for these bits is 0b.
11:8	MINOR	PXE boot agent minor version. The default for these bits is 0b.
7:0	BUILD	PXE boot agent build number. The default for these bits is 0b.

5.6.33 IBA Capabilities (Word 33h)

Word 33h is used to enumerate the boot technologies that have been programmed into the FLASH. It is updated by IBA configuration tools and is not updated or read by IBA.

Table 5-19. IBA Capabilities

Bit	Name	Description
15:14	SIG	Signature. These bits must be set to 1b to indicate that this word has been programmed by the agent or other configuration software.
13:5	Reserved	Reserved for future use. Set these bits to 0b.
4	SAN	SAN capability is present in FLASH. 0b = The SAN capability is not present (default). 1b = The SAN capability is present.
3	EFI	EFI UNDI capability is present in FLASH. 0b = The RPL code is not present (default). 1b = The RPL code is present.
2	RPL	RPL capability is present in FLASH. 1b = The RPL code is present (default). 0b = The RPL code is not present.
1	UNDI	PXE/UNDI capability is present in FLASH. 1b = The PXE base code is present (default). 0b = The PXE base code is not present.
0	BC	PXE base code is present in FLASH. 0b = The PXE base code is present (default). 1b = The PXE base code is not present.

5.6.34 IBA Secondary Port Configuration (Words 34h-35h)

These words provide a unique configuration for the second port of the **82546GB/EB**. The format is the same as that used in words 30h and 31h for LAN A.

5.6.35 Checksum Word Calculation (Word 3Fh)

The Checksum word (3Fh) should be calculated such that after adding all the words (00h-3Fh), including the Checksum word itself, the sum should be BABAh. The initial value in the 16-bit summing register should be 0000h and the carry bit should be ignored after each addition. This checksum is not accessed by the Ethernet controller. If CRC checking is required, it must be performed by software.

Note: Hardware does not calculate checksum word 3Fh during EEPROM write; it must be calculated by software independently and included in the EEPROM write data. Hardware does not compute a checksum over words 00h-0Fh during EEPROM reads in order to determine the validity of the EEPROM image; this field is provided strictly for software verification of EEPROM validity. All hardware configuration based on word 00h-0Fh content is based on the validity of the Signature field of EEPROM Initialization Control Word 1. Signature must be 01b.

5.6.36 82546GB/EB Dual-Channel Fiber Wake on LAN (WOL) Mode and Functionality (Word 0Ah, 20h)

Four bits and two words determine dual-channel fiber WOL mode and functionality. In addition to the power management bits, one of the SDP's must be set in order for the "A" laser to remain on when the system goes into D3. Three states are defined for the EEPROM image:

- Never - The Ethernet controller has WOL disabled and cannot be put into WOL mode.
- Possible - The Ethernet controller can be WOL enabled, but currently has the feature turned off. This is the normal shipping configuration for WOL-capable server adapter cards.
- On - The Ethernet controller is set with WOL functionality enabled.

Table 5-20. WOL Mode and Functionality (Word 0Ah)

State	Bit 2	Bit 7	Resulting Word
Never	0	0	4C03
Possible	1	0	4C0B
On	1	1	4C2B

Table 5-21. WOL Mode and Functionality (Word 20h)

State	Bit 2	Bit 7	Resulting Word
Never	0	0	C109
Possible	1	0	C10D
On	1	1	C18D

5.6.37 EEPROM Images

Refer to the appropriate Ethernet controller's EEPROM map and programming information for sample EEPROM images.

5.7 Parallel FLASH Memory

All Ethernet controllers except the **82540EP/EM** provide an external parallel interface to an optional FLASH or boot EEPROM device. Accesses to the FLASH memory are controlled by the Ethernet controllers, but are accessible to host software as normal PCI reads or writes to the FLASH memory mapping range. Software developers can also map FLASH memory to I/O space. The Ethernet controllers support 8-bit wide parallel FLASH memory up to 4 Mb (512 KB); an appropriate size for typical applications would be 1 Mb (128 KB). The size of the FLASH implemented in the design can be encoded into bits in the EEPROM. FLASH and expansion ROM base address registers are reconfigured based on these EEPROM settings.

Representative FLASH memory devices that have been found to work satisfactorily with the Ethernet controllers are listed in Table 5-22:

Table 5-22. FLASH Memory Manufacturers

Manufacturer	Manufacturer's Part Number
Atmel	AT49LV010 AT49BV002AN-70J1
Silicon Storage Technology	SST39V512 39VF020-90-4I-NH 39VF020-70-4C-NH

The FLASH memory interface trace routing is not critical because the interface runs at a very slow speed. In a tightly space-constrained design, the FLASH memory device is a good choice for placement in relative isolation from the Ethernet controllers.

Note: The **82540EP/EM** provides an external interface to a serial FLASH or Boot EEPROM device. See Appendix B for more information.

Note: This page intentionally left blank.

FLASH Memory Interface 7

All Ethernet controllers (except the **82540EP/EM**) provide an external parallel interface to a FLASH, or boot ROM, device such as the Atmel AT49LV010[1]. All accesses to this device are managed by the Ethernet controller and are accessible to software as normal PCI reads or writes to the FLASH memory mapping range. The Ethernet controller supports parallel FLASH devices with up to 4 Mb (512 KB) of memory. The size of the FLASH implemented with the Ethernet controller can be encoded into bits in the EEPROM. The FLASH and Expansion ROM Base Address Registers are reconfigured based on these EEPROM settings.

Note: Though the Ethernet controller supports devices with up to 512 KB of memory, smaller devices can also be used. Accesses to memory beyond the FLASH device size results in access wrapping as only the lower address bits are utilized by the FLASH.

7.1 FLASH Interface Operation

The FLASH is read from, or written to, each time the host processor performs a read or a write operation to a memory location that is within the FLASH address mapping or upon boot through accesses in the space indicated by the Expansion ROM Base Address Register (see Section 4.1). All accesses to the FLASH, except read accesses, require the appropriate command sequence for the device used. Refer to the specific FLASH data sheet for more details on reading from or writing to FLASH. Accesses to the FLASH are based on a direct decode of processor accesses to a memory window defined in either the Ethernet controller's FLASH Base Address Register (PCI Control Register at offset 14h or 18h) or the Expansion ROM Base Address Register (PCI Control Register at offset 30h).

FLASH accesses must always be assembled or disassembled by the Ethernet controller to or from the FLASH whenever the access is greater than a byte-wide access. Due to slow access times to a typical FLASH, word (32-bit) accesses are not recommended for any cycles that occur after system initialization in order to avoid violating PCI hold specifications. Boot ROM shadowing is an exception to the 16 clock rule and can incur in excess of 25 wait states. The Ethernet controller byte reads to the FLASH take approximately of 256 ns. The Ethernet controller issues retry accesses during this time.

7.2 FLASH Control and Accesses

Write control of the FLASH is controlled by the FWE bits in the EEPROM/FLASH Control and Data Register (EECD.FWE). See Section 13.4.3 for details.

Processor accesses to the FLASH are very slow. The Ethernet controller always issues a target-disconnect at the first data cycle. The Ethernet controller asserts the STOP# signal to indicate the target-disconnect. Section 7.2.1 and Figure 7.2.2 show read and write accesses to the FLASH. Note that burst accesses to the FLASH address space are not allowed.

1. The **82540EP/EM** provides an external interface to a serial FLASH or Boot EEPROM device. See Appendix B for more information.

7.2.1 Read Accesses

Upon reads to the FLASH address space, the Ethernet controller uses the TRDY# signal to insert target wait states until valid data can be read from the FLASH device and presented on the data lines. When TRDY# is asserted, the Ethernet controller drives valid data on the data lines. The processor master can then complete normal data read cycle by asserting IRDY# when it is ready.

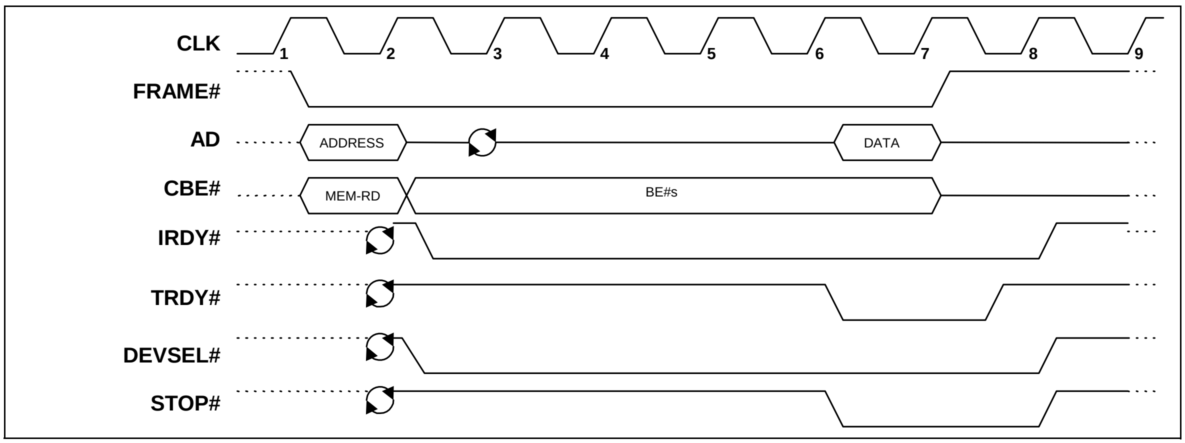


Figure 7-1. FLASH Buffer Read Cycles

7.2.2 Write Accesses

The processor, as the initiator, drives the address lines AD[63:0], the command and byte enable lines [7:0]#, and the control lines IRDY# and FRAME#. It also provides the Ethernet controller with valid data immediately after asserting TRDY#. The Ethernet controller controls the TRDY# signal and deasserts it for a certain number of clocks until valid data is written to the FLASH buffer. By asserting TRDY#, the Ethernet controller signals the processor that the current data phase has completed.

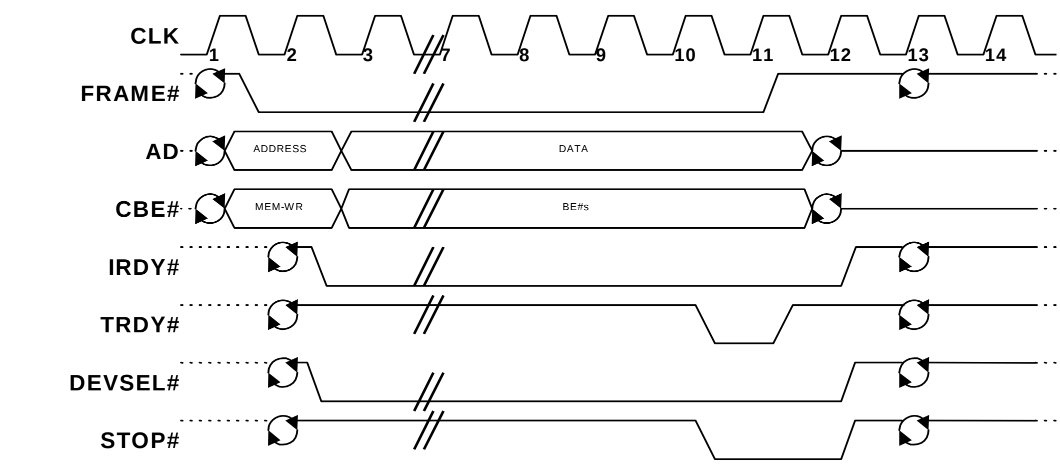


Figure 7-2. FLASH Buffer Write Cycle

Note: This page is intentionally left blank.

Power Management 6

6.1 Introduction to Power Management

The PCI/PCI-X Family of Gigabit Ethernet Controllers support the Advanced Configuration and Power Interface (ACPI) specification as well as Advanced Power Management (APM). This section describes how Power Management is implemented in the Ethernet controllers.

Note: The **82541ER** does not support ACPI or APM wakeup.

Power Management can be disabled via bits in the Initialization Control Word 2 which is loaded from the EEPROM during power-up reset. See the EEPROM description in Section 5.6.12 for further details. Even when disabled, the Power Management register set is still present.

The Ethernet controller supports the following Power Management related features:

- Power states of D0 & $D3_{hot}$ with optional $D3_{cold}$ support
- Power(D3) < Power(D0)
- Wakeup

6.2 Assumptions

The following assumptions apply to the implementation of Power Management for the Ethernet controller.

- Any time LAN_PWR_GOOD is asserted all power supplies are stable, RST# is stable, and the clock is stable.
- Prior to transition from D0 to the D3 state, the operating system ensures that the software device driver is disabled and all pending bus transactions are complete or cleanly terminated.
- The driver sets up the filters prior to the system transitioning the Ethernet controller to the D3 state.
- The system never deactivates the PCI clock in 66 MHz PCI mode or any PCI-X[1] mode without asserting RST#.
- No wakeup capability, except APM Wakeup if enabled in the EEPROM, is required after the system asserts, then de-asserts RST#.
- No wakeup capability, except APM Wakeup if enabled in the EEPROM, is required after the system puts the Ethernet controller in the D3 state and then returns it to D0.
- If the *APMPME* bit (bit 3) in the Wakeup Control Register (WUC.APMPME) is set to 1b, it is permissible to assert PME# even when *PME_En* is 0b.
- The deassertion (rising) edge of RST# puts the Ethernet controller in the D0u state.

1. The **82540EP/EM**, **82541xx**, and **82547GI/EI** do not support PCI-X mode.

6.3 $D3_{cold}$ support

If the AUX pin is connected to logic 1b, the Ethernet controller advertises $D3_{cold}$ Wakeup support. The amount of power required for this function (which includes the entire Ethernet port circuitry) is advertised in the Power Management Data Register which is loaded from the EEPROM.

If $D3_{cold}$ is supported, the *PME_En* and *PME_Status* bits of the Power Management Control/Status Register (PMCSR), as well as their shadow bits in the Wakeup Control Register (WUC) are not reset by RST#. If $D3_{cold}$ Wakeup is not supported, PMCSR and WUC is reset on the deassertion (rising edge) of RST#.

The only effect of setting AUX to 1b is advertising $D3_{cold}$ Wakeup support and changing the reset function of *PME_En* and *PME_Status*. The **82541PI/GI** can enter a fully-disabled low-power state in $D3_{cold}$ if an enable bit is set in the EEPROM. All remaining Ethernet controllers do nothing different in $D3_{cold}$ compared to $D3_{hot}$. AUX_POWER is level sensitive, and any changes are immediately reflected in the $D3_{cold}$ Wakeup advertisements and the *PME_En* and *PME_Status* reset function.

6.3.1 Power States

The Ethernet controller supports D0 and D3 power states defined in the PCI Power Management Specification. D0 is divided into two sub-states: D0u, and D0a. In addition, it supports a Dr state that is entered when RST# is asserted. Dr behaves the same as D3 except that the PCI bus is isolated. Figure 6-1 illustrates the power states and the conditions that cause transitions from state to state.

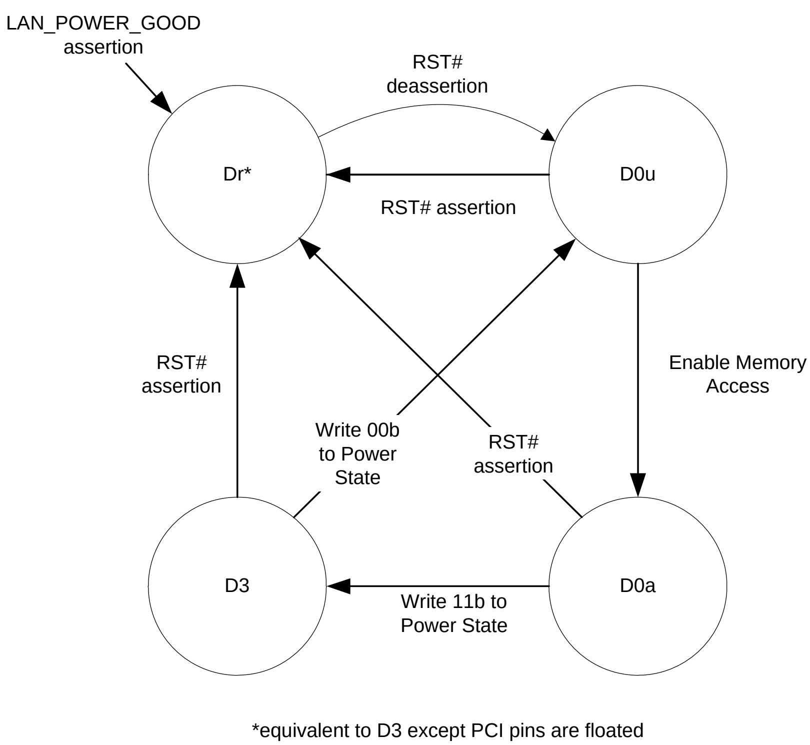


Figure 6-1. Power State Transitions

6.3.1.1 Dr State

At initial boot-up, once LAN_PWR_GOOD is asserted, the Ethernet controller reads the EEPROM. If the *APM Mode* bit in the EEPROM's Initialization Control Word 2 is set then APM Wakeup is enabled.

The system may maintain RST# asserted for an arbitrary time. During this time, and for up to 1 ms afterwards, the Ethernet controller does not assert any PCI signals except PME#.

During operation, the system may assert RST# at any time. In particular, if the system wishes an Ethernet controller to enter the D3cold state it must assert RST# before dropping main power. Any time RST# is asserted, the Ethernet controller transitions to the Dr state. It also floats all PCI signals except PME# and remains in the "reset" state until no more than 1 ms after the deassertion of RST#.

Internally, the Ethernet controller treats the reset state equivalently to D3. Any Wakeups enabled before entering reset is maintained. For power savings, the Ethernet controller shuts down some internal clocks and registers and deasserts PWR_STATE1. If Wakeup is not enabled, the Ethernet controller also deasserts PWR_STATE0. As a result, the Ethernet controller won't transmit any frames in Dr state or send idles in TBI mode (**82544GC/EI**)/internal SerDes (**82546GB/EB** and **82545GM/EM**)[1].

The deassertion (rising edge) of RST# causes a transition to D0u.

6.3.1.2 D0u State

The D0u state is a low-power state used after RST# is deasserted, or when coming out of D3, but before the Ethernet controller is initialized.

When entering D0u, the Ethernet controller disables Wakeups, resets the PHY, and then re-reads the EEPROM. If the *APM Mode* bit in the EEPROM's Initialization Control Word 2 is set, then APM Wakeup is enabled.

Internally, D0u is treated like D3 and some internal clocks and registers are shut down. The D0u state is exited when the system enables memory space access to the Ethernet controller by writing a 1b to the *Memory Access Enable* bit of the PCI Command Register.

Note: In order for hardware to transition from D3 to the D0 state properly, BIOS should not alter the *Memory Access Enable* or the *I/O Access Enable* bit of the PCI Command Register. Also, the PCI configuration space must be programmed when hardware transitions out of D3 to D0.

1. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

6.3.1.3 D0a (D0 active)

Once memory space is enabled, all internal clocks are activated, the Ethernet controller enters an active state, and can then transmit and receive packets if properly configured by the software driver. The controller also signals the PHY (if using the internal PHY) to indicate full speed/ power[1]. If APM Wakeup was activated it remains active. The software driver can deactivate APM Wakeup by writing to the Wakeup Control Register (WUC), or activate other Wakeup Filters by writing to the Wakeup Registers.

6.3.1.4 D3

Prior to transition from D0 to the D3 state, the software driver must ensure the Ethernet controller transmit and receive functions have been disabled and all pending bus transactions are complete or cleanly terminated. If Wakeup capability is needed, the software driver needs to set up the appropriate Wakeup registers and the system needs to write a 1b to the *PME_En* bit of the Power Management Control / Status Register (PMCSR) prior to the transition to D3.

When the system writes a 11b to the *PowerState* field of the Power Management Control/Status Register (PMCSR) the Ethernet controller transitions to D3. Any Wakeups that are enabled remain enabled. Upon transitioning to D3 the Ethernet controller clears the *Memory Access Enable* or the *I/O Access Enable* bit of the PCI Command Register, which disables memory access decode. In D3, the Ethernet controller only responds to PCI configuration accesses. It won't generate master cycles, transmit any frames on the TBI/internal SerDes[2]/internal PHY interface, or transmit idles in TBI mode/internal SerDes if Wakeup is enabled.

For power savings the Ethernet controller shuts down some internal clocks and registers.

To transition back to D0u, the system writes a 00b to the *Power State* field of the Power Management Control/Status Register (PMCSR).

6.3.2 Timing

The following sections give detailed timing for the state transitions. In the diagrams the dotted connecting lines represent the Ethernet controller's requirements, while the solid connecting lines represent the Ethernet controller's guarantees.

Note: The following timing diagrams are not to scale. The clocks edges are shown to indicate running clocks only and are not used to indicate the actual number of cycles for any operation.

If CLK_RUN# functionality is enabled in the EEPROM, then the **82541PI/GI/EI** and **82540EP** Ethernet controllers assert the CLK_RUN# pin when it requires the PCI clock. Otherwise, the clock is not required and the system might shut the PCI clock off.

1. Not applicable to the **82541xx** or **82547GI/EI**.
2. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

6.3.2.1 Power Up (Off to Dr to D0u to D0a)

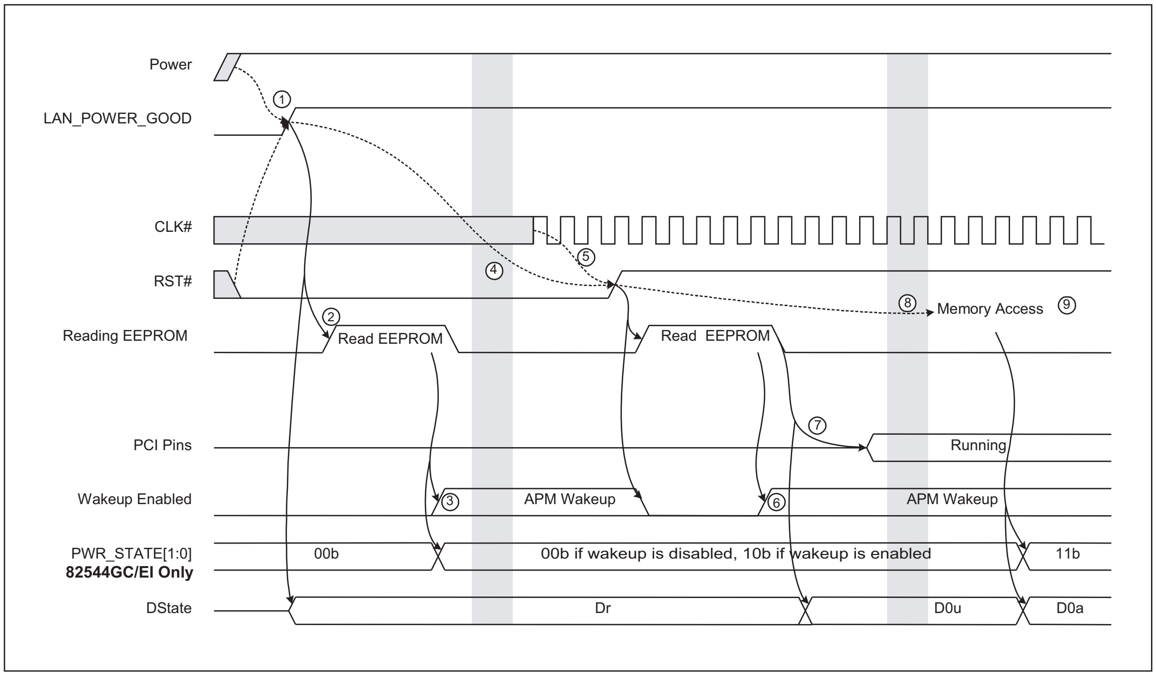


Figure 6-2. Startup Timing

Diagram #	Notes
1	LAN_PWR_GOOD must not be asserted until all power supplies are good and the clock is stable.
2	An EEPROM read starts on the rising edge of LAN_PWR_GOOD and RST#.
3	APM Wakeup mode can be enabled based on what is read from the EEPROM.
4	The system can delay an arbitrary time before deasserting RST#.
5	The PCI 2.2 or 2.3 specification requires the clock to be active 100 µs before deasserting RST#. ($T_{clk\text{-}rst}$ parameter)
6	The deassertion edge of RST# causes the EEPROM to be re-read and Wakeup disabled.
7	Synchronizing the clock generators and circuit adjustments require up to 512 PCI clocks before the Ethernet controller drives PCI signals and responds to PCI transactions.
8	The system can delay an arbitrary time before enabling Memory Access.
9	Writing a 1b to the *Memory Access Enable or I/O Access Enable* bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state. For the **82544GC/EI**, writing a 1b to the *Memory Access Enable or I/O Access Enable* bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state and asserts both PWR_STATE outputs.

6.3.2.2 Transition From D0a to D3 and Back Without PCI Reset

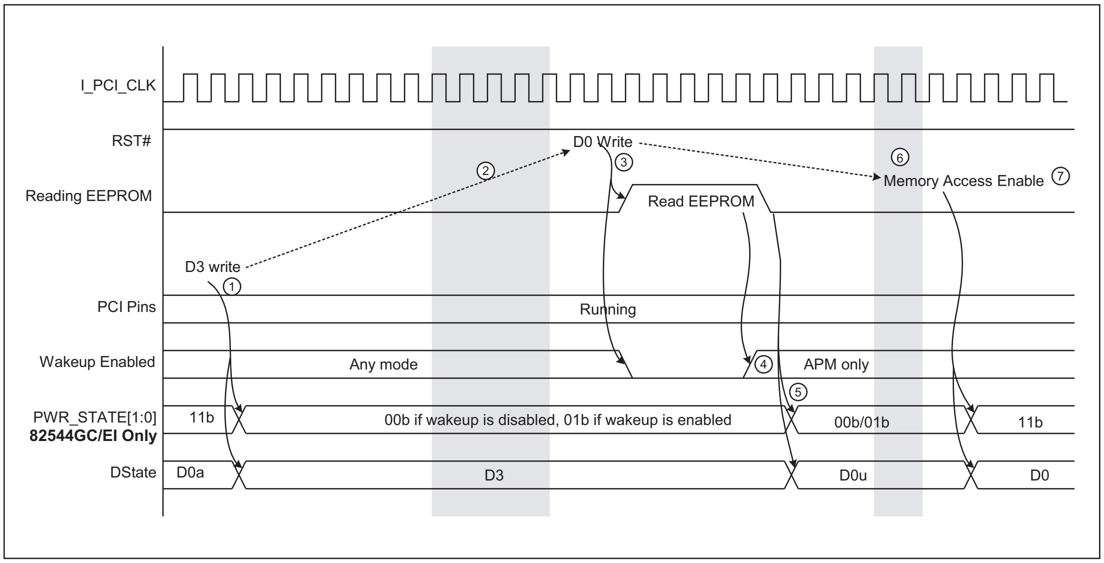


Figure 6-3. Transition from D0a to D3 and Back Without PCI Reset

Diagram #	Notes
1	Writing a 11b to the *Power State* field of the Power Management Control/Status Register (PMCSR) transitions the Ethernet controller to D3.
2	The system can keep the Ethernet controller in D3 state for an arbitrary amount of time.
3	To exit D3 state the system writes 00b to the *Power State* field of the Power Management Control/Status Register (PMCSR).
4	APM Wakeup mode can be enabled based on what is read in the EEPROM.
5	For the **82544GC/EI**, PWR_STATE[1:0] is set to 01b if APM Wakeup is enabled, 00b otherwise.
6	The system can delay an arbitrary time before enabling memory access.
7	Writing a 1b to the *Memory Access Enable or I/O Access Enable* bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state. For the **82544GC/EI**, writing a 1b to the *Memory Access Enable or I/O Access Enable* bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state and asserts both PWR_STATE outputs.

6.3.2.3 Transition From D0a to D3 and Back with PCI Reset

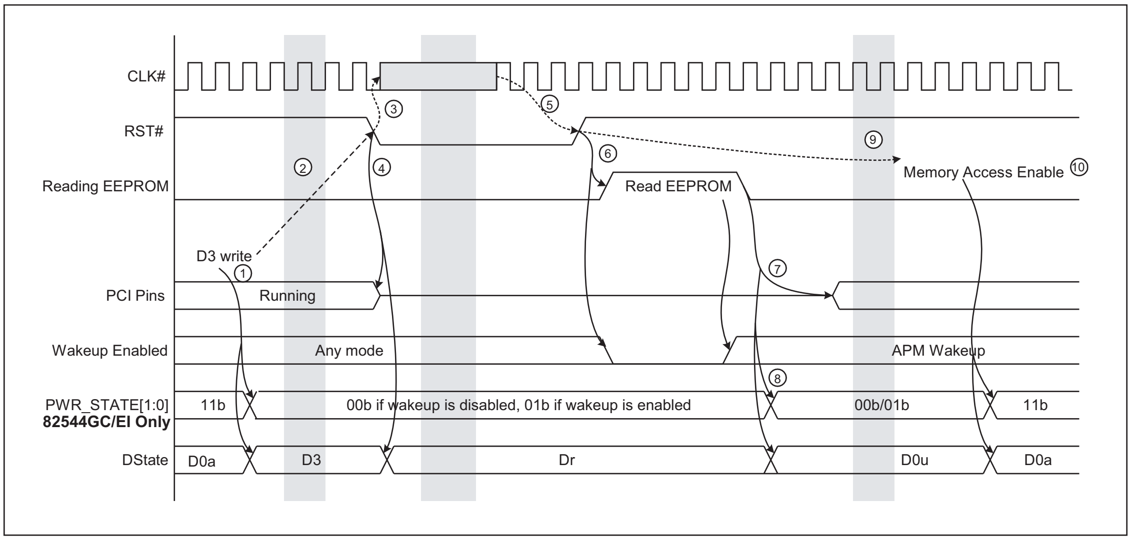


Figure 6-4. Transition From D0a to D3 and Back with PCI Reset

Diagram #	Notes
1	Writing a 11b to the *Power State* field of the Power Management Control/Status Register (PMCSR) transitions the Ethernet controller to D3.
2	The system can delay an arbitrary amount of time between setting D3 mode and asserting RST#.
3	In 66 MHz or PCI-X[a] modes the system must assert RST# before stopping the PCI clock. It may assert RST# without stopping the clock. For the **82541PI/GI/EI** and **82540EP**, If CLK_RUN# is enabled, then they do not require a continuous clock during this time, but does require that the system drive the clock in response to CLK_RUN# assertion.
4	Upon assertion of RST# the Ethernet controller floats all PCI pins except PME# and goes to “Dr” state.
5	In 66 MHz or PCI-X modes, the PCI 2.2 and 2.3 specification requires the system to start the PCI clock 100 µs before deassertion of RST#. In 33 MHz systems the PCI clock can start and stop at any time independent of RST#.
6	The deassertion edge of RST# causes the EEPROM to be re-read and Wakeup disabled.
7	Synchronizing the clock circuits and circuit adjustments require up to 512 PCI clocks before the Ethernet controller drives PCI signals and responds to PCI transactions.
8	For the **82544GC/EI**, O_PWR_STATE is set to 01b if APM Wakeup is enabled, 00b otherwise.
9	The system can delay an arbitrary time before enabling memory access.
10	Writing a 1b to the *Memory Access Enable or I/O Access Enable* bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state. For the **82544GC/EI**, writing a 1b to the *Memory Access Enable or I/O Access Enable* bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state and asserts both PWR_STATE outputs.

a. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

6.3.2.4 PCI Reset Without Transition to D3

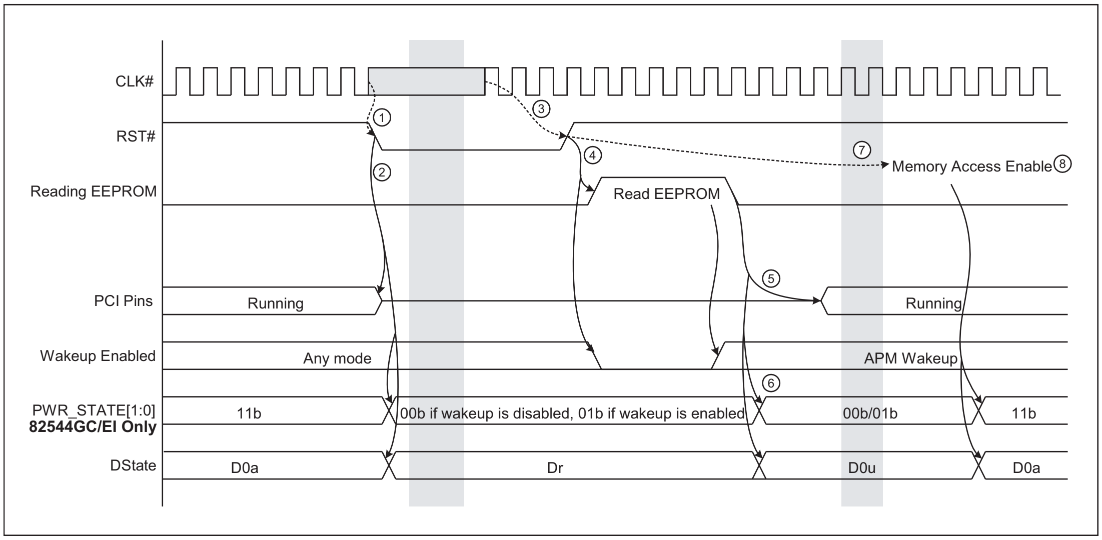


Figure 6-5. PCI Reset Sequence

Diagram #	Notes
1	In 66 MHz or PCI-X[a] modes, the system must assert RST# before stopping the PCI clock. It may assert RST# without stopping the clock.
2	Upon assertion of RST# the Ethernet controller floats all PCI pins except PME# and goes to “Dr” state.
3	In 66 MHz or PCI-X modes the system must assert RST# before stopping the PCI clock. It may assert RST# without stopping the clock. For the **82541PI/GI/EI** and **82540EP**, If CLK_RUN# is enabled, then they do not require a continuous clock during this time, but does require that the system drive the clock in response to CLK_RUN# assertion.
4	The deassertion edge of RST# caused the EEPROM to be re-read and Wakeup disabled.
5	Synchronizing the clock circuits and circuit adjustments require up to 512 PCI clocks before the Ethernet controller drives PCI signals and responds to PCI transactions.
6	For the **82544GC/EI**, PWR_STATE[1:0] is set to 01b if APM Wakeup is enabled, 00b otherwise.
7	The system can delay an arbitrary time before enabling memory access.
8	Writing a 1b to the *Memory Access Enable or I/O Access Enable* bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state. For the **82544GC/EI**, writing a 1b to the *Memory Access Enable or I/O Access Enable* bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state and asserts both PWR_STATE outputs.

a. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

6.3.3 PCI Power Management Registers

Power Management registers are part of the capabilities linked list pointed to by the Capabilities Pointer (Cap_Ptr) in the PCI configuration space. Refer to Section 4.1.

All fields are reset by LAN_PWR_GOOD. All of the fields except *PME_En* and *PME_Status* are reset by the deassertion (rising edge) of RST#. If AUX_POWER = 0b, the *PME_En* and *PME_Status* fields also reset by the deassertion (rising edge) of RST#.

The following table lists the organization of the PCI Power Management Register Block:

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
DCh	Power Management Capabilities (PMC)		Next Item Ptr	Capability ID
E0h	Data	PMCSR_BSE Bridge Support Extensions	Power Management Control / Status Register (PMCSR)	

The following sections describe the register definitions, whether they are required or optional for compliance, and how they are implemented in the Ethernet controller. Complete details can be found in the PCI Power Management Interface specification.

Note: The offset indicated is the byte-offset from the position indicated by Cap_Ptr in the Configuration Space Header.

6.3.3.1 Capability ID
1 Byte Offset = 0 (RO)

Bits	Default	R/W	Description
07:00	01h	Read Only	**ID** – The Ethernet controller returns a value of 01h for this field, indicating the linked list item as being the PCI Power Management Registers.

6.3.3.2 Next Item Pointer
1 Byte Offset = 1 (RO)

Bits	Default	R/W	Description
07:00	E4h	Read Only	**Next Item Pointer** - This field provides an offset into the function's PCI Configuration Space pointing to the location of next item in the function's capability list. Its value of E4h points to the PCI-X[a] capability.

a. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

6.3.3.3 Power Management Capabilities - (PMC) 2 Bytes Offset = 2 (RO)

Bits	Default	R/W	Description
15:11	See text	Read Only	**PME_Support** – This 5-bit field indicates the power states in which the function may assert PME#[a]. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state. bit (11) (XXXX1)b – PME# can be asserted from D0 bit (12) (XXX1X)b – PME# can be asserted from D1 bit (13) (XX1XX)b – PME# can be asserted from D2 bit (14) (X1XXX)b – PME# can be asserted from $D3_{hot}$ bit (15) (1XXXX)b – PME# can be asserted from $D3_{cold}$ If Power Management is not disabled in the EEPROM, the Ethernet controller supports PME# generation from D0 and $D3_{hot}$ states. If Power Management is not disabled and AUX_POWER = 1b, the Ethernet controller also supports the $D3_{cold}$ state. Condition — Value 00000b — Power Management disabled in EEPROM AUX_POWER = 01001b — Power Management enabled, AUX_POWER = 11001b — Power Management enabled,
10	0b	Read Only	**D2_Support** - If this bit is set to 1b, supports the D2 Power Management State. The Ethernet controller returns a value of 0b for this bit indicating that it does not support D2 and cannot handle the PCI clock stopping in PCI 66 MHz mode (or PCI-X[b] mode) without RST# being asserted.
09	0b	Read Only	**D1_Support** - If this bit is set to 1b, supports the D1 Power Management State. The Ethernet controller returns a value of 0b for this bit indicating that it does not support D1.
08:06	000b	Read Only	**AUX Current** – Specifies the auxiliary power current required for PME# generation from $D3_{cold}$ if the Data Register is not implemented.
05	1b	Read Only	**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. The Ethernet controller returns a value of 1b for this bit indicating that it's device driver must be executed following transition to the D0 uninitialized state.
04	0b	Read Only	**Reserved**
03	Loaded from EEPROM	Read Only	**PME_Clock** - When this bit is a 1b it indicates that the function relies on the presence of the PCI clock for PME# operation. The controller loads this bit from the EEPROM. Otherwise, it returns a 0b.
02:00	010b	Read Only	**Version** - A value of 010b indicates that this function complies with the Revision 1.1 of the PCI Power Management Interface Specification.

a. Not applicable to the **82541ER**.
b. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

6.3.3.4 Power Management Control / Status Register - (PMCSR) 2 Bytes Offset = 4 (RO)

Bits	Default	R/W	Description
15	0b (see description)	Read/ Write 1b to clear	***PME_Status*** – This bit is set when the function would normally assert the PME# signal independent of the state of the *PME_En* bit. The Ethernet controller returns a value of 1b for this bit if a Wakeup condition has been detected. Writing a 1b clears this bit and deasserts PME#[a]. If the AUX_POWER input is 1b, the *PME_Status* field is only reset by LAN_PWR_GOOD. If AUX_POWER is 0b, *PME_Status* is also reset on the deassertion (rising edge) of RST#.
14:13	00b 01b if Manageability is enabled (see description)	Read Only	**Data_Scale** - This 2- bit read-only field indicates the scaling factor to be used when interpreting the value of the Data register. This field outputs 01b (to indicate units of 0.1 watt) when Manageability is enabled in the EEPROM and the *Data_Select* field is set to 0, 3, 4, or 7, and 00b otherwise.
12:09	0000b	Read/ Write	**Data_Select** - This 4-bit field is used to select which data is to be reported through the Data register and *Data_Scale* field. These bits are only writable when Power Management is enabled via EEPROM.
08	0b on Power-On reset	Read/ Write	***PME_En*** – If Power Management is not disabled in the EEPROM, writing a 1b to this register enables Wakeup and causes the Ethernet controller to assert PME# when it receives a Wakeup event enabled in the Wakeup Filter Control Register (WUFC). **Note:** This bit cannot be set for the **82541ER**. If Power Management is disabled in the EEPROM, writing a 1b to this bit has no affect, and does not set the bit to 1b. If the AUX_POWER input is 1b, the *PME_En* field is only reset by LAN_PWR_GOOD. If AUX_POWER is 0b, it is also reset on the deassertion (rising edge) of RST#. **Note:** If APM Wakeup is enabled, the PME# pin can be asserted even if *PME_En* is 0b. See Section 6.4.1 for details.
07:02	000000b	Read Only	**Reserved** - The Ethernet controller returns a value of 000000b for this field.
01:00	00b	Read/ Write	**PowerState** - This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. The definition of the field values is as follows: 00b - D0 01b - D1 (ignored if written with this value) 10b - D2 (ignored if written with this value) 11b - D3 If software attempts to write an unsupported state to this field, 00b or 10b, or if Power Management is disabled in the EEPROM, then the Ethernet controller completes the write operation normally on the bus, however the data is discarded and no state change occurs. These bits are cleared and the power state is returned to D0 after the trailing edge of RST#.

a. Not applicable to the **82541ER**.

This register is used to control and monitor power management events in the Ethernet controller. If auxiliary power is present, as indicated by AUX_POWER = 1b, a PCI reset does not clear *PME_En* and *PME_Status*.

6.3.3.5 PMCSR_BSE Bridge Support Extensions
1 Byte Offset = 6 (RO)

This register indicates support for PCI bridge specific functions. Note that these functions are not implemented in the Ethernet controller and the values are set to 00h.

6.3.3.6 Data Register
1 Byte Offset = 7 (RO)

Bits	Default	R/W	Description
07:00	00h (loaded from EEPROM)	Read Only	Data returned. See the following explanation.

This register is used to report power consumption and heat dissipation. Its value and meaning is determined by the value programmed in the *Data_Select* field of the Power Management Control/ Status Register (PMCSR).

Data Select	Meaning
0	D0 Power Consumed
1	D1 Power Consumed
2	D2 Power Consumed
3	D3 Power Consumed
4	D0 Power Dissipated
5	D1 Power Dissipated
6	D2 Power Dissipated
7	D3 Power Dissipated
8	Common power consumption of multi-function devices
9-15	Reserved

The units are defined by the *Data_Scale* field of the Power Management Control/Status Register.

Data Select	Meaning
0	Unknown (used for unsupported states)
1	0.1 Watts (used by Ethernet controller for supported states)
2	0.01 Watts
3	0.001 Watts

If power management is disabled in the EEPROM, then the data register always reads 0b.

If power management is not disabled and when the *Data_Select* field is programmed to 0 or 4, the Ethernet controller sets the Data Register to the *D0 Power* value in the EEPROM. When the *Data_Select* field is programmed to 3 or 7, the Ethernet controller sets the Data Register to the *D3 Power* value in the EEPROM. Otherwise it returns 0b.

6.4 Wakeup

The Ethernet controller supports two types of wakeup mechanisms:

- Advanced Power Management (APM) Wakeup
- ACPI Power Management Wakeup

Note: The **82541ER** contains power management logic, but is not spec-compliant, because it does not assert PME# for Magic Packets, Network Wakeup Packets, or link change status.

The ACPI Power Management Wakeup uses the PME# pin to wake up the system. The Advanced Power Management Wakeup uses the PME# pin.

6.4.1 Advanced Power Management Wakeup

"Advanced Power Management Wakeup", or "APM Wakeup", was previously known as "Wake on LAN". The basic premise is to receive a broadcast or unicast packet with an explicit data pattern, and then to assert a signal to wake up the system. In the earlier generations of the Ethernet controller, this was accomplished by using special signal. The Ethernet controller would assert the signal for approximately 50 ms to signal a wakeup. The **82544GC/EI** uses the APM_WAKEUP pin for this function. For the remaining Ethernet controllers, the PCI PME# signal has been used to wake up the system.

On power-up, the Ethernet controller reads the *APM Enable* bits from the EEPROM Initialization Control Word 2 into the *APM Enable* (APME) bits of the Wakeup Control Register (WUC). These bits control enabling of APM Wakeup.

When APM Wakeup is enabled, the Ethernet controller checks all incoming packets for "Magic Packets". See Section 6.4.3.1.4 for a definition of "Magic Packets*".

Once the Ethernet controller receives a matching magic packet, it:

- Sets the *PME_Status* bit in the Power Management Control / Status Register (PMCSR) and asserts PME#. If the *Assert PME On APM Wakeup (APMPME)* bit is set in the Wakeup Control Register (WUCR).
- Stores the first 128 bytes of the packet in the Wakeup Packet Memory (WUPM).
- Sets the *Magic Packet Received* bit in the Wakeup Status Register (WUS).
- Sets the packet length in the Wakeup Packet Length Register (WUPL).
- Asserts PME# until the driver clears the *Magic Packet Received AMAG* bit in the Wakeup Status Register (WUS), the driver clears the *Assert PME On APM Wakeup* (APMPME) bit in the Wakeup Control Register (WUC), or the driver disables APM Wakeup.
- For the **82544GC/EI** only, asserts APM_WAKEUP for 50 ms. For purposes of APM_WAKEUP assertion, the **82544GC/EI** ignores any additional magic packets received during that 50 ms. If the **82544GC/EI** receives another magic packet afterwards, it reasserts APM_WAKEUP for another 50 ms.

- Maintains the first magic packet received in the Wakeup Packet Memory (WPM) until the driver writes a 0b to the *Magic Packet Received* MAG bit in the Wakeup Status Register (WUS).

"APM Wakeup" is supported in all power states and only disabled if a subsequent EEPROM read results in the *APM Wakeup* bit being cleared or software explicitly writes a 0b to the *APM Wakeup (APM)* bit of the WUC register.

6.4.2 ACPI Power Management Wakeup

The Ethernet controller supports ACPI Power Management based wakeups. It generates system wakeup events from three sources:

- Reception of a "Magic Packet".
- Reception of a Network Wakeup Packet.
- Detection of a link change of state.

Note: The **82541ER** does not support ACPI wakeup events.

Activating ACPI Power Management Wakeup requires:

- The software driver to program the Wakeup Filter Control Register (WUFC). This indicates the packets the driver wishes to wake up and supplies the necessary data to the IP Address Table (IPAT) and the Flexible Filter Mask Table (FFMT), Flexible Filter Length Table (FFLT), and the Flexible Filter Value Table (FFVT). The driver can also set the *Link Status Change Wakeup Enable* (LNKC) bit in the Wakeup Filter Control Register (WUFC) to cause wakeup when the link changes state.
- The OS to write a 1b to the *Pme_En* bit of the Power Management Control / Status Register (PMCSR).

Normally, after enabling wakeup, the OS sets the Ethernet controller to D3 (low-power mode).

Once wakeup is enabled, the Ethernet controller monitors incoming packets, first filtering them according to its standard address filtering method, then filtering them with all of the enabled wakeup filters. If a packet passes both the standard address filtering and at least one of the enabled wakeup filters, the Ethernet controller:

- Sets the *PME_Status* bit in the Power Management Control / Status Register (PMCSR)
- Assert PME#. If the *PME_En* bit in the Power Management Control / Status Register (PMCSR) is set.
- Stores the first 128 bytes of the packet in the Wakeup Packet Memory.
- Sets one or more of the "Received" bits in the Wakeup Status Register (WUS). Note that the Ethernet controller sets more than one bit if a packet matches more than one filter.
- Sets the packet length in the Wakeup Packet Length Register (WUPL).

If enabled, a link state change wakeup causes similar results. For example, setting *PME_Status*, asserting PME#, and setting the *Link Status Changed (LNKC)* bit in the Wakeup Status Register (WUSR) when the link goes up or down.

PME# remains asserted until the OS either writes a 1b to the *PME_Status* bit of the PMCSR register or writes a 0b to the *Pme_En* bit.

After receiving a wakeup packet, the Ethernet controller ignores any subsequent wakeup packets until the driver clears all of the "Received" bits in the Wakeup Status Register (WUS). It also ignores link change events until the driver clears the *Link Status Changed* (LNKC) bit in the Wakeup Status Register (WUSR).

6.4.3 Wakeup Packets

The Ethernet controller supports various wakeup packets using two types of filters:

- Pre-defined Filters
- Flexible Filters

Each of these filters are enabled if the corresponding bit in the Wakeup Filter Control Register (WUFC) is set to 1b.

6.4.3.1 Pre-Defined Filters

The following packets are supported by the Ethernet controller's pre-defined filters:

- Directed Packet (including exact, multicast indexed, and broadcast)
- Magic Packet* (not applicable to the **82541ER**)
- ARP/IPv4 Request Packet (ARP Request Packet for the **82544GC/EI**)
- Directed IPv4 Packet (Directed IP Packet for the **82544GC/EI**)
- Directed IPv6 Packet[1]

Each of these filters are enabled if the corresponding bit in the Wakeup Filter Control Register (WUFC) is set to 1b.

The following explanation of each filter includes a table listing which bytes at which offsets are compared to determine if the packet passes the filter. Note that both VLAN[2] frames and LLC/Snap can increase the given offsets if they are present (see Section 9.3 for details).

For the **82541PI/GI/EI** and **82547GI/EI**, various tables can also include a reference to a possible VLAN Tag and LLC/SNAP Header. These Ethernet controllers detect VLAN and LLC/Snap frames by checking the initial size/type field. They first check for a VLAN header by comparing the size/type field to the value programmed in the VLAN EtherType register. If the field matches, the Ethernet controllers consider the frame a VLAN frame. They then check the VLAN ID against the values programmed in the VLAN Filter Table Array. If the ID matches the packet, processing continues. If the ID doesn't match, or the CTRL.VME bit is 0b, and the VLAN Tag is listed as compare in the table, the packet is not considered a wakeup packet.

After processing a possible VLAN Tag, the **82541PI/GI/EI** and **82547GI/EI** Ethernet controllers check for an LLC/SNAP Header. If the size/type field is less than or equal to 1500 bytes, they check the following 6 bytes for the pattern of AAAA03000000h. If the pattern matches, then the packet processing continues. If the pattern doesn't match, and the LLC/SNAP Header is listed as compare or check in the table, the packet is not considered a wakeup packet.

6.4.3.1.1 Directed Exact Packet

1. Not applicable to the **82544GC/EI**.
2. Not applicable to the **82541ER**.

The Ethernet controller generates a wakeup event after receiving any packet whose destination address matches one of the 16 valid programmed Receive Addresses if the *Directed Exact Wakeup Enable* bit is set in the Wakeup Filter Control Register (WUFC.EX).

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	Match any pre-programmed address

6.4.3.1.2 Directed Multicast Packet

For multicast packets, the upper bits of the incoming packet's destination address indexes a bit vector (*Multicast Table Array*) that indicates whether to accept the packet. If the *Directed Multicast Wakeup Enable* bit set in the Wakeup Filter Control Register (WUFC.MC) and the indexed bit in the vector is one then the Ethernet controller generates a wakeup event. The exact bits used in the comparison are programmed by software in the *Multicast Offset* field of the Receive Control Register (RCTL.MO).

Offset	# of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	See Section 6.4.3.1.2.

6.4.3.1.3 Broadcast

If the *Broadcast Wakeup Enable* bit in the Wakeup Filter Control Register (WUFC.BC) is set, the Ethernet controller generates a wakeup event when it receives a broadcast packet.

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address	FF*6	Compare	

6.4.3.1.4 Magic Packet*[1]

Magic Packet* technology is defined at: http://www.amd.com/products/npd/overview/20212.html.

The Ethernet controller expects the destination address to:

1. Be the broadcast address (FF.FF.FF.FF.FF.FF)
2. Match the value in Receive Address Register 0 (RAH0, RAL0). This is initially loaded from the EEPROM but can be changed by the software driver.
3. Match any other address filtering enabled by the software driver.

The Ethernet controller searches for the contents of Receive Address Register 0 (RAH0, RAL0) as the embedded IEEE address. It considers any non FF byte after a series of at least 6 FFs to be the start of the address for comparison purposes (for example, it catches the case of 7 FFs followed by the address). As soon as one of the first 96 bytes after a string of FFs doesn't match, it continues to search for anther set of at least 6 FFs followed by the 16 copies of the IEEE address later in the packet. Note that this definition precludes the first byte of the destination address from being FF.

1. Not applicable to the **82541ER**.

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header – processed by main address filter
6	6	Source Address		Skip	
12[a]	8	Possible LLC/SNAP Header		Skip	
12[a]	4	Possible VLAN Tag		Skip	
12[a]	4	Type		Skip	
any	6	Synchronizing Stream	FF*6+	Compare	
any+6	96	16 copies of Node Address	A*16	Compare	Compared to Receive Address Register 0 (RAH0, RAL0)

a. Not applicable to the **82541PI/GI/EI** and **82547GI/EI**.

A Magic Packet's destination address must match the address filtering enabled in the configuration registers with the exception that broadcast packets are considered to match even if the *Broadcast Accept* bit of the Receive Control Register (RCTL.BAM) is 0b. If APM Wakeup is enabled in the EEPROM, Ethernet controller starts up with the Receive Address Register 0 (RAH0, RAL0) loaded from the EEPROM. This is enables it to accept packets with the matching IEEE address before the driver comes up.

Note: Accepting broadcast magic packets for wakeup purposes when the *Broadcast Accept* bit of the Receive Control Register (RCTL.BAM) is 0b is a change from the **82544GC/EI** Ethernet controller, which initialized RCTL.BAM to 1b if APM was enabled in the EEPROM, but then required that bit to be 1b to accept broadcast Magic Packets, unless broadcast packets passed another perfect or multicast filter.

6.4.3.1.5 ARP/IPv4 Request Packet[1]

The Ethernet controller supports receiving ARP Request packets for wakeup if the ARP bit is set in the Wakeup Filter Control Register (WUFC). Four IPv4 addresses are supported which are programmed in the IPv4 Address Table (IPv4AT)[2]. A successfully matched packet must contain a broadcast MAC address, a Protocol Type of 0806h, an ARP OPCODE of 01h, and one of the four programmed IPv4 addresses. The Ethernet controller also handles ARP Request packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header – processed by main address filter
6	6	Source Address		Skip	
12 **12 + S[a]**	8 D = (0/8)[a]	Possible LLC/SNAP Header	Type<= 1500 and AAAA_0300_0000h[a]	Skip Check[a]	
12	4 S = (0/4)[a]	Possible VLAN Tag	8100h and check ID[a]	Skip Check[a]	
12 **12 + D + S[a]**	2	Type	0806h	Compare	ARP
14 **14 + D + S[a]**	2	HW Type	0001h	Compare	
16 **16 + D + S[a]**	2	Protocol Type	0800h	Compare	
18 **18 + D + S[a]**	1	Hardware Size	06h	Compare	
19 **19 + D + S[a]**	1	Protocol Address Length	04h	Compare	
20 **20 + D + S[a]**	2	Operation	0001h	Compare	
22 **22 + D + S[a]**	6	Sender HW Address	-	Ignore	
28 **28 + D + S[a]**	4	Sender IP Address	-	Ignore	
32 **32 + D + S[a]**	6	Target HW Address	-	Ignore	
38 **38 + D + S[a]**	4	Target IP Address	IPv4AT[b]	Compare	May match any of 4 values in IPv4AT[b]

a. **82541PI/GI/EI** and **82547GI/EI** only.
b. IPAT for the **82544GC/EI**.

1. ARP Request Packet for the **82544GC/EI**.
2. Four IP addresses are supported which are programmed in the IP Address Table (IPAT) for the **82544GC/EI**.

6.4.3.1.6 Directed IPv4 Packet[1]

The Ethernet controller supports receiving Directed IPv4[2] packets for wakeup if the IPv4 bit is set in the WakeUp Filter Control Register (WUFC). Four IPv4 addresses are supported which are programmed in the IPv4 Address Table (IPv4AT). A successfully matched packet must contain the station's MAC address, a Protocol Type of 0800h, and one of the four programmed IPv4 addresses. The Ethernet controller also handles Directed IPv4 packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header – processed by main address filter
6	6	Source Address		Skip Ignore[a]	
12 12 + S[a]	8 D = (0/8)[a]	Possible LLC/SNAP Header	Type<= 1500 and AAAA_0300_0000h[a]	Skip Check[a]	
12	4 S = (0/4)[a]	Possible VLAN Tag	8100h and check ID[a]	Skip Check[a]	
12 12 + D + S[a]	2	Type	0800h	Compare	IP
14 14 + D + S[a]	1	Version/HDR length	4Xh	Compare	Check IPv4[b] and header length
15 15 + D + S[a]	1	Type of Service	-	Ignore	
16 16 + D + S[a]	2	Packet Length	-	Ignore	
18 18 + D + S[a]	2	Identification	-	Ignore	
20 20 + D + S[a]	2	Fragment Info	-	Ignore	
22 22 + D + S[a]	1	Time to live	-	Ignore	
23 23 + D + S[a]	1	Protocol	-	Ignore	
24 24 + D + S[a]	2	Header Checksum	-	Ignore	
26 26 + D + S[a]	4	Source IP Address	-	Ignore	
30 30 + D + S[a]	4	Destination IP Address	IPv4AT[b]	Compare	May match any of four values in IPv4AT[b]

a. **82541PI/GI/EI** and **82547GI/EI** only.
b. IP instead of IPv4 for the **82544GC/EI**.

1. Directed IP Packet for the **82544GC/EI**.
2. IP instead of IPv4 for the **82544GC/EI**.

6.4.3.2 Directed IPv6 Packet[1]

The Ethernet controller supports receiving Directed IPv6 packets for wakeup if the *IPv6* bit is set in the Wakeup Filter Control Register (WUFC). One IPv6 address is supported and it is programmed in the IPv6 Address Table (IPv6AT). A successfully matched packet must contain the station's MAC address, a Protocol Type of 0800h, and the programmed IPv6 address. The Ethernet controller also handles Directed IPv6 packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

.

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header – processed by main address filter
6	6	Source Address		Skip Ignore[a]	
12 12 + S[a]	8 D = (0/8)[a]	Possible LLC/SNAP Header	Type<= 1500 and AAAA_0300_0000h[a]	Skip Check[a]	
12	4 S = (0/4)[a]	Possible VLAN Tag	8100h and check ID[a]	Skip Check[a]	
12 12 + D + S[a]	2	Type	0800h 86DD[a]	Compare	IP IPv6[a]
14 14 + D + S[a]	1	Version/ Priority Version/Traffic Class[a]	6Xh	Compare	Check IPv6
15 15 + D + S[a]	3	Traffic Class[a]/Flow Label	-	Ignore	
18 18 + D + S[a]	2	Payload Length	-	Ignore	
20 20 + D + S[a]	1	Next Header	- IPv6 Next Header Types[a]	Ignore	
21 21 + D + S[a]	1	Hop Limit	-	Ignore	
22 22 + D + S[a]	16	Source IP Address	-	Ignore	
38 38 + D + S[a]	16	Destination IP Address	IPv6AT	Compare	Match value in IPv6AT

a. **82541PI/GI/EI** and **82547GI/EI** only.

1. Not applicable to the **82544GC/EI**.

6.4.3.3 Flexible Filter

The Ethernet controller supports a total of four flexible filters. Each filter is configured to recognize any arbitrary pattern within the first 128 bytes of the packet. To configure the flexible filter, the software driver must mask values into the Flexible Filter Mask Table (FFMT), the required values into the Flexible Filter Value Table (FFVT), and the minimum packet length into the Flexible Filter Length Table (FFLT). These contain separate values for each filter. The software driver must also enable the filter in the Wakeup Filter Control Register (WUFC) as well as the overall wakeup functionality by setting *PME_En* in the Power Management Control Status Register or the Wakeup Control Register.

Once enabled, the flexible filters scan incoming packets for a match. If the filter encounters any byte in the packet where the mask bit is 1b and the byte doesn't match the byte programmed in the Flexible Filter Value Table (FFVT), then the filter failed that packet. If the filter reaches the required length without failing the packet, it passes the packet and generates a wakeup event. It ignores any mask bits set to 1b beyond the required length. (the wakeup packet is stored, see Section 6.4.3.5).

For the **82541xx** and **82547GI/EI**, the flexible filter does not have any way to automatically skip VLAN or LLC/SNAP headers. If such headers are included, the offsets of the subsequent fields must be adjusted accordingly.

Note: This following flexible packet filters are listed for reference only.

6.4.3.3.1 IPX Diagnostic Responder Request Packet Example[1]

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header – processed by main address filter
6	6	Source Address		Skip	
12	S = (0/4)	Possible VLAN Tag		Compare or Skip	
12 + S	D = (0/8)	Possible LLC/SNAP Header		Compare or Skip	
12 + D + S	2	Type	8137h	Compare	IPX
14 + D + S	16	Some IPX Stuff	-	Ignore	
30 + D + S	2	IPX Diagnostic Socket	0456h	Compare	

1. **82541xx** and **82547GI/EI** only.

6.4.3.3.2 Directed IPX Packet Example

A valid Directed IPX Packet contains the station's MAC address, a Protocol Type of 8137h, and an IPX Node Address that equals to the station's MAC address. It can include *LLC/SNAP Headers* and *VLAN Tags*. Since filtering this packet relies on the flexible filters, which use offsets specified by the OS directly, the OS must account for the extra offset *LLC/SNAP Headers* and *VLAN tags*.

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header – processed by main address filter
6	6	Source Address		Skip	
12 **12 + S**[a]	8 D = (0/8)[a]	Possible LLC/SNAP Header		Skip Compare or Skip[a]	
12	4 S = (0/4)[a]	Possible VLAN Tag		Skip Compare or Skip[a]	
12 **12 + D + S**[a]	2	Type	8137h	Compare	IPX
14 **14 + D + S**[a]	10	Some IPX Stuff	-	Ignore	
24 **24 + D + S**[a]	6	IPX Node Address	Receive Address 0	Compare	Must match Receive Address 0

a. **82541PI/GI/EI** and **82547GI/EI**.

6.4.3.4 IPv6 Neighbor Discovery Filter[1]

In IPv6, a Neighbor Discovery packet is used for address resolution. A flexible filter can be used to check for a "Neighborhood Discovery Packet".

82541xx and 82547GI/EI Only

Offset	# of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header – processed by main address filter, or broadcast
6	6	Source Address		Skip	
12	4	Possible VLAN Tag		Compare or Skip	
12	8	Possible LLC/SNAP Header		Compare or Skip	
12+D+S	2	Type	86DDh	Compare	IP
14+D+S	1	Version/ Traffic Class	6Xh	Compare	Check IPv6
15+D+S	3	Traffic Class/Flow Label	-	Ignore	

1. Not applicable to the **82544GC/EI**.

Offset	# of Bytes	Field	Value	Action	Comment
18+D+S	2	Payload Length	-	Ignore	
20+D+S	1	Next Header	3Ah, 00h, 2Bh, or 3Ch	Check	ICMP, or IPv6 next headers: + routing (2Bh) + dest options (3Ch) + hop-by-hop (00h)
21+D+S	1	Hop Limit	FFh	Check	
22+D+S	16	Source IP Address	-	Ignore	
38+D+S	16	Destination IP Address		Ignore	
54+D+S	N	Possible IPv6 Next Headers	-	Check	Process headers to get next header. Header type must be routing, destination options, or hop-by-hop.
54+D+S+N	1	Type	87h	Check	Neighbor Solicitation
55+D+S+N	1	Code	00h	Check	
56+D+S+N	2	ICMP Header Checksum	-	Ignore	
58+D+S+N	4	Reserved	-	Ignore	
62+D+S+N	16	Target Address	-	Check	Match IPV6AT[0]
78+D+S+N	N	Possible source link-layer address	-	Ignore	
...	any	-	-	Ignore	Packet data
last 4	4	CRC	-	Compare	Validate correct

6.4.3.5 Wakeup Packet Storage

The Ethernet controller saves the first 128 bytes of the wakeup packet in its internal buffer, which can be read through the Wakeup Packet Memory (WUPM) after system wakeup.

Note: This page intentionally left blank.

Ethernet Interface 8

8.1 Introduction

The PCI/PCI-X Family of Gigabit Ethernet Controllers provide a complete CSMA/CD function supporting IEEE 802.3 (10Mb/s), 802.3u (100Mb/s), 802.3z and 802.3ab (1000Mb/s) implementations. They perform all of the functions required for transmission, reception and collision handling called out in the standards.

The internal Gigabit Media Independent Interface/Media Independent Interface (GMII/MII) supports the onboard 10/100/1000 BASE-T transceivers full duplex operation and supports the onboard 10/100 BASE-T transceivers in full or half duplex operation.

Note: The **82541xx**, **82547GI/EI**, and **82540EP/EM** do support SerDes functionality.

8.2 Link Interfaces Overview

82546GB/EB and **82545GM/EM** Ethernet controllers contain an internal 10-bit Fibre Channel Interface (TBI), as specified in IEEE 802.3z, for full-duplex operation with a SerDes transceiver. This configuration is applicable to 1000BASE-SX, -LX, or –CX links. When in internal SerDes mode, they provide the full Physical Coding Sub-layer implementation including Auto-Negotiation as called out in IEEE 802.3z.

82544GC/EI Ethernet controllers support a 10-bit TBI, as specified in IEEE 802.3z, for full-duplex operation with a SerDes transceiver. This configuration is applicable to 1000BASE-SX, -LX, or –CX links. When in TBI mode, they provide the full Physical Coding Sub-layer implementation including Auto-Negotiation as called out in IEEE 802.3z.

Selection between the various configurations is programmable via each MAC's Extended Device Control Register (CTRL_EXT.LINK_MODE bits) and defaulted via EEPROM settings. Note that the external TBI interface is a single resource that can only be associated with a single MAC.

The GMII/MII mode used to communicate between the MAC and the internal PHY supports 10/100/1000 Mbps operation, with both half- and full-duplex operation at 10/100 Mbps, and full-duplex operation at 1000 Mbps.

Note: The Ethernet controllers are optimized for full-duplex operation in 1000 Mbps mode. Half-duplex 1000 Mbps operation is NOT supported and is not recommended.

The internal copper PHY features 10/100/1000-BaseT signalling and is capable of performing intelligent power-management based on both the system power-state and LAN energy-detection (detection of unplugged cables). Power management includes ability to shut-down to extremely low (powered-down) state when not needed, as well as ability to auto-negotiate to lower-speed (and less power-hungry) 10/100 Mbps operation when the system is in low power-states.

8.2.1 Internal SerDes Interface/TBI Mode– 1Gb/s[1]

The **82546GB/EB** and **82545GM/EM** Ethernet controllers contain one or two internal SerDes devices (depending whether or not they support one or two ports). The MAC communicates with the SerDes over a TBI interface. Normally, this interface is not exposed externally.

For the **82554GC/EI**, TBI mode is selectable via an external pin TBI-MODE. Software cannot override this pin. This interface has 125 Mb/s 10-bit data paths for both receive and transmit. The clock at the transmit interface operates at 125 MHz; the receive interface has two clocks running at 62.5 MHz that are 180 degrees out of phase as follows:

- RX_DATA: 10-bit receive data bus
- TX_DATA: 10-bit transmit data bus
- RBC0/I_RBC1: Receive clocks (62.5 MHz; 180 degree phase shift between I_RBC0 and I_RBCI)
- GTX_CLK: Transmit clock (125 MHz)

8.2.1.1 Gigabit Physical Coding Sub-Layer (PCS) for the Internal SerDes[2]

The Ethernet controller integrates the 802.3z PCS function on-chip. The on-chip PCS circuitry is used when the link interface is configured for internal SerDes mode and is bypassed in internal PHY mode.

The packet encapsulation is based on the Fibre Channel physical layer (FC0/FC1) and uses the same coding scheme to maintain transition density and DC balance. The physical layer device is a SerDes and is used for 1000BASE-SX, -LX, or -CX configurations.

8.2.1.2 8B10B Encoding/Decoding

The Gigabit PCS circuitry uses the same transmission coding scheme used in the Fibre Channel physical layer specification. The 8B10B coding scheme was chosen by the IEEE standards committee in order to provide a balanced, continuous stream with sufficient transition density to allow for clock recovery at the receiving station. There is a 25 percent overhead for this transmission code which accounts for the data signaling rate of 1250 Mb/s with 1000 Mb/s of actual data.

1. TBI (10-Bit Interface) - 1GB/s for the **82554GC/EI**.
2. Gigabit Physical Coding Sub-Layer (PCS) for TBI (**82554GC/EI**).

8.2.1.3 Code Groups and Ordered Sets

Code group and ordered set definitions are defined in clause 36 of the IEEE 802.3z standard. These represent special symbols used in the encapsulation of Gigabit Ethernet packets. Table 8-1 lists a brief description of defined ordered sets for informational purposes only.

Table 8-1. Code Group and Ordered Set Usage

Code	Ordered_Set	# of Code Groups	Usage
/C/	Configuration	4	General reference to configuration ordered sets, either /C1/ or /C2/, which is used during Auto Negotiation to advertise & negotiate link operation information between link partners. Last two code groups contain config base and next page registers.
/C1/	Configuration 1	4	See /C/. Differs from /C2/ in second code group for maintaining proper signaling disparity.
/C2/	Configuration 2	4	See /C/. Differs from /C1/ in second code group for maintaining proper signaling disparity.
/I/	IDLE	2	General reference to IDLE ordered sets. IDLE characters are continually transmitted by the end stations and are replaced by encapsulated packet data. The transitions in the IDLE stream allow the SerDes to maintain clock and symbol synchronization between to link partners.
/I1/	IDLE 1	2	See /I/. Differs from /I2/ in second code group for maintaining proper signaling disparity.
/I2/	IDLE 2	2	See /I/. Differs from /I1/ in second code group for maintaining proper signaling disparity.
/R/	Carrier_Extend	1	This ordered set is used to indicate carrier extension to the receiving PCS. It is also used as part of the end_of_packet encapsulation delimiter as well as IPG for packets in a burst of packets.
/S/	Start_of_Packet	1	The SPD (start_of_packet delimiter) ordered set is used to indicate the starting boundary of a packet transmission. This symbol replaces the last byte of the preamble received from the MAC layer.
/T/	End_of_Packet	1	The EPD (end_of_packet delimiter) is comprised of three ordered sets. The /T/ symbol is always the first of these and indicates the ending boundary of a packet.
/V/	Error_Propagation	1	The /V/ ordered set is used by the PCS to indicate error propagation between stations. This is normally intended to be used by repeaters to indicate collisions.

8.2.2 GMII – 1 Gb/s

The internal Gigabit Media Independent Interface (GMII) is similar to the 10/100 Mb/s Media Independent Interface (MII). The GMII uses the MII management interface and registers. These common elements of operation allow the Ethernet controller to determine PHY capabilities for any supported speed of operation and configuration of the hardware based on those capabilities.

Most of the MII and GMII signals use the same names, but the width of the RX and TX data busses and the semantics of the associated control signals differ between MII and GMII operation. The GMII transmit path clocking also differs significantly from MII clocking.

8.2.3 MII – 10/100 Mb/s

The internal MII implementation for the Ethernet controller provides full IEEE 802.3 and IEEE 802.3u compliant operation for 10Mb/s and 100Mb/s operation in conjunction with the onboard MII compliant PHY.

The MII uses a clocked, nibble-wide (4-bit) data path in each direction. The clock rate for Fast Ethernet operation is 25 MHz with data transfer speed of 4 bits x 25 MHz = 100 Mb/s. For 10 Mb/s operation the clock rate is 2.5 MHz and also uses the nibble-wide data path.

8.3 Internal Interface[1]

The Ethernet controller supports the IEEE 802.3 MII Management Interface also known as the Management Data Input/Output (MDI/O) Interface. This interface allows upper-layer devices to monitor and control the state of the PHY.

For the **82546GB/EB**, **82545GM/EM**, **82541xx**, **82540EP/EM**, and **82547GI/EI**, the MDI/O interface consists of an internal connection, a special protocol that runs across the connection, and an internal set of addressable registers. For the **82541xx** and **82547GI/EI**, the physical interface between the MAC and PHY is not available externally.

For the **82544GC/EI**, the MDI/O interface consists of a physical connection, a special protocol that runs across the connection, and an internal set of addressable registers. The physical interface consists of a (B_MDIO) data line and a clock line (O_MDC).

- O_MDC:

  Management Data Clock, used by the PHY as a clock timing reference for information transfer on the B_MDI/O signal. The O_MDC is not a continuous signal and can be frozen by the Ethernet controller when no management data is transferred. The O_MDC signal has a maximum operating frequency of 2.5 MHz.

- B_MDO:

  Management Data I/O, a bidirectional data signal used to transfer control information and status between the Ethernet controller and the PHY (read and write PHY management registers). The B_MDO signal is sampled by the rising edge of the O_MDC signal.

Software can use MDI/O to read and write registers in a internal PHY by accessing the Ethernet controller's MDIC register.

8.4 Duplex Operation

The **82546GB/EB** and **82545GM/EM** supports half-duplex and full-duplex 10/100 Mb/s mode. Half-duplex in 1000 Mb/s mode using either the Internal SerDes or GMII interface is NOT supported.

The **82544GC/EI**, **82540EP/EM**, **82541xx**, and **82547GI/EI**, support half-duplex and full-duplex 10/100 Mb/s mode or 1000 MB/s mode. However, only full-duplex mode is supported when the **82544GC/EI** TBI interface option is used.

1. MDIO/MDC Interface for the **82544GC/EI**, **82540EP/EM**, **82541xx**, and **82547GI/EI**.

Configuration of the duplex operation of the Ethernet controller can be forced or determined via the Auto-Negotiation process. See Section 8.6 for details on link configuration setup and resolution.

8.4.1 Full Duplex

All aspects of the IEEE 802.3, 802.3u, 802.3z, and 802.3ab specifications are supported in full duplex operation. Full duplex operation is enabled by several mechanisms depending on the speed configuration of the Ethernet controller and the specific capabilities of the PHY used in the application. During full duplex operation, the Ethernet controller may transmit and receive packets simultaneously across the link interface.

In Internal Serdes mode for the **82546GB/EB** and **82545GM/EM** (TBI mode for the **82544GC/EI**), the transmission and reception of packets is indicated by symbols embedded in the data stream. These symbols delineate the packet encapsulation and the protocol does not rely on other control signals. See Section 8.2.1.3 for details.

8.4.2 Half Duplex

Note: The Ethernet controller operates in half duplex mode only when configured for internal PHY mode. For the **82546GB/EB** and **82545GM/EM, i**nternal SerDes mode does not support half duplex operation.

In half duplex mode, the Ethernet controller attempts to avoid contention with other traffic on the wire, by monitoring the carrier sense signal provided by the internal PHY, and deferring to passing traffic. When the Internal Carrier Sense signal is deasserted or after sufficient InterPacket Gap (IPG) has elapsed after a transmission, frame transmission can begin.

In the case of a collision, the internal PHY asserts a collision signal. Transmission of the frame stops within four clock times and then the Ethernet controller sends a JAM sequence onto the link. After the end of a collided transmission, the Ethernet controller backs off and attempts to retransmit per the standard CSMA/CD method. Note that the retransmission is done from the data stored internally in the Ethernet controller transmit packet buffer. The Ethernet controller does not access data in host memory again.

In the case of a successful transmission, the Ethernet controller is ready to transmit any other frames queued in its transmit FIFO within the minimum Inter Frame Spacing (IFS) of the link.

The internal carrier sense signal is expected to be asserted before one slot time has elapsed; however, the transmission completes successfully even if internal carrier sense is not asserted. If internal carrier sense is not asserted within the slot time window, the PHY is not behaving properly and can either be configured incorrectly or be in a link down situation. Note that this event is counted in the Transmit Without CRS statistic register (see Section 13.7.12).

Half duplex reception is as indicated for full duplex in Section 8.4.1 except for 1000 Mb/s specific operation, as described in Section 8.4.2.1 and Section 8.4.2.2.

The Ethernet controller does not provide support for half-duplex operation as specified in the IEEE 802.3z specification when operating at 1000 Mb/s in internal PHY mode.

For receives, the Ethernet controller supports carrier extended packets and packets generated during packet bursting operations (see Section 8.4.2.1 and Section 8.4.2.2). The Ethernet controller can be configured to transmit in packet burst mode via the TCTL.PBE bit in the Transmit Control register (see Section 13.4.46).

Carrier extension is only defined in the IEEE 802.3z standard for half-duplex operation for operation frequencies above 100 Mb/s (Gigabit Ethernet).

8.4.2.1 Carrier Extension (1000 Mb/s Only)

One of the objectives of the IEEE 802.3z standard development was to support a maximum collision domain of 200 m and retain the IEEE 802.3 Ethernet frame format. The scaling of the line transfer rate by 10x to 1 Gb/s reduced the bit time by 10 and effectively reduced the theoretical collision domain to an unusable size with the minimum packet size of 64 bytes. To overcome this, the 802.3z specification development added the notion of carrier extension to the standard.

Carrier extension provides a method to increase the duration of the carrier event to a minimum usable duration in order to meet the collision domain objective. Packets that are signaled from the CSMA/CD layer that do not meet the minimum slot time of 512 bytes have extension bytes appended to them in order to meet this minimum slot time requirement. The extension bytes are defined within the context of the frame encapsulation discussion of the 802.3z standard and are recognized by 802.3z compliant devices (see Figure 8-1).

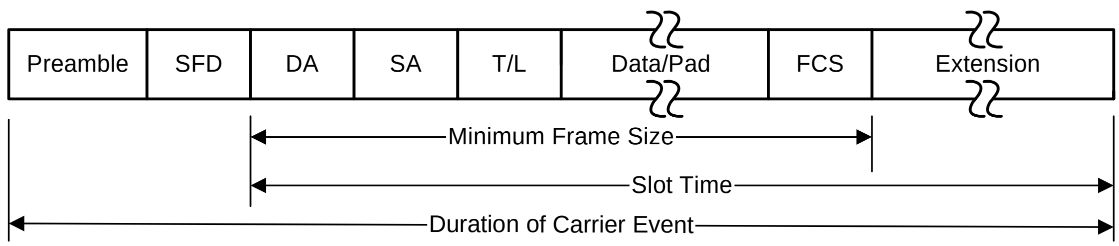


Figure 8-1. Carrier Extended Frame Format

The Ethernet controller supports the reception and transmission of carrier extended packets. Carrier extension is implemented via specifying the collision distance parameter, COLD, in the Transmit Control register (TCTL). Note that this field is evaluated whether in full- or half- duplex operation.

8.4.2.2 Packet Bursting

In an attempt to recover some of the lost overhead encountered with short duration packets using carrier extension, the IEEE 802.3z standard incorporates the implementation of packet bursting. Packet bursting is a mechanism that allows a transmitting device to “own-the-wire” for a longer duration and “pack” extra packets in a burst without relinquishing ownership of the medium. A burst length timer is implemented which allows the Ethernet controller to continue to send packets until the timer expires (if packets are available for transmission).

In the case where a transmitting station has more than one packet to send, it can transmit the first packet (extending to 512 bytes if necessary) and then begin the transmission of subsequent packets. Packet transmission can continue until either there are no more packets ready for transmission, or the burst timer has expired. The burst timer limit is specified as 8 KB.

The normal rules for IPG are followed during packet bursting after the first packet has met the minimum slot time requirements, with the exception that the Inter Frame Content (IFC) is extension symbols rather than IDLEs. Under some circumstances, it might be desirable to extend this IPG time during a burst. This can be done via the *AIFS* field in the AIT register. See Section 13.4.35.

8.5 Auto-Negotiation and Link Setup[1]

Configuration of the link can be accomplished by several methods ranging from software forcing the link settings to Auto-Negotiation by the internal PHY. Section 8.6.1 describes the process of bringing the link up including configuration of the MAC and PHY, as well as the various methods of determining duplex and speed configuration.

The PHY performs Auto-Negotiation per 802.3ab clause 40 and extensions to clause 28. Link resolution is obtained from the PHY after the link has been established via the MDIO interface, by the controller via specific input signals from the PHY or by the controller's specific auto detection functions.

Upon power up, or device reset via the RST# input, the controller initiates Auto-Negotiation based on the default settings in the Device Control and Transmit Configuration Word registers, as well as settings read from the EEPROM. If enabled in the EEPROM, the Ethernet controller will immediately perform Auto-Negotiation.

8.6 Auto-Negotiation and Link Setup[2]

Configuration of the link can be accomplished by several methods ranging from software forcing the link settings to Auto-Negotiation by either the MAC (Internal Serdes mode) or the internal PHY (GMII/MII mode). The following sections describe the process of bringing the link up including configuration of the MAC and PHY, as well as the various methods of determining duplex and speed configuration.

The process of determining link configuration differs slightly depending on the Ethernet controller type and version. In Internal Serdes mode, the MAC performs Auto-Negotiation per clause 37 of the 802.3z standard. The transceiver used in this mode (the SerDes) does not participate in the Auto-Negotiation process as all aspects of Auto-Negotiation are controlled by the MAC.

For internal PHY mode, the PHY performs Auto-Negotiation per 802.3ab clause 40 and extensions to clause 28. Link resolution is obtained from the PHY after the link has been established via the MDI/O interface, by the Ethernet controller via specific input signals from the PHY, or by the Ethernet controller's specific auto detection functions.

The method for configuring the link between two link partners is highly dependent on the mode of operation as well as the functionality provided by the specific physical layer device (PHY or SerDes). For Internal Serdes mode, the Ethernet controller provides the complete 802.3z PCS function on-chip. For GMII/MII mode, the PCS and Auto-Negotiation functions are maintained within the PHY.

1. **82541xx**, **82547GI/EI**, and **82540EP/EM** only.
2. Applicable to the **82546GB/EB**, **82545GM/EM**, and **82544GC/EI** only.

The following section describes the link configuration process in the Internal Serdes for the **82546GB/EB** and **82545GM/EM** (TBI mode for the **82544GC/EI**) and internal PHY modes.

8.6.1 Link Configuration in Internal Serdes/TBI Mode[1]

Internal Serdes for the **82546GB/EB** and **82545GM/EM** (TBI for the **82544GC/EI**) Mode link configuration can be performed via the on-chip PCS function in the Ethernet controller. The hardware supports both hardware and software Auto-Negotiation methods for determining the link configuration, as well as allowing for manual configuration to force the link.

Hardware Auto-Negotiation is the preferred method.

8.6.1.1 Link Speed

Internal Serdes for the **82546GB/EB** and **82545GM/EM** (TBI for the **82544GC/EI**) Mode is only defined for 1000 Mb/s operation. Other link speeds are not supported.

When the **82546GB/EB** and **82545GM/EM** is in internal Serdes mode, the speed determination function is disabled and the Device Status register bits (STATUS.SPEED) bits indicate a value of 10b for 1000 Mb/s.

For the **82544GC/EI**, when the TBI_MODE input is asserted for TBI mode, the speed determination function is disabled and the Device Status register bits (STATUS.SPEED) bits indicate a value of 10b for 1000 Mb/s.

8.6.1.2 Auto-Negotiation

At power up, or Ethernet controller reset via the RST# input, it initiates Auto-Negotiation based on the default settings in the Device Control and Transmit Configuration Word registers, as well as settings read from the EEPROM. If enabled in the EEPROM, the Ethernet controller immediately performs Auto-Negotiation.

TBI Mode Auto-Negotiation, as defined in clause 37 of the IEEE 802.3z standard, provides a protocol for two Ethernet controllers to advertise and negotiate a common operational mode across a Gigabit Ethernet link. The Ethernet controller fully supports the IEEE 802.3z Auto-Negotiation function when using the internal Serdes mode for the **82546GB/EB** and **82545GM/EM** or when using the TBI and on-chip PCS for the **82544GC/EI**.

TBI Mode Auto-Negotiation is used to determine the following information:

- Duplex resolution
- Flow control configuration

Speed for Internal Serdes mode (TBI mode for the **82544GC/EI**) is fixed at 1000 Mb/s, so speed settings in the Device Control register are unaffected by the Auto-Negotiation process.

There are two implementations accessible in the design:

1. A full hardware Auto-Negotiation implementation that does not require software intervention in order to successfully reach a negotiated link configuration.
2. Software driven negotiation.

---

1. TBI mode for the **82544GC/EI**.

A set of registers is provided to facilitate either hardware or software Auto-Negotiation.

The hardware supports both hardware and software Auto-Negotiation methods for determining link configuration as well as allowing for manual configuration to force the link. The IEEE 802.3z specification defines a set of resources that software can use to control a hardware implementation of Auto-Negotiation, but this definition is sub-optimal for Internal Serdes mode (TBI mode for the **82544GC/EI**) and hardware Auto-Negotiation is the preferred method.

In addition, it specifies optional resources that exist only to support the exchange of “Next Pages”, something that is not required for the Ethernet controller. The hardware defined in this specification accepts and exchanges next pages in Internal Serdes mode (TBI mode for the **82544GC/EI**), but does so by dropping all incoming next pages and sending only null next pages. The Ethernet controller can only send null next pages when in hardware Auto-Negotiation. A full next page exchange can take place if software performs Auto-Negotiation.

The Ethernet controller fully complies with IEEE 802.3z with respect to next page exchange in that both link partners must request next page exchange in order to do so.

8.6.1.3 Hardware Auto-Negotiation

Hardware supports negotiation of the link configuration per clause 37 of the 802.3z standard. This is accomplished by the exchange of /C/ ordered sets that contain the txConfigWord register values from TXCW in the third and fourth symbols of the ordered sets.

Bits FD and LU of the Device Status register (STATUS), and ANC of the RXCW register provide status information regarding the negotiated link.

Auto-Negotiation can be initiated by the following:

- LRST transition from 1b to 0b in CTRL register
- ANE transition from 0b to 1b in TXCW register
- Receipt of /C/ ordered set during normal operation
- Receipt of different value of the /C/ ordered set during the negotiation process
- Transition from loss of synchronization to synchronized state (if ANE is enabled)

Resolution of the negotiated link determines device operation with respect to flow control capability and duplex settings. These negotiated capabilities override advertised and S/W controlled device configuration.

Software must configure the *TXCW.txConfigWord* field to the desired advertised base page. The bits in the Device Control register are not mapped to the *txConfigWord* field in hardware until after Auto-Negotiation completes. The Figure 8-2 and Figure 8-3 show txConfigWord and the mapping to the Config_reg Base Page encoding per clause 37 of the standard. Table 8-2 lists the bit contents.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Np	RSV	RSV	RSV	RSV	RSV	RSV	AS	PS	Hd	Fd	RSV	RSV	RSV	RSV	RSV

Figure 8-2. TXCW.txConfigWord

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Np	Ack	Rf2	Rf1	RSV	RSV	RSV	Ps2	Ps1	Hd	Fd	RSV	RSV	RSV	RSV	RSV

Figure 8-3. 802.3z Advertised Base Page Mapping

Table 8-2. Bits Content in TXCW.txConfigWord

Bit	Description
Np	Next Page Indication When set indicates a request for next page exchange
AS	Asymmetric Pause Connection is Desired When set, results in independent enabling/disabling of the flow control receive and transmit. When cleared, results in symmetric enabling/disabling of the flow control receive and transmit
PS	Pause Function When set, indicates that the Ethernet controller is capable and intends to stop upon reception of 802.3x flow control Pause packets. When cleared, indicates that the Ethernet controller is not capable, or does not intend to stop upon reception of flow control Pause packets.
HD	Half-Duplex Ability When set, indicates that the Ethernet controller is capable of working in half-duplex mode of operation
FD	Full Duplex Ability When set, indicates that the Ethernet controller is capable of working in full-duplex mode of operation
RSV	Reserved Should be written as 0b

The reserved bits should be written as zero. The remote fault bits [13:12] can be set by software to indicate remote fault type to the link partner if desired. The AS and PS bits are used for advertisement of PAUSE frame operation. Refer to clause 37 of the 802.3z specification for details.

8.6.1.4 Software Auto-Negotiation

Auto-Negotiation can also be performed by software with TXCW.ANE set to 0b. Data stored in the *txConfigWord* field is transmitted during the configuration process. Software should not (in general) read back the contents of this register.

If hardware loses receive synchronization, the contents of the TXCW register changes and during the time of the change, the value read back can be inconsistent. In the absence of loss of synchronization, the value read back is stable and equal to the last written value.

Software controls the negotiation process by writing the appropriate values to the *txConfigWord* and transmitting /C/ ordered sets by setting txConfig (in TXCW) to 1b. Software must monitor the RXCW register for status of the negotiation process and respond via writes to the TXCW register appropriately.

The software algorithm must follow the state machine implementation of sub-clause 37.3.1.5 of IEEE 802.3z, Figure 37-6. The link timer specification is 10 ms (+10 ms/-0 ms). In some systems, response time for the S/W implementation can make it difficult to meet this requirement if system utilization is high due to latencies on the PCI bus.

For more information, refer to the register definitions for TXCW and RXCW in Sections 13.4.13 and 13.4.14, respectively.

8.6.1.5 Forcing Link

In cases where the Ethernet controller is connected to a non-Auto-Negotiating link partner, the hardware allows for manual configuration of the link via the Device Control register (CTRL). Forcing link can be accomplished by software writing a 1b to CTRL.SLU which forces the TBI PCS logic into a link up state if the LOS input is not asserted. Setting the SLU bit enables the MAC to communicate with the internal SerDes and allows recognition of the LOS signal. If auto-negotiation is enabled (TXCW.ANE = 1b) Set Link Up is ignored. The LINK UP# output, as well as internal status logic, indicates link status.

The TXCW.ANE bit must be set to logic 0b to allow for forcing link. When link is forced via the CTRL.SLU bit, the link cannot come up unless the LOS input is deasserted, implying there is a valid signal being received by the optics or the SerDes.

An interrupt bit, RXCFG, flags software that the hardware is receiving configuration symbols (/C/ codes). Software should mask (enable) this interrupt when forcing link. When the link is forced, the link partner can begin to Auto-Negotiate based due to a reset or enabling of Auto-Negotiation. The reception of /C/ codes causes an interrupt to software and the proper hardware configuration can be set.

8.6.2 Internal GMII/MII Mode

Link configuration in GMII/MII mode is generally determined by the PHY via Auto-Negotiation. The software driver must intervene in cases where a successful link is not negotiated or a programmer desires to manually configure the link. The following sections discuss the methods of link configuration for internal PHY mode.

8.6.2.1 Auto-Negotiation

In GMII/MII mode, the PHY performs the Auto-Negotiation function. The operational details of this function are described in the IEEE P802.3ab draft standard.

Auto-Negotiation provides a method for two link partners to exchange information in a systematic manner in order to establish a link configuration providing the highest common level of functionality supported by both partners. Once configured, the link partners exchange configuration information to resolve link settings such as:

- Speed: 10/100/1000 Mb/s
- Duplex: Full- or Half-
- Flow Control Operation

PHY specific information required for establishing the link is also exchanged, but is not relevant to the operation of the Ethernet controller.

If flow control is enabled in the MAC, the settings for the desired flow control behavior must also be made by software in the PHY and Auto-Negotiation must be restarted. After Auto-Negotiation completes, the software driver must read the MII registers in the PHY to determine the resolved flow control behavior of the link and reflect these parameters in the Ethernet controller register (CTRL.TFCE and CTRL.RFCE).

Once PHY Auto-Negotiation is complete, the PHY asserts the link indication signal. Software MUST set the “set link up” bit in the Device Control Register (CTRL.SLU) before the Ethernet controller recognizes the link. Setting the SLU bit permits the MAC to recognize the LINK signal from the PHY, which indicates the PHY has gotten the link up, and to receive and transmit data.

8.6.2.2 Link Speed

The speed of the link in GMII/MII mode can be determined by several methods with the Ethernet controller. These include:

- Software forced configuration of link speed
- Automatically detecting the Auto-Negotiated speed from the PHY
- Direct indication of speed configuration from the PHY

These methods are discussed in the following sections.

8.6.2.2.1 Forcing Speed

There can be circumstances when the software driver must force the link speed of the Ethernet controller. This can occur when the link is manually configured.

The software driver can force speed in the MAC by setting the CTRL.FRCSPD (force-speed) bit to 1b, and then setting the speed bits in the Device Control register (CTRL.SPEED) to the desired speed setting. See Section 13.4.1 for details.

When forcing the Ethernet controller to a specific speed configuration, the driver must also ensure the PHY is configured to a speed setting consistent with the MAC. This statement implies that software accesses to the PHY either force the speed, or read the MII management status register bits that indicate link speed within the PHY itself.

Forcing the speed setting with CTRL.SPEED also can be accomplished by setting the CTRL_EXT.SPD_BYPS bit. This bit bypasses the internal clock switching logic, and gives complete control to the driver when the speed setting takes place. The CTRL.FRCSPD bit uses the internal clock switching logic, which delays the effect of the speed change.

8.6.2.2.2 Using Auto-Speed Detection (ASD)

The Ethernet controller provides a method in hardware for automatically sensing the speed of the link by observing the receive clock signal generated by the PHY once the link is established. The Auto-Speed Detection (ASD) function is enabled via the ASDE bit in the Device Control register (CTRL.ASDE). ASD provides a method of determining the link speed without the need for software accesses to the MII management registers. ASD is not supported in Internal Serdes mode for the **82546GB/EB** and **82545GM/EM** or TBI mode for the **82544GC/EI**.

In internal PHY mode, the internal receive clock input operates at the byte rate of the link interface. By sensing this clock, the Ethernet controller makes a determination of the link speed and sets the proper configuration in the control registers without software intervention.

The ASD function is initiated upon the assertion of a valid link by the PHY via an internal signal input. After the speed is detected, the Device Control and Device Status register bits are set and reflect the speed of the link. As described earlier, software must set the CTRL.SLU bit to allow the speed selection to take effect.

STATUS.ASDV [9:8], provides the results of speed status indication for diagnostics purposes regardless of whether the Auto-Speed Detection feature is enabled. This function is initiated with a write to the CTRL_EXT.ASDCHK bit. See Section 13.4.6 for details.

8.6.2.2.3 Automatic Detection of Link Speed using SPD-IND

With the CTRL register configure as CTRL.FRCSPD = 0, the speed is reconfigured automatically each time a new linkup event is detected. This configuration is recommended why the PHY is configured for Auto-Negotiation.

8.6.2.3 Duplex

The duplex configuration of the link is also resolved during the Auto-Negotiation process. As previously mentioned, the Ethernet controller supports both full- and half-duplex operation in internal PHY mode. When the PHY asserts its link signal to the MAC, it also communicates the duplex setting.

Software can override the duplex setting via the CTRL.FD bit when the CTRL.FRCDPLX (force duplex) bit is set. If CTRL.FRCDPLX is 0b, the CTRL.FD bit is ignored.

8.6.2.4 MII Management Registers

The software driver is required under some circumstances to read from, or write to, the MII management registers in the PHY. These accesses are performed via the MDIC registers. The MII registers allow the software driver to have direct control over the PHY's operation, which includes:

- Resetting the PHY
- Setting preferred link configuration for advertisement during the Auto-Negotiation process
- Restarting the Auto-Negotiation process
- Reading Auto-Negotiation status from the PHY
- Forcing the PHY to a specific link configuration
- Extended capabilities

The standard set of PHY management registers can be found in the IEEE P802.3ab standard.

8.6.2.5 Comments Regarding Forcing Link

Forcing link in GMII/MII mode requires the software driver to configure both the MAC and the PHY in a consistent manner with respect to each other as well as the link partner. After initialization, the software driver configures the desired modes in the MAC, then accesses the PHY MII registers to set the PHY to the same configuration.

In internal PHY mode, setting the CTRL.SLU bit forces a link up condition in the MAC. The duplex setting at this point should be forced by software on the CTRL.FD bit.

8.6.3 Internal SerDes Mode[1] Control Bit Resolution

Tables 8-3, 8-4, and 8-5[2] list how on-chip Auto-Negotiation affects control bits in the Ethernet controller. Table 8-5 lists the case where software Auto-Negotiation is not performed and link is forced.

Table 8-3. Internal Serdes Mode[1] – Hardware Enabled

TXCW.ANE = 1b

Control Bit	Effect on Control Bits
CTRL.FD	Ignored; duplex is set by priority resolution of TXCW and RXCW
CTRL.SLU	Ignored; it is not possible to force link configuration (ANE takes precedence)
CTRL.RFCE	Set by priority resolution (read only)
CTRL.TFCE	Set by priority resolution (read only)
CTRL.SPEED	No impact; speed always 1000 Mb/s in Internal Serdes[a] mode
STATUS.FD	Set by priority resolution
STATUS.LU	Duplicate of RXCW.ANC (Auto-Negotiation complete)
STATUS.SPEED	Internal SerDes[a] Mode is always 1000 Mb/s; fixed at 10b

a. TBI for the **82544GC/EI**.

Table 8-4. Internal Serdes[1] Mode – Software Enabled

TXCW.ANE = 0b

Control Bit	Effect on Control Bits
CTRL.FD	Duplex is set by software priority resolution
CTRL.SLU	Set by software when Auto-Negotiation is complete.
CTRL.RFCE	Set by software as a result of software priority resolution
CTRL.TFCE	Set by software as a result of software priority resolution
CTRL.SPEED	No impact; speed always 1000 Mb/s in Internal SerDes[a] mode
STATUS.FD	Reflects the value of CTRL.FD
STATUS.LU	Reflects CTRL.SLU and internal link indication
STATUS.SPEED	Internal Serdes[a] Mode is always 1000 Mb/s; fixed at 10b

a. TBI for the **82544GC/EI**.

1. TBI Mode for the **82544GC/EI**.
2. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

Table 8-5. Internal Serdes Mode[1] – Auto-Negotiation Skipped

TXCW.ANE = 0b

Control Bit	Effect on Control Bits
CTRL.FD	Duplex is set by software for the desired mode of operation
CTRL.SLU	Set by software
CTRL.RFCE	Set by software for the desired mode of operation
CTRL.TFCE	Set by software for the desired mode of operation
CTRL.SPEED	No impact; speed always 1000 Mb/s in Internal SerDes[a] mode
STATUS.FD	Reflects the value of CTRL.FD
STATUS.LU	Reflects CTRL.SLU and internal link indication
STATUS.SPEED	Internal SerDes[a] Mode is always 1000 Mb/s; fixed at 10b

a. TBI for the **82544GC/EI**.

8.6.4 Internal PHY Mode Control Bit Resolution

Tables 8-6, 8-7, 8-8, and 8-9 list how Auto-Negotiation affects control bits in the Ethernet Controller.

Refer to IEEE 802.3z, clause 37 for information related duplex and flow control link resolution per the 802.3z Auto-Negotiation method. The Ethernet controller fully complies to the specified resolution functions.

Table 8-6. GMII/MII Mode – PHY Speed Indication

CTRL.FRCSPD = CTRL.ASDE = CTRL.FRCDPLX = 0b

Control Bit	Effect on Control Bits
CTRL.FD	Duplex is set per internal signal after link up assertion by PHY.
CTRL.SLU	Software should set to allow PHY to control.
CTRL.RFCE	Must be set by software after reading flow control resolution from MII registers.
CTRL.TFCE	Must be set by software after reading flow control resolution from MII registers.
CTRL.SPEED	Ignored; no impact on speed.
STATUS.FD	Reflects the value of CTRL.FD as above.
STATUS.LU	Reflects link status and SLU set.
STATUS.SPEED	Speed status bits reflect speed resolved from speed indication inputs from PHY.

1. TBI for the **82544GC/EI**.

Table 8-7. GMII/MII Mode – Auto-Speed Detection

CTRL.FRCSPD = CTRL.FRCDPLX = 0b; CTRL.ASDE = 1b

Control Bit	Effect on Control Bits
CTRL.FD	Duplex is set per internal duplex indication after link up assertion by PHY.
CTRL.SLU	Software should set to allow PHY to control.
CTRL.RFCE	Must be set by software after reading flow control resolution from MII registers.
CTRL.TFCE	Must be set by software after reading flow control resolution from MII registers.
CTRL.SPEED	Ignored; no impact on speed.
STATUS.FD	Reflects the value of CTRL.FD as above.
STATUS.LU	Reflects internal link status
STATUS.SPEED	Speed status bits reflect speed resolved from ASD function.

Table 8-8. GMII/MII Mode – Force Speed

CTRL.FRCSPD = 1b; CTRL.FRCDPLX = 0b; CTRL.ASDE = X

Control Bit	Effect on Control Bits
CTRL.FD	Duplex is set per internal duplex indicates input after link up assertion by PHY.
CTRL.SLU	Software should set to allow PHY to control.
CTRL.RFCE	Must be set by software after reading flow control resolution from MII registers.
CTRL.TFCE	Must be set by software after reading flow control resolution from MII registers.
CTRL.SPEED	Set by software to set speed of the MAC; must match PHY speed settings.
STATUS.FD	Reflects the value of CTRL.FD.
STATUS.LU	Reflects internal link status.
STATUS.SPEED	Speed status bits reflect speed forced by CTRL.SPEED.

Table 8-9. GMII/MII Mode – Force Link

CTRL.FRCSPD = CTRL.FRCDPLX = CTRL.SLU = 1b

Control Bit	Effect on Control Bits
CTRL.FD	Set by software.
CTRL.SLU	Set by software, and assumed PHY is also forced to assert link.
CTRL.RFCE	Set by software for the desired mode of operation.
CTRL.TFCE	Set by software for the desired mode of operation.
CTRL.SPEED	Set by software.
STATUS.FD	Reflects the value of CTRL.FD.
STATUS.LU	Reflects CTRL.SLU set and internal link status.
STATUS.SPEED	Reflects CTRL.SPEED.

8.6.5 Loss of Signal/Link Status Indication

For the **82546GB/EB** and **82545GM/EM**, the internal LOS signal allows for indication of physical link status to the Ethernet controller's MAC.

For the **82544GC/EI**, the LOS input is provided to allow for indication of physical link status to the Ethernet controller. When the **82544GC/EI** is configured in TBI mode, the input is typically connected to the loss-of-signal connection from the optics while in internal PHY mode.

If the LSC (Link Status Change) interrupt is enabled, the hardware posts an interrupt to be serviced by the software driver when the link goes up or down. See Section 3.4.3 for more details.

8.6.5.1 Internal Serdes Mode [1]

When asserted, LOS indicates there is no activity on the fiber due to either an unplugged cable or a defective optical device. An assertion on LOS implies the link is not available and the hardware is disabled. This is true whether the link is forced by the Set Link Up bit (CTRL.SLU) or if the Ethernet controller is configured to perform Auto-Negotiation. When Auto-Negotiation is enabled, the Ethernet controller is forced to restart Auto-Negotiation but does not complete the negotiation process until the LOS is deasserted.

8.6.5.2 Internal PHY Mode

While in internal PHY mode, an internal signal provides status of the physical link as indicated by the PHY. Indication that the link is not up disables MAC operation. Upon determination of a valid link, the assertion of the internal link signal asserts the LSC interrupt (if enabled) to indicate to the software driver to check the link status.

1. TBI mode for the **82544GC/EI**.

8.7 10/100 Mb/s Specific Performance Enhancements

8.7.1 Adaptive IFS[1]

The Ethernet controller supports back-to-back transmit Inter-Frame-Spacing (IFS) of 960 ns in 100 Mb/s operation and 9.6 μs in 10 Mb/s operation. Although back-to-back transmission is normally desirable, sometimes it can actually hurt performance in half-duplex environments due to excessive collisions. Excessive collisions are likely to occur in environments where one station is attempting to send large frames back-to-back, while another station is attempting to send acknowledge (ACK) packets.

The Ethernet controller contains an Adaptive IFS Throttle - AIT register (see Section 13.4.35) that enables the implementation of a driver-based adaptive IFS algorithm for collision reduction. Adaptive IFS throttles back-to-back transmissions in the transmit MAC and delays their transfer to the CSMA/CD transmit function. Normally, this register should be set to zero. However, if additional delay is desired between back-to-back transmits, then this register can be set with a value greater than zero. By setting this register with a higher value, collisions can be reduced in certain half-duplex environments, because the adapter is less aggressive in acquiring the wire, and therefore less likely to collide with another adapter that attempts to transmit after minimum IFS.

Note: IFS and IPG (inter-packet gap) are equivalent terms and may be used interchangeably in this manual.

The AIFS field provides a similar function to the IGPT field in the TIPG register (see Section 13.4.34). However this Adaptive IFS throttle register counts in units of transmit clocks (which are 8 ns, 80 ns, 800 ns for 10, 100, 1000 Mb/s mode respectively), and is 16 bits wide, thus providing a greater maximum delay value.

Using values lower than a certain minimum (determined by the ratio of transmit clock to link speed), has no effect on back-to-back transmission. This is because the Ethernet controller does not start transmission until the minimum IEEE IFS (9.6 us at 10 Mb, 960 ns at 100 Mb, and 96 ns at 1 Gb) has been met regardless of the value of Adaptive IFS. For example, if the Ethernet controller is configured for 100 Mb/s operation, the minimum IEEE IFS at 100 Mb/s is 960 ns. Setting AIFS to a value of 10 (decimal) would not affect back-to-back transmission time on the wire, because the 800 ns delay introduced (10 * 80n s = 800 ns) is less than the minimum IEEE IFS delay of 960 ns. However, setting this register with a value of 20 (decimal), which corresponds to 1600 ns for the above example, would delay back-to-back transmits because the ensuing 1600 ns delay is greater than the minimum IFS time of 960 ns.

It is important to note that this register has no effect on transmissions that occur immediately after receives or on transmissions that are not back-to-back (unlike the IPGR1 and IPGR2 values in the TIPG register described in Section 13.4.34). In addition, Adaptive IFS also has no effect on re-transmission timing (re-transmissions occur after collisions). Therefore, AIFS is only enabled in back-to-back transmission. The AIFS value is NOT additive to the TIPG.IPGT value; instead, the actual IPG equals the larger of AIFS and TIPG.IPGT.

1. Not applicable to the **82541xx** or **82547GI/EI**.

8.7.2 Flow Control

Flow control as defined in IEEE specification 802.3x, as well as the specific operation of asymmetrical flow control defined by 802.3z, are supported. The following registers are defined for the implementation of flow control:

Table 8-10. Flow Control Registers

Register Name	Description
Flow Control Address Low, High (FCAL/H)	6-byte flow control multicast address
Flow Control Receive Thresh Hi (FCRTH)	13-bit high water mark indicating receive buffer fullness
Flow Control Transmit Timer Value (FCTTV)	16 bit timer value to include in transmitted PAUSE frame
Flow Control Type (FCT)	16-bit field to indicate flow control type
Flow Control Receive Thresh Lo (FCRTL)	13-bit low water mark indicating receive buffer emptiness

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

The implementation of asymmetric flow control allows for one link partner to send flow control packets while being allowed to ignore their reception. For example, not required to respond to PAUSE frames.

For the **82541xx** and **82547GI/EI**, there are two forms of flow control that can be established via auto-negotiation: symmetric and asymmetric. Symmetric flow control is for point-to-point links; asymmetric for hub-to-end-node connections. Symmetric flow control allows either node to flow-control the other. Asymmetric flow control allows a repeater or switch to flow-control a DTE, but not vice versa

8.7.3 MAC Control Frames & Reception of Flow Control Packets

Three comparisons are used to determine the validity of a flow control frame:

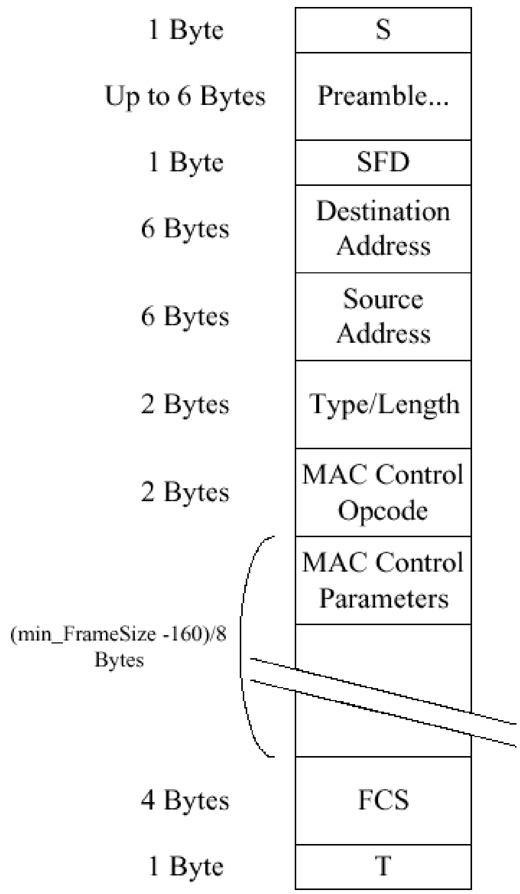
1. A match on the 6-byte multicast address for MAC Control Frames or to the station address of the device (Receive Address Register 0).
2. A match on the type field.
3. A comparison of the MAC Control Opcode field.

Standard 802.3x defines the MAC Control Frame multicast address as 01_80_C2_00_00_01h. This address must be loaded into the Flow Control Address Low/High registers (FCAL/H).

The Flow Control Type register (FCT) contains a 16-bit field that is compared against the flow control packet's type field to determine if it is a valid flow control packet: XON or XOFF. 802.3x reserves this value as 8808h. This number must be loaded into the Flow Control Type (FCT) register.

The final check for a valid PAUSE frame is the MAC Control Opcode. At this time only the PAUSE control frame opcode is defined. It has a value of 0001h.

Frame based flow control differentiates XOFF from XON based on the value of the PAUSE timer field. Non-zero values constitute XOFF frames while a value of zero constitutes an XON frame. Values in the timer field are in units of slot time. A “slot time” is hard wired to 64 byte times, or 512 ns.



Note: “S” is the Start-of-Packet delimiter and “T” is the first part of the End-of-Packet delimiters for 802.3z encapsulation.

Figure 8-4. 802.3x MAC Control Frame Format

The receiver is enabled to receive flow control frames if flow control is enabled through the RFCE bit in the Device Control register (CTRL). Software sets this bit consistently with the advertised capability in the Transmit Configuration Word Register (TXCW).

Flow control capability must be negotiated between link partners via the Auto-Negotiation process. The Auto-Negotiation process can modify the value of these bits based on the resolved capability between the local device and the link partner.

Once the receiver has validated the reception of an XOFF, or PAUSE frame, the Ethernet controller performs the following:

- Increment the appropriate statistics register(s)
- Set the TXOFF bit in the Device Status Register (STATUS)
- Initialize the pause timer based on the packet's PAUSE timer field
- Disable packet transmission or schedule the disabling of transmission after the current packet completes.

Resumption of transmission can occur under the following conditions:

- Expiration of the PAUSE timer
- Reception of on XON frame (a frame with its PAUSE timer set to 0b)

Either condition clears the STATUS.TXOFF bit and transmission can resume. Hardware records the number of received XON frames in the XONRXC counter.

8.7.4 Discard PAUSE Frames and Pass MAC Control Frames

Note: When receive flow control is enabled (CTRL.RFCE) is 1b, the following special filtering is performed on PAUSE and MAC Control frames. When receive flow control is disabled, these frames are filtered like any other frames and the rest of this section can be ignored.

Two bits in the Receive Control register (RCTL) are implemented specifically for control over receipt of PAUSE and MAC control frames. These bits are Discard PAUSE Frames (DPF) and Pass MAC Control Frames (PMCF). See Section 13.4.22 for DPF and PMCF bit definitions.

The DPF bit forces the discarding of any valid PAUSE frame addressed to the Ethernet controller's station address. If the packet is a valid PAUSE frame and is addressed to the station address (receive address [0]), the Ethernet controller does not pass the packet to host memory if the DPF bit is set to logic high. The DPF bit does not affect pause frames that are addressed to the MAC control frame multicast address (01-80-C2-00-00-01). These frames are DMA'ed if they pass standard address filtering, including receive address 1 to 15, multicast hash filtering, or the *Multicast Promiscuous* bit is enabled. TheDPF has no affect on PAUSE operation, only the DMA function.

The PMCF bit allows for the passing of any valid MAC control frames to the system which do not have a valid PAUSE opcode. In other words, the frame can have the correct MAC control frame multicast address (or the MAC station address) as well as the correct type field match with the FCT register, but does not have the defined PAUSE opcode of 0001h. Frames of this type are transferred to host memory when PMCF is logic high. The results of this filter are logically ORed into the standard filters, so even if PMCF is 0b, any MAC control frame that isn't a PAUSE frame that passes standard address filtering is DMA'ed

8.7.5 Transmission of PAUSE Frames

Transmitting PAUSE frames is enabled by software writing a 1b to the CTRL.TFCE bit. This bit is mapped to bit 8 of the TXCW txConfigWord field. (ASM_DIR bit).

Similar to the reception flow control packets described earlier, XOFF packets can be transmitted only if this configuration has been negotiated between the link partners via the Auto-Negotiation process. In other words, the setting of this bit indicates the desired configuration. The resolution of the Auto Negotiation process is discussed in Sections 8.6.3 and 8.6.4.

The contents of the Flow Control Receive Threshold High register (FCRTH) determine at what point hardware transmits a PAUSE frame. Hardware monitors the fullness of the receive FIFO and compares it with the contents of FCRTH. When the threshold is reached, hardware sends a PAUSE frame with its pause time field equal to FCTTV. Once the receive buffer fullness reaches the low water mark, hardware sends an XON message (a PAUSE frame with a timer value of 0b). Software enables this capability with the XONE field of the FCRTL.

Hardware sends one more PAUSE frames if it has previously sent one and the FIFO overflows (so the threshold must not be set greater than the FIFO size). This function is intended to minimize the number of packets dropped if the first PAUSE frame does not reach its target.

Transmitting Flow Control frames should only be enabled in full duplex mode per the IEEE 802.3 standard. Software should ensure that the transmission of flow control packets is disabled when the Ethernet controller is operating in half-duplex mode.

8.7.6 Software Initiated PAUSE Frame Transmission

The Ethernet controller has the added capability to transmit an XOFF frame through software. This function is accomplished by software writing a 1b to the SWXOFF bit of the Transmit Control register (TCTL). Once this bit is set, hardware initiates the transmission of a PAUSE frame in a manner similar to that automatically generated by hardware.

The SWXOFF bit is self clearing after the PAUSE frame has been transmitted.

The state of the CTRL.TFCE bit or the negotiated flow control configuration does not affect software generated PAUSE frame transmission.

Software sends an XON frame by programming a zero in the PAUSE timer field of the FCTTV register.

Caution: Use of SWXOFF is not recommended due to security concerns.

8.7.7 External Control of Flow Control Operation[1]

Transmitting XOFF and XON frames can be triggered by external pins. When enabled through FCRTH.XFCE, the XOFF and XON inputs can be used to provide external effective threshold information that initiate XOFF and XON transmission, respectively.

1. **82544GC/EI** only.

When the XOFF signal is asserted high, the device transmits a single XOFF frame. The assertion of I_XON (after deassertion of XOFF) initiates an XON frame transmission if enabled by FCRTL.XONE. The assertion/deassertion of XON is required between assertions of XOFF in order to send another XOFF frame, providing a built-in hysteresis mechanism.

Output signals are also provided from the **82544GC/EI** to indicate the device is either above the programmed flow control high threshold or below the flow control low threshold (ABV_HI and BLW_LOW respectively).

Flow control transmission must also be enabled through CTRL.TFCE.

Note: This page intentionally left blank.

802.1q VLAN Support 9

The PCI/PCI-X Family of Gigabit Ethernet Controllers provide several specific mechanisms to support 802.1q VLANs:

- Optional adding (for transmits) and stripping (for receives) of IEEE 802.1q VLAN tags
- Optional ability to filter packets belonging to certain 802.1q VLANs

Note: The **82541ER** Ethernet controller does not support VLAN tags.

9.1 802.1q VLAN Packet Format

Table 9-1 compares the format of an untagged 802.3 Ethernet packet with an 802.1q VLAN tagged packet. The CRC for the 802.1q tagged frame is re-computed, so that it covers the entire tagged frame including the 802.1q tag header.

Table 9-1. VLAN Packet Format Comparison

802.3 Packet	#Octets
DA	6
SA	6
Type/Length	2
Data	46-1500
CRC	4

802.1q VLAN Packet	#Octets
DA	6
SA	6
8021.q Tag	4
Type/Length	2
Data	46-1500
CRC*	4

Maximum frame size for a standard 803.3ac (802.1q VLAN and/or 802.1p priority) packet is 1522 octets as opposed to 1518 octets for a normal 802.3 Ethernet packet. If jumbo frames are used, enabling 802.3ac adds 4 bytes to the packet to accommodate the q-tag. If multiple descriptors are required for a transmit, the q-tag information for the packet is extracted from only the last descriptor of the packet. VLAN tagging is supported independently of packet size.

9.1.1 802.1q Tagged Frames

For 802.1q, the Tag Header field consists of four octets containing the Tag Protocol Identifier (TPID) and Tag Control Information (TCI), each utilizing 2 octets. The first 16 bits of the tag header make up the TPID. It contains the “protocol type” which identifies the packet as a valid 802.1q tagged packet.

The two octets making up the TCI contain three fields (see Table 9-2 for details):

- User Priority (UP)
- Canonical Form Indicator (CFI). The CFI should be 0b for transmits. For receives, the Ethernet controller has the capability to filter out packets that have this bit set. See the CFIEN and CFI bits in the RCTL as described in Section 13.4.22.
- VLAN Identifier (VID)

Table 9-2. 802.1q Tagged Frames

Octet 1								Octet 2							
UP			CFI	VID											

9.2 Transmitting and Receiving 802.1q Packets

Since the 802.1q tag is only four bytes, adding and stripping of tags can done completely in software. (For transmits, software inserts the tag into packet data before it builds the transmit descriptor list, and for receives, software strips the four byte tag from the packet data before delivering the packet to upper layer software.)

However, because adding and stripping of tags in software results in more overhead for the host, the Ethernet controller has additional capabilities to add and strip tags in hardware, as discussed in the following two sections.

9.2.1 Adding 802.1q Tags on Transmits

Software can command the Ethernet controller to insert an 802.1q VLAN tag on a per packet basis. If CTRL.VME is set to 1b, and the VLE bit in the transmit descriptor is set to 1b, then the Ethernet controller inserts a VLAN tag into the packet that it transmits over the wire. The Tag Protocol Identifier (TPID) field of the 802.1q tag comes from the VLAN Ether Type (VET) register, and the Tag Control Information (TCI) of the 802.1q tag comes from the special field of the transmit descriptor (TDESC.SPECIAL).

9.2.2 Stripping 802.1q Tags on Receives

Software can instruct the Ethernet controller to strip 802.1q VLAN tags from received packets. If the CTRL.VME bit is set to 1b, and the incoming packet is an 802.1q VLAN packet (its Ethernet Type field matched the VET register), then the Ethernet controller strips the 4-byte VLAN tag from the packet, and stores the TCI in the Special field of the receive descriptor.

The Ethernet controller also sets the VP bit in the receive descriptor to indicate that the packet had a VLAN tag that was stripped. If the CTRL.VME bit is not set, the 802.1q packets can still be received if they pass the receive filter. In this case, the VLAN tag is not stripped and the VP bit is not set. Refer to Table 9-3 for more information regarding receive packet filtering.

9.3 802.1q VLAN Packet Filtering

VLAN filtering is enabled by setting the RCTL.VFE bit to 1b. If enabled, hardware compares the type field of the incoming packet to a 16-bit field in the VLAN EtherType (VET) register. If the VLAN type field in the incoming packet matches the VET register, the 802.1q VLAN packet is then compared against the VLAN Filter Table Array (VFTA) for acceptance.

The Virtual LAN ID field indexes a 4096 bit vector. If the indexed bit in the vector is 1b, there is a Virtual LAN match. Software can set the entire bit vector to 1b's if the node does not implement 802.1q filtering.

In summary, the 4096 bit vector is composed of 128 32-bit registers. Matching to this bit vector follows the same algorithm as indicated in Section 13.5.1 for Multicast Address filtering. The VLAN Identifier (VID) field consists of 12 bits. The upper 7 bits of this field are decoded to determine the 32-bit register in the VLAN Filter Table Array to address and the lower 5 bits determine which of the 32 bits in the register to evaluate for matching.

Two other bits in the Receive Control register (see Section 13.4.22), CFIEN and CFI, are also used in conjunction with 802.1q VLAN filtering operations. CFIEN enables the comparison of the value of the CFI bit in the 802.1q packet to the Receive Control register CFI bit as an acceptance criteria for the packet.

Note: The VFE bit does not affect whether the VLAN tag is stripped. It only affects whether the VLAN packet passes the receive filter.

Table 9-3 lists reception actions according to control bit settings.

Table 9-3. Packet Reception Decision Table

Is packet 802.1q?	CTRL. VME	RCTL. VFE	ACTION
No	X	X	Normal packet reception.
Yes	0	0	Receive a VLAN packet if it passes the standard filters (only). Leave the packet as received in the data buffer. Clear the VP bit in the receive descriptor.
Yes	0	1	Receive a VLAN packet if it passes the standard filters and the VLAN filter table. Leave the packet as received in the data buffer (the VLAN tag is not stripped). Clear the VP bit in the receive descriptor.
Yes	1	0	Receive a VLAN packet if it passes the standard filters (only). Strip off the VLAN information (four bytes) from the incoming packet and store in the descriptor. Set the VP bit in the receive descriptor.
Yes	1	1	Receive a VLAN packet if it passes the standard filters and the VLAN filter table. Strip off the VLAN information (four bytes) from the incoming packet and store in the descriptor. Set the VP bit in the receive descriptor.

Note: This page intentionally left blank.

Configurable LED Outputs 10

10.1 Configurable LED Outputs[1]

The PCI/PCI-X Family of Gigabit Ethernet Controller's MAC implements four output drivers intended for driving external LED circuits. Each MAC's four LED outputs can be individually configured to select the particular event, state, or activity that is indicated on that output. In addition, each LED can be individually configured for output polarity as well as for blinking vs. non-blinking (steady-state) indication.

The configuration for LED outputs is specified via the LEDCTL register. In addition, the hardware-default configuration for two of the LED outputs, LED0/LINK_UP# and LED2/LINK100# can be specified via EEPROM fields, thereby supporting LED displays configurable to a particular OEM preference.

10.1.1 Selecting an LED Output Source

Each of the four LED indications can be independently configured. The LEDCTL register MODE field corresponding to each LED selects the expression generating the LED output. The LED outputs are, by default, active low; it is assumed they are connected to the negative side (cathode) of an external LED. They will, by default, output a low value upon the assertion of the event (such as COLLISION) or state (such as LINK1000#) selected. Note that the active sense of the LED outputs can be inverted). See Section 10.1.2 for details.

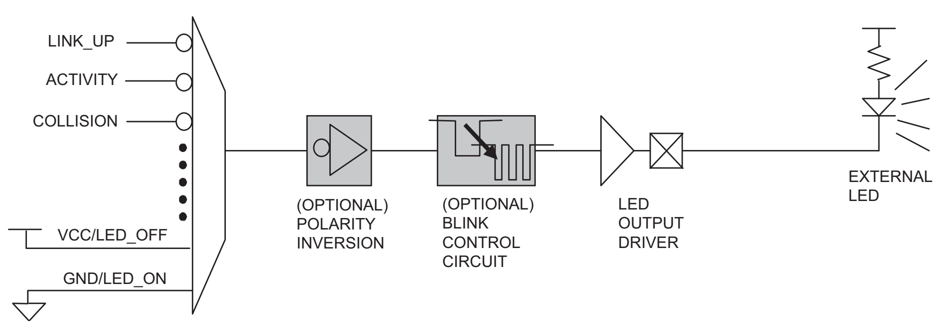


Figure 10-1. Selecting an LED Output Source

1. Section 10 does not apply to the **82544GC/EI**.

LED outputs can be based on the following expressions:

- LINK_UP is asserted while link of any speed is maintained
- LINK_10 indicates link at 10 Mbps
- LINK_100 indicates link at 100 Mbps
- LINK_1000 indicates link at 1000 Mbps
- LINK_100/1000 indicates link at either 100 or 1000 Mbps
- LINK_10/1000 indicates link at either 10 or 1000 Mbps
- ACTIVITY is asserted when link is established and packets are being transmitted or received
- LINK/ACTIVITY is asserted when link is established but there is NO transmit or receive activity
- COLLISION is asserted each time a collision is observed
- PAUSED is asserted while the Ethernet controller’s transmitter is paused due to flow control
- PCIX_MODE is asserted when the Ethernet controller is in PCI-X mode (versus PCI mode)
- FULL_DUPLEX is asserted when the link is configured for full duplex operation
- BUS_SPEED is asserted in PCI 66 MHz or PCI-X 133 MHz configurations (high-speed operation)
- BUS_SIZE is asserted in 64-bit PCI or PCI-X configurations
- LED_ON is always asserted (low); LED_OFF is always deasserted (high)

10.1.2 Polarity Inversion

The LEDCTL.IVRT field enables the selected LED source to be optionally inverted. This can be used to drive external circuitry where an active high indication of one of the selectable states/ events is required (such as multi-color LED circuits).

Note: Polarity inversion (LEDCTL.IVRT = 1b) and blinking (LEDCTL.BLINK = 1b) at the same time for a given LED is not recommended. Introducing additional polarity inversion on a selected state/ event while blink-control is also enabled can produce nonsensical LED behavior (such as blinking LED’s during periods of NO activity or when link is down).

10.1.3 Blink Control

Each LED’s output circuitry also includes a blink-control circuit that can additionally be enabled. The blink control circuitry turns its output sequentially on (low) for 200 ms, then off for another 200 ms, each time its input is active/asserted. The LEDCTL.BLINK field controls whether a blink circuit is enabled for an LED output.

The blink control is especially useful for ensuring that certain brief events, such as momentary ACTIVITY or COLLISION events, cause LED transitions which are sufficiently visible to a human eye. The circuit re-evaluates after each on/off blink cycle, ensuring a continuous blink pattern throughout periods of continuous event/state assertion (such as heavy ACTIVITY periods or long PAUSED times).

Note: It is especially important to note with respect to the blink-control circuit that:

- the blink circuit, when enabled, exists as the LAST stage of the LED circuitry, after any (optional) signal inversion
- the blink sequence occurs when the circuit input is asserted low

As a result, it is possible to select combinations of IVRT and BLINK which do not make sense or produce unexpected results, such as examples previously noted. It is recommended that BLINK only be selected for indicating ACTIVITY, COLLISION, PAUSED, or the combination LINK/ ACTIVITY signal. events/states, and that IVRT = 0b when blink is selected.

Note: Selecting the LEDCTL.MODE = LINK/ACTIVITY with BLINK = 1b selects a unique LED output expression (this configuration is meaningful ONLY when IVRT inversion is disabled). In this configuration, the LED is off (output high) if there is no LINK, on if there is LINK but no ACTIVITY, and blinking if there is LINK with ACTIVITY.

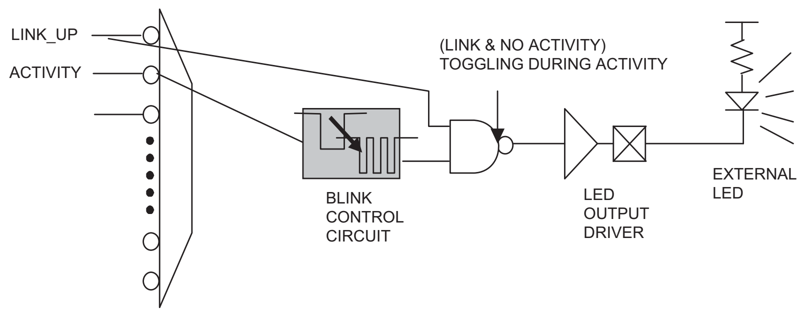


Figure 10-2. Blink Control

Note: This page is intentionally left blank.

PHY Functionality and Features 11

11.1 Auto-Negotiation

Auto-Negotiation between the PCI/PCI-X Family of Gigabit Ethernet Controllers and its link partner is performed by the PHY. Under normal, expected operating conditions, the MAC automatically establishes common speed and duplex settings via the PHY. This section details PHY configuration features involved in the auto-negotiation process.

11.1.1 Overview

Auto-Negotiation by the PHY is initiated upon any of the following conditions:

- Power-up reset (copper and fiber)
- PHY detects loss of link (copper and fiber)
- PHY detects re-appearance of energy on the link (copper and fiber)
- MAC control of PHY power-management is enabled (CTRL.EN_PHY_PWR_MGMT = 1b and MAC transitions to low power state (D3) where continued PHY operation required for wakeup/manageability (copper and fiber)
- PHY hardware reset asserted using the MAC CTRL.PHY_RST bit (copper only)
- PHY soft-reset initiated via the PHY Control Register (bit 15, copper only)
- Explicit Auto-Negotiation Re-Start initiated via the PHY Control Register (bit 9, copper only)
- Explicit transition of PHY from internal IEEE power-down to normal mode via the PHY Control Register (bit 11, copper only)
- Explicit transition of PHY from internal IEEE power-down to normal mode via the PHY Control Register by setting CTRL.LRST = 1 and TXCW.ANE = 1 (fiber only)

Hardware defaults for the PHY configurations enable the PHY to advertise its full 1000BASE-T and 1000BASE-X capability, and to auto-negotiate to the best possible operation[1] without any software intervention required. If the remote device does not have Auto-Negotiation capability, the Ethernet controller PHY uses the parallel detect function to determine the speed of the remote device for 100BASE-TX and 10BASE-T modes. Under certain circumstances, it might be desirable to configure auto-negotiation options to restrict certain behavior. For example, operate in half-duplex mode only.[2]

Note: Any PHY auto-negotiation options configured by software are only persistent while the LAN power (indicated by LAN_PWR_GOOD) remains available. Following a complete loss of power, the PHY reverts to auto-negotiation using its hardware-defaults.

1. 1000 half-duplex not supported.
2. TXCW and RXCW registers are used for fiber auto-negotiation advertising. For fiber, the MAC can be forced to 1000 full-duplex when connected to a non-auto-negotiating fiber switch.

11.1.2 Next Page Exchanges

If 1000BASE-T mode is advertised, then the Ethernet controller PHY automatically sends the appropriate next pages to advertise the capability and negotiate master/slave mode of operation. If a developer does not want to transmit additional next pages, the next page bit (PCI-Config Register bit 15) can be set to 0b and the software need take no further action.

If next pages in addition to the ones required for 1000BASE-T are needed, then the software can set Auto-Negotiation Expansion Register bit 15 to 1b, and send and receive additional next pages via the Next Page Transmit Register (NPT) and Link Partner Next Page Register (LPN), respectively. The PHY stores the previous results from the Link Partner Next Page Register (LPN) in internal registers so that new next pages can overwrite the Link Partner Next Page Register (LPN).

Note: 1000BASE-T next page exchanges are automatically handled without any software intervention, regardless of whether or not additional next pages are sent.

11.1.3 Register Update

Changes to PHY Control Register bits 6, 8, 12, and 13, and PHY Specific Control Register bits 3, 4, 6:5, 9:8 and 11, do not take effect unless one of the following takes place (copper only):

- PHY soft reset (PHY Control Register bit 15)
- Restart Auto-Negotiation (PHY Control Register bit 9)
- Transition of PHY from IEEE power-down to normal operation (PHY Control Register bit 11)
- The link goes down

To enable or disable Auto-Negotiation, PHY Control Register bit 12 should be changed simultaneously with either PHY Control Register bits 15 or 9. For example, to disable Auto-Negotiation and force 10BASE-T half-duplex mode, the PHY Control Register should be written with 8000h.

To disable Auto-Negotiation (fiber only), set TXCW.ANE = 0.

The Auto-Negotiation Expansion Register and the 1000BASE-T Control Register are internally latched once every time the Auto-Negotiation enters the *Ability Detect* state in the arbitration state machine. As a result, a write to the Auto-Negotiation Expansion Register or the 1000BASE-T Control Register has no effect once the PHY begins to transmit Fast Link Pulses (FLPs). This guarantees that sequences of FLPs transmitted are consistent with one another.

The Next Page Transmit Register is treated similarly to the Auto-Negotiation Expansion Register and the 1000BASE-T Control Register during additional next page exchanges.

11.1.4 Status

Once the PHY completes auto-negotiation, it updates the various statuses in the PHY Status Register, Link Partner Ability Register (Base Page), Auto-Negotiation Expansion Register, and 1000BASE-T Status Register. For 1000BASE-T operation, the Auto-Negotiation Expansion Register and the Link Partner Ability Register (Base Page) are updated. Speed, duplex, page received, and Auto-Negotiation completion statuses are also available in the PHY Specific Status Register (PSTATUS) and the PHY Interrupt Status Register (PINTS).

For fiber, the CTRL.STATUS register will reflect link status.

Assuming normal MAC configuration, the MAC status register STATUS reports bits SPEED, FD (duplex/half indication), and LU (link up status) shortly after the PHY (or MAC, for fiber) completes auto-negotiation.

11.2 MDI/MDI-X Crossover (copper only)

The Ethernet controller PHY automatically determines whether or not it needs to cross over between pairs as shown in the following table so that an external crossover cable is not required. If the PHY interoperates with a device that cannot automatically correct for crossover, the Ethernet controller PHY makes the necessary adjustment prior to commencing Auto-Negotiation. If the PHY operates with a device that implements MDI/MDI-X crossover, a random algorithm as described in IEEE 802.3 clause 40.4.4 determines which device performs the crossover.

When the Ethernet controller PHY interoperates with legacy 10BASE-T devices that do not implement Auto-Negotiation, the PHY follows the same algorithm as described above since link pulses are present. However, when interoperating with legacy 100BASE-TX devices that do not implement Auto-Negotiation (link pulses are not present), the Ethernet controller PHY uses signal-detection to determine whether to crossover.

Auto MDI/MDI-X crossover is the default hardware configuration, but can be disabled via the PHY Specific Control Register bits 6:5 (PSCON).

The pin mapping in MDI/MDI-X modes are as follows:

Pin	MDI			MDIX		
	1000BASE-T	**100BASE-TX**	**10BASE-T**	**1000BASE-T**	**100BASE-TX**	**10BASE-T**
MDI[0]+/-	BI_DA +/-	TX +/-	TX +/-	BI_DB +/-	RX +/-	RX +/-
MDI[1]+/-	BI_DB +/-	RX +/-	RX +/-	BI_DA +/-	TX +/-	TX +/-
MDI[2]+/-	BI_DC +/-	unused	unused	BI_DD +/-	unused	unused
MDI[3]+/-	BI_DD +/-	unused	unused	BI_DC +/-	unused	unused

11.2.1 Polarity Correction (copper only)

The Ethernet controller PHY automatically corrects for polarity errors on the receive pairs in 1000BASE-T and 10BASE-T modes. In 100BASE-TX mode, the polarity does not matter.

In 1000BASE-T mode, receive polarity errors are automatically corrected based on the sequence of the symbols. Once the descrambler is locked, the polarity is also locked on all pairs. The polarity becomes unlocked only when the receiver loses lock.

11.2.2 10/100 Downshift (82540EP/EM Only)

Gigabit speed operation requires a 4-pair cable to operate. Some existing cables have only two pairs. Other cables might have 4 pairs, but one might be broken, leaving three working pairs. Over two or three pairs, two gigabit link partners might be able to successfully auto-negotiate 1000 Mbps speed, but then be unable to achieve link.

The downshift feature enables the **82540EP/EM** PHY to auto-negotiate with another gigabit link partner using a two or three pair cable and downshift to link at 100 Mbps or 10 Mbps, whichever is the highest speed below gigabit that the link partner is capable of.

By default, downshift is turned on. Refer to Table 13-39 to disable the downshift feature.

11.3 Cable Length Detection (copper only)

In 100/1000 Mbps operation, the Ethernet controller PHY attempts to indicate the approximate length of the CAT 5 cable attached. The estimated cable length is reported as one of the following ranges:

- <= 50 m
- 50 – 80 m
- 80 – 110 m
- 110 – 140 m
- >= 140 m

The estimated cable length can be obtained by reading the PHY Specific Status Register bits 9:7.

11.4 PHY Power Management (copper only)

The Ethernet controller PHY supports power-management based on either link status, MAC power-state, or both. During link-down states, the PHY utilizes its energy-detection capabilities to consume the least amount of power while still being capable of resuming link-up automatically. the Ethernet controller can be configured to automatically reduce PHY power during certain MAC D3 states by re-negotiating a low-speed link (the default behavior, but this can be disabled).

11.4.1 Link Down – Energy Detect (copper only)

When the link is operational and the Ethernet controller PHY detects loss of link, it initiates an Auto-Negotiation session. Failing to re-establish link via auto-negotiation, the PHY reverts to an Energy Detect power-down state. Following link loss, the PHY monitors receive energy on the wire. If the PHY detects energy on the wire, it starts to initiate Auto-Negotiation sending FLPs for five seconds. If at the end of five seconds the Auto-Negotiation has not completed, then the PHY stops sending FLPs and returns to monitoring receive energy. While monitoring receive energy, the PHY sends out a single 10 Mbps NLP (Normal Link Pulse) every one second in an attempt to wake up a connected device.

The above behavior is considered to be an advanced Energy-Detect (Energy Detect+) mode of PHY operation. The PHY can also be configured for a regular Energy Detect mode, which behaves similarly, except does not send out NLPs while monitoring receive energy. In this configuration, the Ethernet controller PHY can be woken by a connected device, but does not wake up the connected device itself.

The PHY Specific Control Register bits 9 and 8 described in Section 13.4.7.1.13 are used to configure the specific Energy-Detect mode of behavior for the PHY.

11.4.2 D3 State, No Link Required (copper only)

Each time the MAC transitions to a D3 or D0u power-state with no link required (wakeup disabled and no manageability enabled), the PHY enters its IEEE power-down mode, consuming the least amount of power possible. When powered-down, the PHY does not perform any form of Energy Detection, and does not generate any energy (NLPs) on the wire itself.

MAC transitions back to D0 power-states, either through explicit system/software mechanisms or by hardware reset operations, return the PHY back to a functional power-state. This will also re-initiate an auto-negotiation attempt advertising all speeds possible (10/100/1000 Mbps)[1], reverting to an Energy Detect state if unsuccessful in establishing link-up.

11.4.3 D3 Link-Up, Speed-Management Enabled (copper only)

If the MAC is configured for PHY power management (CTRL.EN_PHY_PWR_MGMT = 1b) and the PHY is linked at 1000 Mbps, then upon MAC transitions to D3 or D0u power-states where link IS required (either wakeup or manageability are enabled), the PHY re-initiates an Auto-Negotiation operation, advertising only 10/100 Mbps capability. This results in D3 operation at a lower-speed link and a reduced power level.

If a wakeup, management operation, or other system event causes the MAC to revert to fully-operational D0 state, the PHY initiates another Auto-Negotiation operation, advertising all 10/100/1000 Mbps speed capability, in order to return to maximum-speed operation.

11.4.4 D3 Link-Up, Speed-Management Disabled (copper only)

If the MAC is configured for no PHY power management (CTRL.EN_PHY_PWR_MGMT = 0b), and the MAC transitions to D3 power-states where link is required for either wakeup or manageability, then the PHY simply remains operational at its current line speed, without initiating a new Auto-Negotiation operation. This configuration is not recommended, since D3 power consumption at 1000 Mbps exceeds 20 mA Vaux.

1. Half duplex not supported.

11.5 Initialization

Note: Section 11.5 through Section 11.14 apply only to the **82541xx** and **82547GI/EI** Ethernet controllers.

At power-up or reset, the PHY core performs the initialization as shown in Figure 11-1. The software driver has access to the PHY register 0d, bits 15 and 11 for PHY reset and PHY Power Down control, respectively.

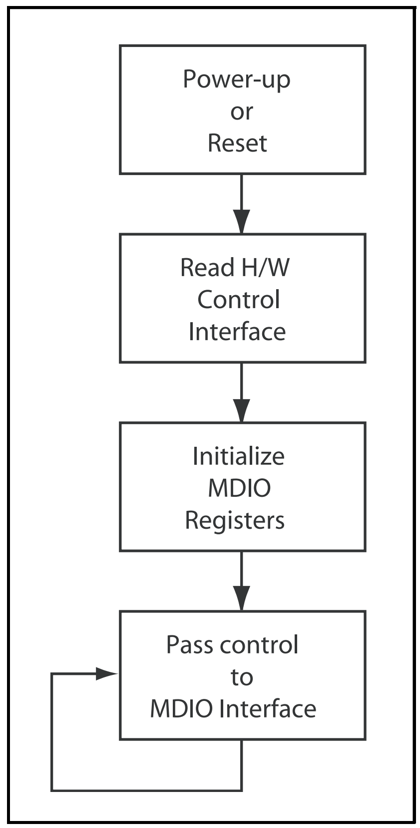


Figure 11-1. PHY Initialization Sequence

11.5.1 MDIO Control Mode

In the MDIO Control mode, the PHY uses the Hardware Control Interface to set up initial (default) values of the MDIO registers. Once initial values are set, bit control reverts to the MDIO interface.

The PHY can perform some low level initializations such as DSP configuration based upon EEPROM settings. The details of those initializations are reserved.

11.6 Determining Link State

The PHY and its link partner determine the type of link established through one of three methods:

- Auto-Negotiation
- Parallel Detection
- Forced Operation

Auto-Negotiation is the only method allowed by the 802.3ab standard for establishing a 1000BASE-T link, although forced operation could be used for test purposes. For 10/100 links, any of the three methods can be used. The sections that follow discuss each in greater detail.

Figure 11-2 provides an overview of link establishment. First the PHY checks if Auto-Negotiation is enabled. By default, the PHY supports Auto-Negotiation (PHY register 0, bit 12). If not, the PHY forces operation as directed. If Auto-Negotiation is enabled, the PHY begins transmitting Fast Link Pulses (FLPs) and receiving FLPs from its link partner. If FLPs are received by the PHY, Auto-Negotiation proceeds. It also can receive 100BASE-TX MLT3 and 10BASE-T Normal Link Pulses (NLPs). If either MLT3 or NLPs are received, it aborts FLP transmission and immediately brings up the corresponding half-duplex link.

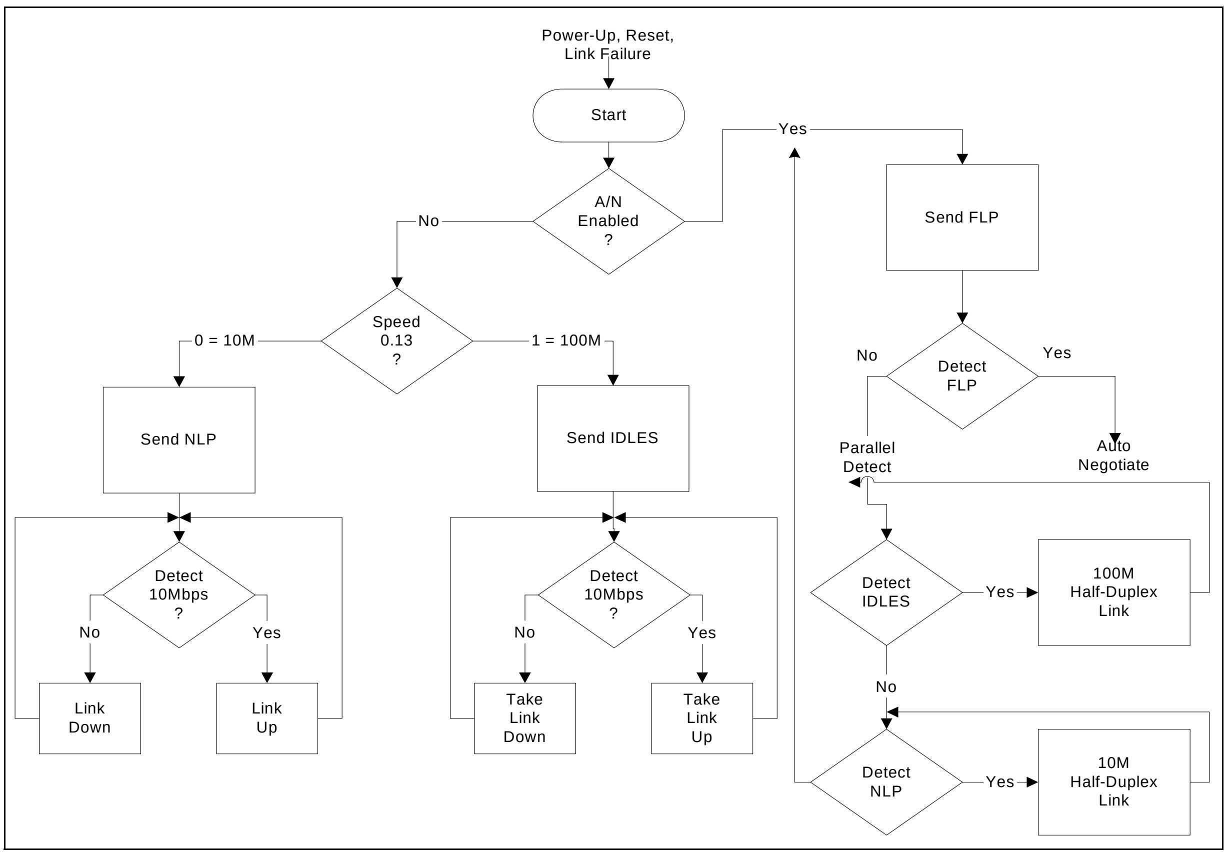


Figure 11-2. Overview of Link Establishment

11.6.1 False Link

When the PHY is first powered on, reset, or encounters a link down state, it must determine the line speed and operating conditions to use for the network link.

The PHY first checks the MDIO registers (initialized via the Hardware Control Interface or written by software) for operating instructions. Using these mechanisms, programmers can command the PHY to do one of the following:

- Force twisted-pair link operation to:
  - 1000T Full Duplex
  - 1000T Half Duplex
  - 100TX, Full Duplex
  - 100TX, Half Duplex
  - 10BASE-T, Full Duplex
  - 10BASET, Half Duplex
- Allow Auto-Negotiation/parallel-detection.

In the first six cases (forced operation), the PHY immediately begins operating the network interface as commanded. In the last case, the PHY begins the Auto-Negotiation/parallel-detection process.

11.6.2 Forced Operation

Forced operation can be used to establish 10 and 100 links, and 1000 links for test purposes. In this method, Auto-Negotiation is disabled completely and the link state of the PHY is determined by PHY register 0d.

Note: When speed is forced, the MDI/MDI-X crossover feature is not functional.

In forced operation, the programmer sets the link speed (10, 100, or 1000) and duplex state (full or half). For Gigabit (1000) links, the programmer must explicitly designate one side as the Master and the other as the Slave. Table 10-1 summarizes link establishment procedures.

Table 10-1. Determining Duplex State Via Parallel Detection

Configuration	Result
Both sides set for Auto-Negotiate.	Link is established via Auto-Negotiation.
Both sides set for forced operation.	No problem as long as duplex settings match.
One side set for Auto-Negotiation and the other for forced, half-duplex.	Link is established via parallel detect.
One side set for Auto-Negotiation and the other for forced full-duplex.	Link is established; however, sides disagree, resulting in transmission problems. Forced side is full-duplex, Auto-Negotiation side is half-duplex.

11.6.3 Auto Negotiation

The PHY supports the IEEE 802.3u Auto-Negotiation scheme with next page capability. Next Page exchange uses PHY register 7d to send information and PHY register 8d to receive them. Next Page exchange can only occur if both ends of the link advertise their ability to exchange Next Pages.

11.6.4 Parallel Detection

Parallel detection can only be used to establish 10 and 100 links. It occurs when the PHY tries to negotiate (transmit FLPs to its link partner), but instead of sensing FLPs from the link partner, it senses 100BASE-TX MLT3 code or 10BASE-T Normal Link Pulses (NLPs) instead. In this case, the PHY immediately stops Auto-Negotiation (terminates transmission of FLPs) and immediately brings up whatever link corresponds to what it has sensed (MLT3 or NLPs). If the PHY senses both of the technologies together, a parallel detection fault is detected and the PHY continues sending FLPs

With parallel detection, it is impossible to determine the true duplex state of the link partner, and the IEEE standard requires the PHY to assume a half-duplex link. Parallel detection also does not allow exchange of flow-control ability (PAUSE and ASM_DIR) or Master/Slave relationship required by 1000BASE-T. For this reason, parallel detection cannot be used to establish Gigabit Ethernet links.

11.7 Link Criteria

Once the link state is determined—via Auto-Negotiation, parallel detection or forced operation—the PHY and its link partner bring up the link.

11.7.1 1000BASE-T

For 1000BASE-T links, the PHY and its link partner enter a training phase. They exchange idle symbols and use the information gained to set their adaptive filter coefficients.

Either side indicates completion of the training phase to its link partner by changing the encoding of the idle symbols it transmits. When both sides so indicate, the link is up. Each side continues sending idle symbols whenever it has no data to transmit. The link is maintained as long as valid idle, data, or carrier extension symbols are received.

11.7.2 100BASE-TX

For 100BASE-TX links, the PHY and its link partner immediately begin transmitting idle symbols. Each side continues sending idle symbols whenever it has no data to transmit. The link is maintained as long as valid idle symbols or data is received.

In 100Mbps mode, the PHY establishes a link whenever the scrambler becomes locked and remains locked. Link will remain up unless the descrambler receives idles at less than a specified rate.

11.7.3 10BASE-T

For 10BASE-T links, the PHY and its link partner begin exchanging Normal Link Pulses (NLPs). The PHY transmits an NLP every 16 ms, and expects to receive one every 10 to 20 ms. The link is maintained as long as normal link pulses are received.

11.8 Link Enhancements

The PHY offers two enhanced link functions, each of which are discussed in the sections that follow:

- SmartSpeed
- Flow Control

11.8.1 SmartSpeed

SmartSpeed is an enhancement to Auto-Negotiation that enables the PHY to react intelligently to network conditions that prohibit establishment of a 1000BASE-T link, such as cable problems. Such problems might enable Auto-Negotiation to complete, but then inhibit completion of the training phase. Normally, if a 1000BASE-T link fails, the PHY returns to the Auto-Negotiation state with the same speed settings indefinitely. With SmartSpeed enabled, after five failed attempts, the PHY automatically downgrades the highest ability it advertises to the next lower speed: from 1000 to 100 to 10. Once a link is established, and if it is later broken, the PHY automatically upgrades the capabilities advertised to the original setting.

11.8.1.1 Using SmartSpeed

SmartSpeed is enabled by setting PHY register 16d, bit 7 to 1b. When SmartSpeed downgrades the PHY advertised capabilities, it sets bit 5 of PHY register 19. When link is established, its speed is indicated in PHY register 17, bits 15:14. SmartSpeed automatically resets the highest-level Auto-Negotiation abilities advertised, if link is established and then lost for more than two seconds.

11.8.2 Flow Control

Flow control enables congested nodes to pause traffic. MACs indicate their ability to implement flow control during Auto-Negotiation.

The PHY transparently supports MAC-to-MAC advertisement of flow control through its Auto-Negotiation process. Prior to Auto-Negotiation, the MAC indicates its flow control capabilities via PHY register 4d, bit 10 (Pause) and PHY register 4d, bit 11 (ASM_DIR). After Auto-Negotiation, the link partner’s flow control capabilities are indicated in PHY register 5d, bits 11:10.

Table 10-2 lists the intended operation for the various settings of ASM_DIR and Pause. This information is provided for reference only; it is the responsibility of the MAC to implement the correct function. The PHY merely enables the two MACs to communicate their abilities to each other.

Table 10-2. Pause And Asymmetric Pause Settings

ASM_DIR Settings Local (PHY Register 4d, Bit 10) and Remote (PHY Register 5d, Bit 10)	Pause Setting - Local (PHY Register 4d, Bit 9)	Pause Setting - Remote (PHY Register 5d, Bit 9)	Result
Both ASM_DIR = 1b	1b	1b	Symmetric - Either side can flow control the other
	1b	0b	Asymmetric - Remote can flow control local only
	0b	1b	Asymmetric - Local can flow control remote
	0b	0b	No flow control
Either or both ASM_DIR = 0b	1b	1b	Symmetric - Either side can flow control the other
	Either or both = 0b		No flow control

11.9 Management Data Interface

The PHY supports the IEEE 802.3 MII Management Interface also known as the Management Data Input/Output (MDIO) Interface. The MDIO interface consists of a physical connection to the MAC, a specific protocol which runs across the connection, and a 16-bit MDIO register set.

PHY Registers 0d through 10d and 15d are required and their functions are specified by the IEEE 802.3 specification. Additional registers are included for expanded functionality.

11.10 Low Power Operation

The Ethernet controller can be get into a low-power state according to MAC control (Power Management controls) or via PHY register 0d. In either power down mode, the Ethernet controller is not capable of receiving or transmitting packets.

11.10.1 Powerdown via the PHY Register

The PHY can be powered down using the control bit found in PHY register 0d, bit 11. This bit powers down a significant portion of the port but clocks to the register section remain active. This enables the PHY management interface to remain active during power-down. The power-down bit is active high. When the PHY exits software power-down (PHY register 0d, bit 11 = 1b), it re-initializes all analog functions, but retains its previous configuration settings**.**

11.10.2 Smart Power-Down

Smart Power-Down (SPD) is a link-disconnect capability applicable to all power management states, and is intended for mobile applications. Smart powerdown combines a power saving mechanism with the fact that link might disappear and resume.

SPD is enabled by PHY register 20d, bit 5 or by the *SPD Enable* bit in the EEPROM, and is entered when the PHY detects link lost. Auto-Negotiation must also be enabled. While in the SPD state, the PHY powers down circuits and clocks that are not required for detection of link activity. The PHY is still able to detect link pulses (including parallel detect) and wake up to engage in link negotiation. The PHY does not send link pulses (NLP) while in the SPD state. Register accesses are still possible.

Connecting a member of the family to another system with the SPD feature can lead to link failures if both ports are allowed to enter the SPD state.

11.11 1000 Mbps Operation

11.11.1 Introduction

This section provides an overview of 1000BASE-T functions, followed by discussion and review of the internal functional blocks shown in Figure 11-3.

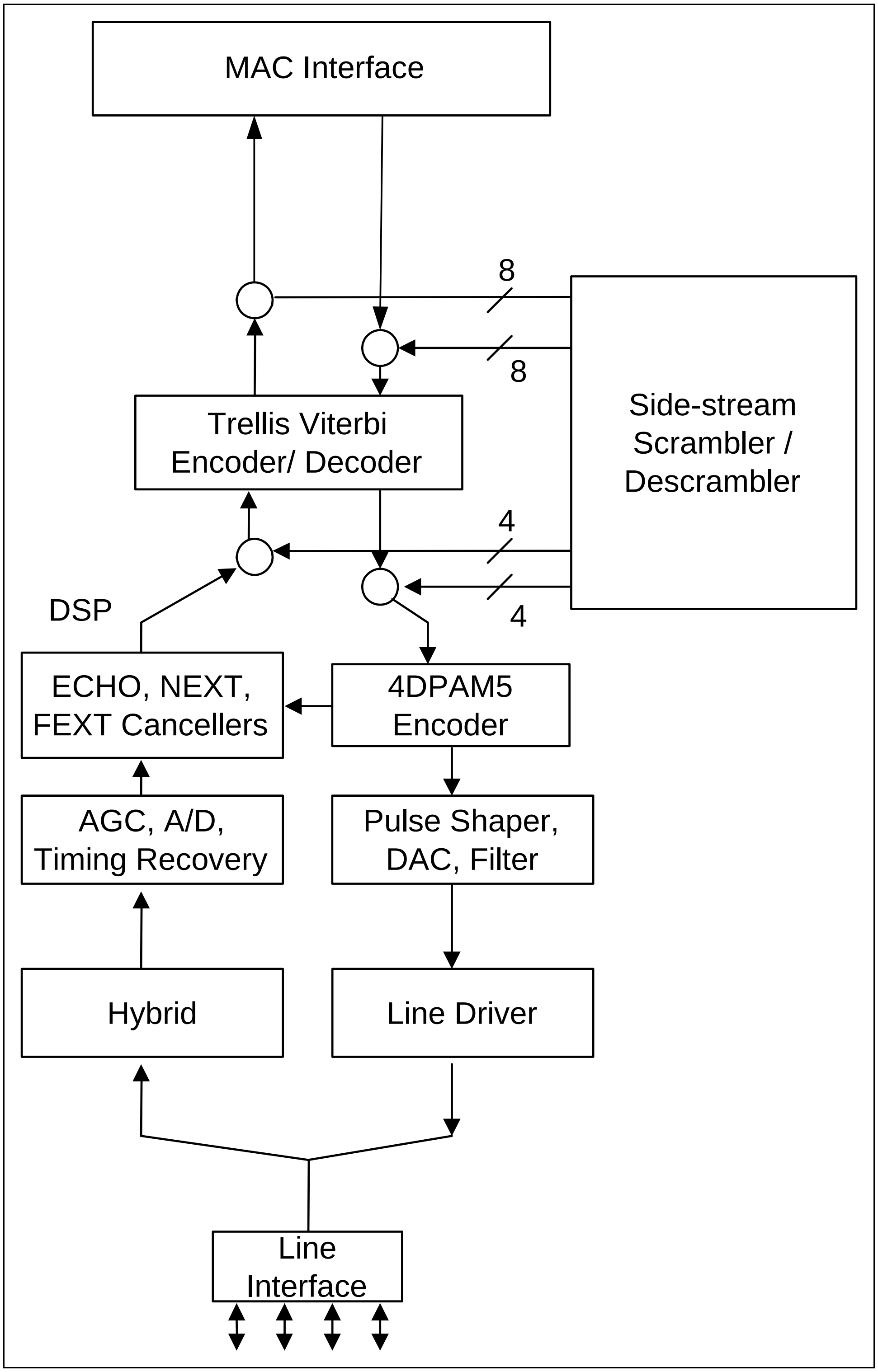


Figure 11-3. 1000 Base-T PHY Functions Overview

11.11.2 Transmit Functions

This section describes functions used when the Media Access Controller (MAC) transmits data through the PHY and out onto the twisted-pair connection.

11.11.2.1 Scrambler

The scrambler randomizes the transmitted data. The purpose of scrambling is two fold:

1. Scrambling eliminates repeating data patterns from the 4DPAM5 waveform to reduce EMI.
2. Each channel (A, B, C, D) gets a unique signature that the receiver uses for identification.

The scrambler is driven by a Linear Feedback Shift Register (LFSR), which is randomly loaded at power-up. The LFSR function used by the Master differs from that used by the Slave, giving each direction its own unique signature. The LFSR, in turn, generates uncorrelated outputs. These outputs randomize the inputs to the 4DPAM5 and Trellis encoders and randomize the sign of the 4DPAM5 outputs.

11.11.3 Transmit FIFO

The transmit FIFO re-synchronizes data transmitted by the MAC to the transmit reference used by the PHY.

11.11.3.1 Transmit Phase-Locked Loop PLL

This function generates the 125 MHz timing reference used by the PHY to transmit 4DPAM5 symbols. When the PHY is the Master side of the link, the crystal input is the reference for the transmit PLL. When the PHY is the Slave side of the link, the recovered receive clock is the reference for the transmit PLL.

11.11.3.2 Trellis Encoder

The Trellis Encoder uses the two high-order bits of data and its previous output to generate a ninth bit, which determines if the next 4DPAM5 pattern should be even or odd. This function provides forward error correction and enhances the signal-to-noise (SNR) ratio by a factor of 6 dB.

11.11.3.3 4DPAM5 Encoder

The 4DPAM5 encoder translates 8B codes transmitted by the MAC into 4DPAM5 symbols. The encoder operates at 125 Mhz, which is both the frequency of the MAC interface and the baud rate used by 1000BASE-T.

Each 8B code represents one of 256 data patterns. Each 4DPAM5 symbol consists of one of five signal levels (-2,-1,0,1,2) on each of the four twisted pair (A,B,C,D) representing 5^4 or 625 possible patterns per baud period. Of these, 113 patterns are reserved for control codes, leaving 512 patterns for data. These data patterns are divided into two groups of 256 even and 256 odd data patterns. As a result, each 8B octet has two possible 4DPAM5 representations—one even and one odd pattern.

11.11.3.4 Spectral Shaper

This function causes the 4DPAM5 waveform to have a spectral signature that is very close to that of the MLT3 waveform used by 100BASE-TX. This enables 1000BASE-T to take advantage of infrastructure (cables, magnetics) designed for 100BASE-TX.

The shaper works by transmitting 75% of a 4DPAM5 code in the current baud period, and adding the remaining 25% into the next baud period.

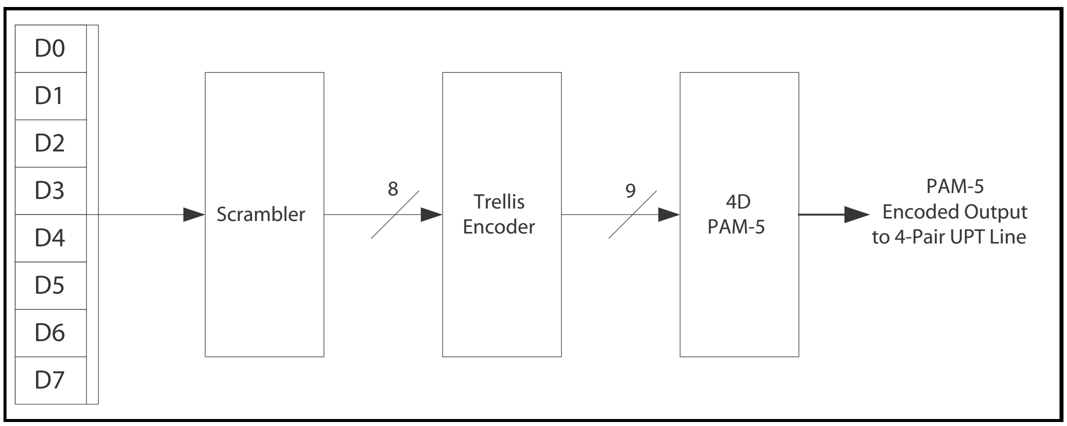
11.11.3.5 Low-Pass Filter

To aid with EMI, this filter attenuates signal components more than 180 Mhz. In 1000BASE-T, the fundamental symbol rate is 125 Mhz.

11.11.3.6 Line Driver

The line driver drives the 4DPAM5 waveforms onto the four twisted-pair channels (A, B, C, D), adding them onto the waveforms that are simultaneously being received from the link partner.

11.11.3.7 Transmit/Receive Flow



Scrambler Polynomials:

$1 + x^{13} + x^{33}$ (Master PHY Mode)

$1 + x^{20} + x^{33}$ (Slave PHY Mode)

Figure 11-4. 1000BASE-T Transmit Flow And Line Coding Scheme

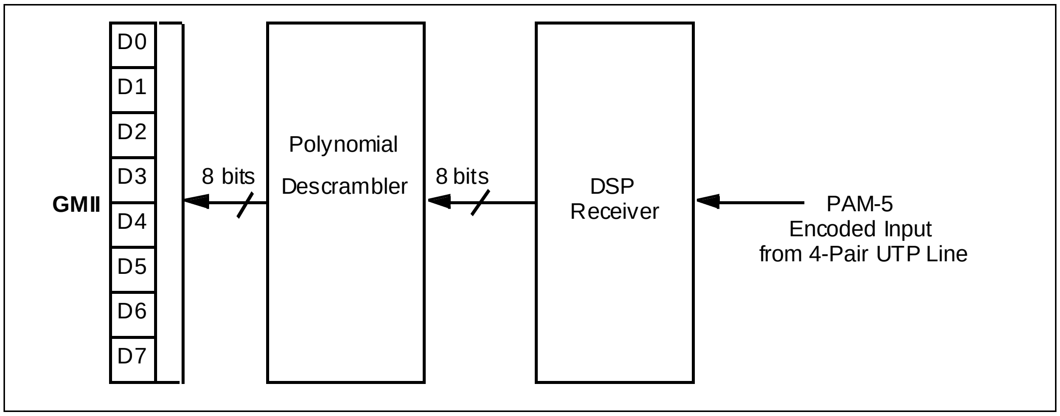


Figure 11-5. 1000BASE-T Receive Flow

11.11.4 Receive Functions

This section describes function blocks that are used when the PHY receives data from the twisted pair interface and passes it back to the MAC.

11.11.4.1 Hybrid

The hybrid subtracts the transmitted signal from the input signal, allowing the use of simple 100BASE-TX compatible magnetics.

11.11.4.2 Automatic Gain Control

The Automatic Gain Control (AGC) normalizes the amplitude of the received signal, adjusting for the attenuation produced by the cable.

11.11.4.3 Timing Recovery

This function re-generates a receive clock from the incoming data stream which is used to sample the data. On the Slave side of the link, this clock is also used to drive the transmitter.

11.11.4.4 Analog-to-Digital Converter

The Analog-to-Digital (ADC) function converts the incoming data stream from an analog waveform to digitized samples for processing by the DSP core.

11.11.4.5 Digital Signal Processor

The Digital Signal Processor (DSP) provides per-channel adaptive filtering, which eliminates various signal impairments including:

- Inter-symbol interference (equalization).
- Echo caused by impedance mismatch of the cable.
- Near-end crosstalk (NEXT) between adjacent channels (A, B, C, D).

- Far-end crosstalk (FEXT)
- Propagation delay variations between channels of up to 120 ns.
- Extraneous tones that have been coupled into the receive path.

The adaptive filter coefficients are initially set during the training phase. They are continuously adjusted (adaptive equalization) during operation through the decision-feedback loop.

11.11.4.6 Descrambler

The descrambler identifies each channel by its characteristic signature, removing the signature and re-routing the channel internally. In this way, the receiver can correct for channel swaps and polarity reversals. The descrambler uses the same base LFSR used by the transmitter on the other side of the link.

The descrambler requires approximately 15 µs. to lock, normally accomplished during the training phase.

11.11.4.7 Viterbi Decoder/Decision Feedback Equalizer (DFE)

The Viterbi decoder generates clean 4DPAM5 symbols from the output of the DSP. The decoder includes a Trellis encoder identical to the one used by the transmitter. The Viterbi decoder simultaneously looks at the received data over several baud periods. For each baud period, it predicts whether the symbol received should be even or odd, and compares that to the actual symbol received. The 4DPAM5 code is organized in such a way that a single level error on any channel changes an even code to an odd one and vice versa. In this way, the Viterbi decoder can detect single-level coding errors, effectively improving the Signal-To-Noise (SNR). When an error occurs, this information is quickly fed back into the equalizer to prevent future errors.

11.11.4.8 4DPAM5 Decoder

The 4DPAM5 decoder generates 8B data from the output of the Viterbi decoder.

11.12 100 Mbps Operation

The MAC passes data to the PHY over the MII. The PHY encodes and scrambles the data, then transmits it using MLT-3 for 100TX over copper. The PHY descrambles and decodes MLT-3 data received from the network. When the MAC is not actively transmitting data, the PHY sends out idle symbols on the line.

11.13 10 Mbps Operation

The PHY operates as a standard 10 Mbps transceiver. Data transmitted by the MAC as 4-bit nibbles is serialized, Manchester-encoded, and transmitted on the MDI[0]+/- outputs. Received data is decoded, de-serialized into 4-bit nibbles and passed to the MAC across the internal MII. The PHY supports all the standard 10 Mbps functions.

11.13.1 Link Test

In 10 Mbps mode, the PHY always transmits link pulses. If the Link Test Function is enabled, it monitors the connection for link pulses. Once it detects 2 to 7 link pulses, data transmission is enabled and remains enabled as long as the link pulses or data reception continues. If the link pulses stop, the data transmission is disabled.

If the Link Test function is disabled, the PHY might transmit packets regardless of detected link pulses. Setting PHY register 16d, bit 14 can disable the Link Test function.

11.13.2 10Base-T Link Failure Criteria and Override

Link failure occurs if Link Test is enabled and link pulses stop being received. If this condition occurs, the PHY returns to the Auto-Negotiation phase if Auto-Negotiation is enabled. Setting PHY register 16d, bit 14 disables the Link Integrity Test function, then the PHY transmits packets, regardless of link status.

11.13.3 Jabber

If the MAC begins a transmission that exceeds the jabber timer, the PHY disables the Transmit and loopback functions and asserts collision indication to the MAC. The PHY automatically exits jabber mode after 250-750 ms. This function can be disabled by setting PHY register 16d, bit 10 to 1b.

11.13.4 Polarity Correction

The PHY automatically detects and corrects for the condition where the receive signal (MDI_PLUS[0]/MDI_MINUS[0]) is inverted. Reversed polarity is detected if eight inverted link pulses, or four inverted end-of-frame markers, are received consecutively. If link pulses or data are not received for 96-130 ms, the polarity state is reset to a non-inverted state.

11.13.5 Dribble Bits

The PHY device handles dribble bits for all of its modes. If between one to four dribble bits are received, the nibble is passed across the interface. The data passed across is padded with 1b's if necessary. If between five to seven dribble bits are received, the second nibble is not sent onto the internal MII bus to the MAC. This ensures that dribble bits between 1-7 do not cause the MAC to discard the frame due to a CRC error.

11.14 PHY Line Length Indication

The PHY has a mechanism to deliver coefficient data for use in measuring cable length. If this capability is required, please contact your Intel representative for details.

Note: This page is intentionally left blank.

Dual Port Characteristics 12

12.1 Introduction[1]

The **82546GB/EB** architecture includes two instances of both the MAC and PHY (see Figure 2-1). With both MAC/PHY pairs operating, the Ethernet controller appears as a multi-function PCI device containing two identically-functioning devices. To avoid confusion, each MAC (when combined with either an internal PHY or an internal TBI transceiver/SerDes) is referred to as "LANx", where x = "A" or x = "B" to refer to each logical LAN device (LAN A or LAN B).

This section details specific features common to each MAC or PHY, resources/interfaces for which dedicated independent hardware/software interfaces exists for each LAN, as well as resources which are shared by both LAN devices.

The Ethernet controller normally appears to the system as a single, multi-function PCI device. It provides the ability to selectively disable one of the internal LAN functions, thereby allowing it to appear to the system as a single-function, single-LAN device. The mechanisms for controlling this behavior and the resulting appearance to the system are described in Section 12.5 entitled, "LAN Disable".

12.2 Features of Each MAC

The Ethernet controller is designed to have the capability to appear as two independent instances of a gigabit controller. The following section details major features that can be considered to be distinct features available to each Ethernet controller MAC independently.

12.2.1 PCI/PCI-X interface

The Ethernet controller contains a single physical PCI/PCI-X interface. The Ethernet controller is designed so that each of the logical LAN devices (LAN A and LAN B) appear as a distinct PCI/PCI-X bus device implementing, along with other registers, the following PCI device header space:

Byte Offset	Byte 0	Byte 1	Byte 2	Byte 3
0h	Device ID		Vendor ID	
4h	Status Register		Command Register	
8h	Class Code 020000h			Revision ID 00h
Ch	BIST 00h	Header Type 00h	Latency Timer	Cache Line Size
10h	Base Address 0			
14h	Base Address 1			
1h8	Base Address 2			
1Ch	Base Address 3			
20h	Base Address 4			

1. Section 12 only applies to the **82546GB/EB**.

Byte Offset	Byte 0	Byte 1	Byte 2	Byte 3
24h	Base Address 5			
28h	Cardbus CIS Pointer (not used)			
2Ch	Subsystem ID		Subsystem Vendor ID	
30h	Expansion ROM Base Address			
34h	Reserved			Cap_Ptr
38h	Reserved			
3Ch	Max_Latency 00h	Min_Grant FFh	Interrupt Pin 01h or 00h)	Interrupt Line 00h

Many of the fields of the PCI header space contain hardware default values that are either fixed or can be overridden using EEPROM, but cannot be independently specified for each logical LAN device. The following fields are considered to be common to both LAN devices:

Vendor ID	The Vendor ID of the Ethernet controller can be specified via EEPROM, but only a single value can be specified. The value is reflected identically for both LAN devices.
Revision	The revision number of the Ethernet controller is reflected identically for both LAN devices.
Header Type	This field indicates if a device is single function or multifunction. The value reflected in this field is reflected identically for both LAN devices, but the actual value reflected depends on LAN Disable configuration. When both Ethernet controller LAN ports are enabled, both PCI headers return 80h in this field, acknowledging being part of a multi-function device. LAN A exists as device “function 0”, while LAN B exists as device “function 1”. If one of the LAN ports is disabled, then only a single-function device is indicated (this field returns a value of 00h), and the LAN exists as device “function 0”.
Subsystem ID	The Subsystem ID of the Ethernet controller can be specified via EEPROM, but only a single value can be specified. The value is reflected identically for both LAN devices.
Subsystem Vendor ID	The Subsystem Vendor ID of the Ethernet controller can be specified via EEPROM, but only a single value can be specified. The value is reflected identically for both LAN devices.
Class Code, Cap_Ptr, Max Latency, Min Grant	These fields reflect fixed values that are constant values reflected for both LAN devices.

The following fields are implemented unique to each LAN device:

Device ID	The Device ID reflected for each LAN device can be independently specified via EEPROM.
Command, Status	Each LAN device implements its own command/status registers.
Latency Timer, Cache Line Size	Each LAN device implements these registers uniquely. The system should program these fields identically for each LAN to ensure consistent behavior and performance of each device.
Memory BAR, Flash BAR, IO BAR, Expansion ROM BAR	Each LAN device implements its own Base Address registers, allowing each device to claim its own address region(s).
Interrupt Pin	Each LAN device independently indicates which interrupt pin (INTA# or INTB#) is used by that Ethernet controller's MAC to signal system interrupts. The value for each LAN device can be independently specified via EEPROM, but only if both LAN devices are enabled.

12.2.2 MAC Configuration Register Space

All device control/status registers detailed in Section 13.4, Main Register Descriptions, are implemented per-LAN device. Each LAN device can be accessed using memory or I/O cycles, depending on the specific BAR setting(s) established for that LAN device.

Register accesses to each MAC instance are independent. In PCI bus operation, a register access to one LAN which is retried as a delayed-read requires subsequent accesses to that LAN to retry the read identically until complete. An outstanding delayed-read for one LAN device does not impact the Ethernet controller's ability to accept a register access to the other LAN. Similarly, in PCI-X bus operation, and register access resulting in a split & split-completion by one LAN device in no way prevents the other LAN device from accepting and servicing (or splitting) an access to its register space.

12.2.3 SDP, LED, INT# output

Each LAN device provides an independent set of LED outputs and software-programmable I/O pins (SDP). Four LED outputs and four SDP pins are provided per LAN device. These pins and their function are bound to a specific LAN device (eight SDP pins cannot be associated with a single LAN device, for example).

Each LAN device can use a dedicated pin for signalling interrupts to the system. Two pins, INTA# and INTB#, exist on the Ethernet controller to signal interrupts by the different LAN devices. The specific pin used by each LAN is configurable when both LAN devices are enabled.

12.3 Shared EEPROM

The Ethernet controller uses a single EEPROM device to configure hardware default parameters for both LAN devices, including Ethernet Individual Addresses (IA), LED behaviors, receive packet-filters for manageability and wakeup capability, etc. Certain EEPROM words are used to specify hardware parameters which are LAN device-independent (such as those that affect circuits behavior). Other EEPROM words are associated with a specific LAN device. LAN A and LAN B accesses the EEPROM to obtain their respective configuration settings.

12.3.1 EEPROM Map

The EEPROM map identifies those words configuring both LAN devices or the entire Ethernet controller component as “LAN A/B Shared”. Those words configuring a specific LAN device parameters are identified as either “LAN A” or “LAN B”.

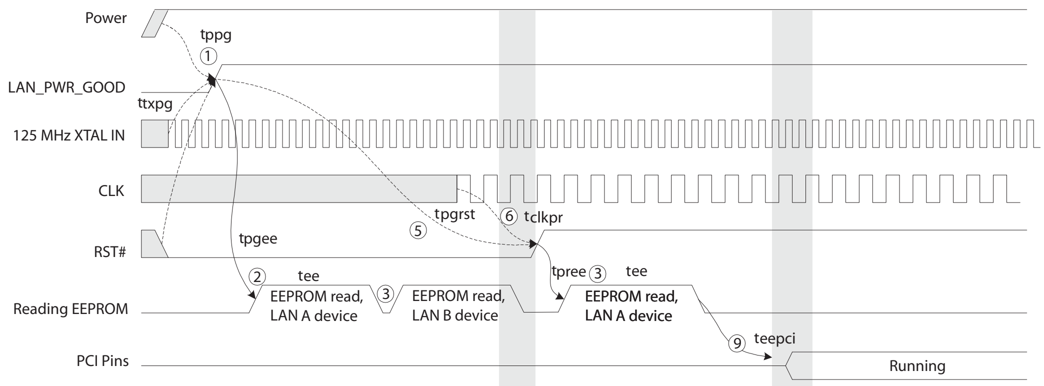
The following EEPROM words warrant additional notes specifically related to dual-LAN support:

Ethernet Address (IA) (LAN A/B shared)	The EEPROM specifies the IA associated with the LAN A device and used as the hardware default of the Receive Address Registers for that device. The hardware-default IA for the LAN B device is automatically determined by the same EEPROM word, and is set to the value of {Ethernet IA LAN A with its least significant bit inverted}.
Initialization Control 1, Initialization Control 2 (LAN A/B shared)	These EEPROM words specify hardware-default values for parameters that apply a single value to both LAN devices, such as link configuration parameters required for auto-negotiation, wakeup settings, PCI/PCI-X bus advertised capabilities, etc.
Initialization Control 3 (LAN A, LAN B unique)	This EEPROM word configures default values associated with each LAN device’s hardware connections, including which link mode (internal PHY, external TBI SerDes) is used with this LAN device. Because a separate EEPROM word configures the defaults for each LAN, extra care must be taken to ensure that the EEPROM image does not specify a resource conflict. For example, multiple LAN devices both attempting to utilize the external TBI transceiver interface at once.
Management Control	This EEPROM word configures manageability parameters. Note that this word controls whether an internal ASF controller is enabled/disabled for this LAN, and whether the SMBus is enabled/disabled for this LAN. Extra care must be taken to ensure that the EEPROM image does not specify a resource conflict – if an internal ASF controller is being used, it can only be enabled for a single LAN device. The SMBus can only be enabled for a single LAN device.

12.3.2 EEPROM Arbitration

The Ethernet controller uses a single EEPROM to store hardware configuration words for both LAN devices. The words used by each specific LAN device are noted in the EEPROM map. Each LAN device obtains its EEPROM configuration parameters by performing its own independent EEPROM read. Each LAN device reads the entire EEPROM image, verifying the EEPROM signature, and applying the word(s) appropriate for the specific LAN device. The Ethernet controller internally arbitrates between EEPROM access by the two LAN devices, to ensure that each device is able to perform a complete, uninterrupted EEPROM read sequence.

The result of multiple LAN devices' reading EEPROM is that power-on and reset-initiated EEPROM read sequences might appear slightly differently from the sequences illustrated during the discussion of power-state transitions (Section 6.3.2). Those illustrations indicate EEPROM read periods without distinguishing between reads by LAN A versus LAN B devices. At initial power-on, both LAN devices always execute an EEPROM read sequence. However, since the enabling/disabling of a particular LAN device occurs on the deassertion of PCI reset, the post-reset EEPROM read sequence(s) are only performed by LAN device(s) that are enabled. The following illustration more clearly illustrates the EEPROM read sequence for a scenario where a single LAN device is enabled:



12.4 Shared FLASH

The Ethernet controller provides an interface to an external FLASH/ROM memory device, as described in Section 7. This FLASH/ROM device can be mapped into memory and/or I/O address space for each LAN device through the use of PCI Base Address Registers (BARs). Bit 3 of the EEPROM Initialization Control Word 3 associated with each LAN device selectively disables/enables whether the FLASH can be mapped for each LAN device by controlling the BAR register advertisement and writeability.

12.4.1 FLASH Access Contention

Unlike the shared EEPROM implementation, the Ethernet controller does NOT implement any internal arbitration between FLASH accesses initiated through the LAN A device and those initiated through the LAN B device. If accesses from both LAN devices are initiated during the same approximate time window, access contention can occur. If contention occurs, the external FLASH addresses can be corrupted or unstable throughout the access. During writes to FLASH, contention can result in corrupt or unstable data values; contention during reads can result in erroneous read data being returned.

Note: Access contention to FLASH by both LAN devices is more than likely to result in indeterminate data results (during read transactions), corrupted FLASH (during write transactions), or other unpredictable behavior.

To avoid this contention, accesses from both LAN devices MUST be synchronized using external software synchronization of the memory or I/O transactions responsible for the access. It might be possible to ensure contention-avoidance simply by nature of software sequentially.

12.5 LAN Disable

For a LOM design, it might be desirable for the system to provide BIOS-setup capability for selectively enabling or disabling LOM devices. This might allow an end-user more control over system resource-management, avoid conflicts with add-in NIC solutions, etc. The Ethernet controller provides support for selectively enabling or disabling one or both LAN device(s) in the system.

12.5.1 Overview

Device presence (or non-presence) must be established early during BIOS execution in order to ensure that BIOS resource-allocation (of interrupts, of memory or IO regions) is done according to devices that are present only. This is frequently accomplished using a BIOS CVDR (Configuration Values Driven on Reset) mechanism. The Ethernet controller LAN-disable mechanism is implemented in order to be compatible with such a solution. The Ethernet controller samples two pins (FLASH data pins, bits 1 and 0) on reset to determine the LAN-enable configuration.

When a particular LAN is disabled, all internal clocks to that LAN are disabled, the device is held in reset, and the internal PHY for that LAN is powered-down. The device does not respond to PCI configuration cycles. Effectively, the LAN device becomes invisible to the system from both a configuration and power-consumption standpoint.

Note: Since the LAN-disable mechanisms is implemented using the FLASH data pins, this mechanism can only be used when no FLASH device is present (FLASH disabled). An Ethernet controller-based NIC built with support for a FLASH device always enables both LAN devices.

12.5.2 Values Sampled on Reset

The Ethernet controller samples values from the pins FLSH_DATA[1] and FLSH_DATA[0] on the rising edge of LAN_PWR_GOOD and RST#. Based on the values sampled, the LAN devices are enabled/disabled according to the following table:

Pin sampled	LAN device controlled	Enable/Disable
FLSH_DATA[0]	LAN A device	Vcc/logic 1b = enabled Vss/logic 0b = disabled
FLSH_DATA[1]	LAN B device	???

12.5.3 Multi-Function Advertisement

If one of the LAN devices is disabled, the Ethernet controller no longer is a multi-function device. It normally reports a 01h in the PCI Configuration Header field *Header Type*, indicating multi-function capability. However, if a LAN id disabled, it reports a 0h in this filed to signify single-function capability.

12.5.4 Interrupt Use

When both LAN devices are enabled, the Ethernet controller uses both the INTA# and INTB# pins for interrupt-reporting. The EEPROM Initialization Control Word 3 (bit 4) associated with each LAN device controls which of these two pins is used for each LAN device. The specific interrupt pin used is reported in the PCI Configuration Header *Interrupt Pin* field associated with each LAN device.

However, if either LAN device is disabled, then the INTA# be used for the remaining LAN device, regardless of the EEPROM configuration. Under these circumstances, the *Interrupt Pin* field of the PCI Header always reports a value of 1h, indicating INTA# usage.

12.5.5 Power Reporting

When both LAN devices are enabled, the PCI Power Management Register Block has the capability of reporting a Common Power value. The Common Power value is reflected in the data field of the PCI Power Management registers. The value reported as Common Power is specified via EEPROM, and is reflected in the data field each time the *Data_Select* field has a value of 8h (8h = Common Power Value Select).

When either LAN is disabled and the Ethernet controller appears as a single-function device, the Common Power value, if selected, reports 0h (undefined value), as Common Power is undefined for a single-function device.

12.5.6 Summary

The following table lists the various LAN enabled/disabled configurations possible:

CVDR values sampled-on-reset		LAN device	Enabled/ Disabled	PCI function	Interrupt Line Used
FLSH_DATA[1]	**FLSH_DATA[0]**				
1	1	A	√ (enabled)	0	INTA# or INTB# (specified by LAN A EEPROM *InitCtrl3* value)
		B	√ (enabled)	1	INTA# or INTB# (specified by LAN B EEPROM *InitCtrl3* value)
0	1	A	÷	0	INTA#
		B	X (disabled)	n/a	n/a
1	0	A	C	n/a	n/a
		B	X	0	INTA#
0	0	A	X	n/a	n/a
		B	X	n/a	n/a

Register Descriptions 13

13.1 Introduction

This section details the state inside the PCI/PCI-X Family of Gigabit Ethernet Controllers that are visible to the programmer. In some cases, it describes hardware structures invisible to software in order to clarify a concept.

The address space within the Ethernet controller is divided up into eight main categories:

- PCI
- General Configuration and Wakeup
- Interrupt
- MAC Receive
- MAC Transmit
- PHY Receive, Transmit and Special Function
- Statistics
- Diagnostic State (not used in normal operation)

The Ethernet controller's address space is mapped into four regions with PCI Base Address Registers described in Table 13-2. These regions are shown as follows.

Internal registers and memories (including PHY)	Memory	128 KB
Flash (optional)	Memory	64 - 512 KB
Expansion ROM (optional)	Memory	64 - 512 KB
Internal registers and memories, Flash (optional)	I/O	8 Bytes

Both the Flash an Expansion ROM Base Address Registers map the same Flash memory. The internal registers and memories and Flash can be access through I/O space by doing a level of indirection, as explained later.

Note: The PHY registers are accessed indirectly through the MDI/O interface described in Section 8.2.

13.2 Register Conventions

All registers in the Ethernet controller are defined to be 32 bits, should be accessed as 32-bit double words, and are aligned on a 64-bit boundary. There are exceptions to this rule:

- PCI configuration registers
- I/O space registers (IOADDR and IODATA) are aligned on 32-bit boundaries
- Register pairs where two 32-bit registers make up a larger logical size
- Accesses to Flash memory (through Expansion ROM space or secondary Base Address Register space) can be byte, word, double word or quadword accesses.
- **Reserved bit positions.** Some registers contain certain bits that are marked as "reserved." These bits should never be set to a value of 1b by software. Reads from registers containing reserved bits can return indeterminate values in the reserved bit positions unless read values are explicitly stated. When read, these reserved bits should be ignored by software.

- **Reserved and/or undefined addresses.** Any register not explicitly declared in this specification should be considered to be reserved and should not be written. Writing to reserved or undefined register addresses can cause indeterminate behavior. Reads from reserved or undefined configuration register addresses can return indeterminate values unless read values are explicitly stated for specific addresses.
- **Initial values.** Most registers define the initial hardware values prior to being programmed. In some cases, hardware initial values are undefined and are listed as such via the text "undefined," "unknown," or "X." Some such values might need setting through EEPROM configuration or software in order for proper operation to occur; this need is dependent on the function of the bit. Other registers might cite a hardware default that is overridden by a higher precedence operation. Operations that might supersede hardware defaults can include a valid EEPORM load, completion of a hardware operation (such as hardware Auto-Negotiation), or writing of a different register whose value is then reflected in another bit.

For registers that should be accessed as 32-bit double words, partial writes (less than a 32-bit double word) is ignored. Partial reads return all 32 bits of data regardless of the byte enables.

Partial reads to read-on-clear registers (for example, ICR) can have unexpected results since all 32 bits are actually read regardless of the byte enables. Partial reads should not be performed.

All statistics registers are implemented as 32-bit registers. 64-bit accesses to these registers must have the upper byte enables deasserted. 32-bit registers with addresses not on a quadword boundary cannot be accessed through a 64-bit access.

Note: The PHY registers are accessed indirectly through the MDI/O interface.

13.2.1 Memory and I/O Address Decoding

13.2.1.1 Memory-Mapped Access to Internal Registers and Memories

The internal registers and memories can be accessed as direct memory-mapped offsets from the base address register (BAR0 or BAR0/BAR1, see Section 4.1). Refer to Table 13-2 for the appropriate offset for each specific internal register.

13.2.1.2 Memory-Mapped Access to FLASH

The external Flash can be accessed using direct memory-mapped offsets from the Flash base address register (BAR1 or BAR2/BAR3, see Section 4.1). The Flash is only accessible if enabled through the EEPROM Initialization Control Word, and if the Flash Base Address register contains a valid (non-zero) base memory address. For accesses, the offset from the Flash BAR corresponds to the offset into the flash actual physical memory space.

13.2.1.3 Memory-Mapped Access to Expansion ROM

The external Flash can also be accessed as a memory-mapped expansion ROM. Accesses to offsets starting from the Expansion ROM Base address (see Section 4.1) reference the Flash provided that access is enabled through the EEPROM Initialization Control Word, and if the Expansion ROM Base Address register contains a valid (non-zero) base memory address.

13.2.2 I/O-Mapped Internal Register, Internal Memory, and Flash[1]

To support pre-boot operation (prior to the allocation of physical memory base addresses), all internal registers, memories, and Flash can be accessed using I/O operations. I/O accesses are supported only if an I/O Base Address is allocated and mapped (BAR2 or BAR4, see Section 4.1), the BAR contains a valid (non-zero value), and I/O address decoding is enabled in the PCI/PCIX configuration.

When an I/O BAR is mapped, the I/O address range allocated opens a 32-byte window in the system I/O address map. Within this window, two I/O addressable registers are implemented: IOADDR and IODATA. The IOADDR register is used to specify a reference to an internal register, memory, or Flash, and then the IODATA register is used as a window to the register, memory or Flash address specified by IOADDR:

Offset	Abbreviation	Name	RW	Size
00000000h	IOADDR	Internal Register, Internal Memory, or Flash Location Address 00000h - 1FFFFh — Internal Registers and Memories 20000h - 7FFFFh — Undefined 80000h - FFFFFh — Flash	RW	4 bytes
00000004h	IODATA	Data field for reads or writes to the Internal Register Internal Memory, or Flash location as identified by the current value in IOADDR. All 32 bits of this register are read/write-able.	RW	4 bytes

13.2.2.1 IOADDR

The IOADDR register must always be written as a DWORD access (for example, the C/BE#[3:0] byte enables must all be enabled). Writes that are less than 32 bits are ignored. Reads of any size return a DWORD of data. However, the chipset or CPU can only return a subset of that DWORD.

For Intel architecture programmers, the IN and OUT instructions must be used to cause I/O cycles to be used on the PCI bus. Since writes must be to a 32-bit quantity, the source register of the OUT instruction must be EAX (the only 32-bit register supported by the OUT command). For reads, the IN instruction can have any size target register, but it is recommended that the 32-bit EAX register be used.

Since only a particular range is addressable, the upper bits of this register are hard coded to 0b. Bits 31 through 20 are not write-able and always read back as 0b.

At hardware reset (LAN_PWR_GOOD) or PCI Reset, this register value resets to 00h. Once written, the value is retained until the next write or reset.

13.2.2.2 IODATA

The IODATA register must always be written as a DWORD access when the IOADDR register contains a value for the Internal Register and Memories (00000h - 1FFFCh). In this case, writes less than 32 bits are ignored.

1. Not applicable to the **82547GI/EI**.

The IODATA register can be written as a byte, word, or Dword access when the IOADDR register contains a value for the Flash (80000h - FFFFFh). In this case, the value in IOADDR must be properly aligned to the data value. Additionally, the lower 2 bits of the IODATA PCI-X access must correspond to the byte, word, or Dword access.Table 13-1 lists the supported configurations:

Table 13-1. IODATA Register Configurations

Access Type	Ethernet Controller IOADDR Register Bits [1:0]	PCI-X[1] IODATA Access AD[1:0] Bits in Address Phase	PCI-X IODATA Access AD C/BE#[3:0] Bits in Data Phase
BYTE (8 bits)	00b	00b	1110b
	01b	01b	1101b
	10b	10b	1011b
	11b	11b	0111b
WORD (16 bits)	00b	00b	1100b
	10b	10b	0011b
DWORD (32 bits)	00b	00b	0000b

1. The **82540EP/EM** does not support PCI-X.

Software might need to implement special code to access the Flash memory at a byte or word at a time. Example code that reads a Flash byte is shown here to illustrate the impact of Table 13-1:

```
char *IOADDR;
char *IODATA;

IOADDR = IOBASE + 0;
IODATA = IOBASE + 4;

*(IOADDR) = Flash_Byte_Address;
Read_Data = *(IODATA + (Flash_Byte_Address % 4));
```

Reads to IODATA of any size returns a Dword of data. However, the chipset or CPU can only return a subset of that Dword.

For Intel architecture programmers, the IN and OUT instructions must be used to cause I/O cycles to be used on the PCI bus. Where 32-bit quantities are required on writes, the source register of the OUT instruction must be EAX (the only 32-bit register supported by the OUT command).

Writes and reads to IODATA when the IOADDR register value is in an undefined range (20000h - 7FFFCh) should not be performed. Results are indeterminate.

There are no special software timing requirements on accesses to IOADDR or IODATA. All accesses are immediate except when data is not readily available or acceptable. In this case, the Ethernet controller delays the results through normal bus methods.

Table 13-2. Ethernet Controller Register Summary

Category	Offset	Abbreviation	Name	R/W	Page
General	00000h	CTRL	Device Control	R/W	224
General	00008h	STATUS	Device Status	R	229
General	00010h	EECD	EEPROM/Flash Control/Data	R/W	232
General	00014h	EERD	EEPROM Read (not applicable to the **82544GC/EI**)	R/W	234
General	0001Ch	FLA	Flash Access (applicable to the **82541xx** and **82547GI/EI** only)	R/W	236
General	00018h	CTRL_EXT	Extended Device Control	R/W	237
General	00020h	MDIC	MDI Control	R/W	242
General	00028h	FCAL	Flow Control Address Low	R/W	283
General	0002Ch	FCAH	Flow Control Address High	R/W	283
General	00030h	FCT	Flow Control Type	R/W	284
General	00038h	VET	VLAN EtherType	R/W	284
General	00170h	FCTTV	Flow Control Transmit Timer Value	R/W	285
General	00178h	TXCW	Transmit Configuration Word (not applicable to the **82540EP/EM**, **82541xx** and **82547GI/EI**)	R/W	286
General	00180h	RXCW	Receive Configuration Word (not applicable to the **82540EP/EM**, **82541xx** and **82547GI/EI**)	R	287
General	00E00h	LEDCTL	LED Control (not applicable to the **82544GC/EI**)	R/W	289
DMA	01000h	PBA	Packet Buffer Allocation	R/W	292
Interrupt	000C0h	ICR	Interrupt Cause Read	R	293
Interrupt	000C4h	ITR	Interrupt Throttling (not applicable to the **82544GC/EI**)	R/W	295
Interrupt	000C8h	ICS	Interrupt Cause Set	W	296
Interrupt	000D0h	IMS	Interrupt Mask Set/Read	R/W	297
Interrupt	000D8h	IMC	Interrupt Mask Clear	W	298
Receive	00100h	RCTL	Receive Control	R/W	300
Receive	02160h	FCRTL	Flow Control Receive Threshold Low	R/W	304
Receive	02168h	FCRTH	Flow Control Receive Threshold High	R/W	305
Receive	02800h	RDBAL	Receive Descriptor Base Low	R/W	306
Receive	02804h	RDBAH	Receive Descriptor Base High	R/W	306
Receive	02808h	RDLEN	Receive Descriptor Length	R/W	307
Receive	02810h	RDH	Receive Descriptor Head	R/W	307
Receive	02818h	RDT	Receive Descriptor Tail	R/W	308
Receive	02820h	RDTR	Receive Delay Timer	R/W	308
Receive	0282Ch	RADV	Receive Interrupt Absolute Delay Timer (not applicable to the **82544GC/EI**)	R/W	309
Receive	02C00h	RSRPD	Receive Small Packet Detect Interrupt (not applicable to the **82544GC/EI**)	R/W	310
Transmit	00400h	TCTL	Transmit Control	R/W	310
Transmit	00410h	TIPG	Transmit IPG	R/W	312
Transmit	00458h	AIFS	Adaptive IFS Throttle - AIT	R/W	314
Transmit	03800h	TDBAL	Transmit Descriptor Base Low	R/W	315
Transmit	03804h	TDBAH	Transmit Descriptor Base High	R/W	316
Transmit	03808h	TDLEN	Transmit Descriptor Length	R/W	316
Transmit	03810h	TDH	Transmit Descriptor Head	R/W	317
Transmit	03818h	TDT	Transmit Descriptor Tail	R/W	318
Transmit	03820h	TIDV	Transmit Interrupt Delay Value	R/W	318

Category	Offset	Abbreviation	Name	R/W	Page
TX DMA	03000h	TXDMAC	TX DMA Control (applicable to the **82544GC/EI** only)	R/W	319
TX DMA	03828h	TXDCTL	Transmit Descriptor Control	R/W	319
TX DMA	0282Ch	TADV	Transmit Absolute Interrupt Delay Timer (not applicable to the **82544GC/EI**)	R/W	321
TX DMA	03830h	TSPMT	TCP Segmentation Pad and Threshold	R/W	322
RX DMA	02828h	RXDCTL	Receive Descriptor Control	R/W	324
RX DMA	05000h	RXCSUM	Receive Checksum Control	R/W	325
Receive	05200h-053FCh	MTA[127:0]	Multicast Table Array (n)	R/W	327
Receive	05400h-05478h	RAL(8∗n)	Receive Address Low (n)	R/W	329
Receive	05404h-0547Ch	RAH(8∗n)	Receive Address High (n)	R/W	329
Receive	05600h-057FCh	VFTA[127:0]	VLAN Filter Table Array (n) Not applicable to the **82541ER**	R/W	330
Wakeup	05800h	WUC	Wakeup Control	R/W	331
Wakeup	05808h	WUFC	Wakeup Filter Control	R/W	332
Wakeup	05810h	WUS	Wakeup Status	R	333
Wakeup	05838h	IPAV	IP Address Valid	R/W	335
Wakeup	05840h-05858h	IP4AT IPAT (**82544GC/EI**)	IPv4 Address Table IP Address Table (**82544GC/EI**)	R/W	336
Wakeup	05880h-0588Ch	IP6AT	IPv6 Address Table (not applicable to the **82544GC/EI**)	R/W	337
Wakeup	05900h	WUPL	Wakeup Packet Length	R/W	338
Wakeup	05A00h-05A7Ch	WUPM	Wakeup Packet Memory	R/W	338
Wakeup	05F00h-05F18h	FFLT	Flexible Filter Length Table	R/W	338
Wakeup	09000h-093F8h	FFMT	Flexible Filter Mask Table	R/W	339
Wakeup	09800h-09BF8h	FFVT	Flexible Filter Value Table	R/W	340
Statistics	04000h	CRCERRS	CRC Error Count	R	341
Statistics	04004h	ALGNERRC	Alignment Error Count	R	341
Statistics	04008h	SYMERRS	Symbol Error Count	R	342
Statistics	0400Ch	RXERRC	RX Error Count	R	342
Statistics	04010h	MPC	Missed Packets Count	R	343
Statistics	04014h	SCC	Single Collision Count	R	343
Statistics	04018h	ECOL	Excessive Collisions Count	R	344
Statistics	0401Ch	MCC	Multiple Collision Count	R	344
Statistics	04020h	LATECOL	Late Collisions Count	R	345
Statistics	04028h	COLC	Collision Count	R	345
Statistics	04030h	DC	Defer Count	R	346
Statistics	04034h	TNCRS	Transmit - No CRS	R	346
Statistics	04038h	SEC	Sequence Error Count	R	347
Statistics	0403Ch	CEXTERR	Carrier Extension Error Count	R	347
Statistics	04040h	RLEC	Receive Length Error Count	R	348
Statistics	04048h	XONRXC	XON Received Count	R	348
Statistics	0404Ch	XONTXC	XON Transmitted Count	R	349
Statistics	04050h	XOFFRXC	XOFF Received Count	R	349

Category	Offset	Abbreviation	Name	R/W	Page
Statistics	04054h	XOFFTXC	XOFF Transmitted Count	R	349
Statistics	04058h	FCRUC	FC Received Unsupported Count	R/W	350
Statistics	0405Ch	PRC64	Packets Received (64 Bytes) Count	R/W	350
Statistics	04060h	PRC127	Packets Received (65-127 Bytes) Count	R/W	351
Statistics	04064h	PRC255	Packets Received (128-255 Bytes) Count	R/W	351
Statistics	04068h	PRC511	Packets Received (256-511 Bytes) Count	R/W	352
Statistics	0406Ch	PRC1023	Packets Received (512-1023 Bytes) Count	R/W	352
Statistics	04070h	PRC1522	Packets Received (1024-Max Bytes)	R/W	353
Statistics	04074h	GPRC	Good Packets Received Count	R	353
Statistics	04078h	BPRC	Broadcast Packets Received Count	R	354
Statistics	0407Ch	MPRC	Multicast Packets Received Count	R	354
Statistics	04080h	GPTC	Good Packets Transmitted Count	R	355
Statistics	04088h	GORCL	Good Octets Received Count (Low)	R	355
Statistics	0408Ch	GORCH	Good Octets Received Count (Hi)	R	355
Statistics	04090h	GOTCL	Good Octets Transmitted Count (Low)	R	356
Statistics	04094h	GOTCH	Good Octets Transmitted Count (Hi)	R	356
Statistics	040A0h	RNBC	Receive No Buffers Count	R	356
Statistics	040A4h	RUC	Receive Undersize Count	R	357
Statistics	040A8h	RFC	Receive Fragment Count	R	357
Statistics	040ACh	ROC	Receive Oversize Count	R	358
Statistics	040B0h	RJC	Receive Jabber Count	R	358
Statistics	040B4h	MGTPRC	Management Packets Received Count (not applicable to the **82544GC/EI** or **82541ER**)	R	359
Statistics	040B8h	MGTPDC	Management Packets Dropped Count (not applicable to the **82544GC/EI** or **82541ER**)	R	360
Statistics	040BCh	MGTPTC	Management Pkts Transmitted Count (not applicable to the **82544GC/EI** or **82541ER**)	R	360
Statistics	040C0h	TORL	Total Octets Received (Lo)	R	360
Statistics	040C4h	TORH	Total Octets Received (Hi)	R	360
Statistics	040C8h	TOTL	Total Octets Transmitted (Lo)	R	361
Statistics	040CCh	TOTH	Total Octets Transmitted (Hi)	R	361
Statistics	040D0h	TPR	Total Packets Received	R	362
Statistics	040D4h	TPT	Total Packets Transmitted	R	362
Statistics	040D8h	PTC64	Packets Transmitted (64 Bytes) Count	R	363
Statistics	040DCh	PTC127	Packets Transmitted (65-127 Bytes) Count	R	363
Statistics	040E0h	PTC255	Packets Transmitted (128-255 Bytes) Count	R	364
Statistics	040E4h	PTC511	Packets Transmitted (256-511 Bytes) Count	R	364
Statistics	040E8h	PTC1023	Packets Transmitted (512-1023 Bytes) Count	R	365
Statistics	040ECh	PTC1522	Packets Transmitted (1024 Bytes or Greater) Count	R	365
Statistics	040F0h	MPTC	Multicast Packets Transmitted Count	R	366
Statistics	040F4h	BPTC	Broadcast Packets Transmitted Count	R	366
Statistics	040F8h	TSCTC	TCP Segmentation Context Transmitted Count	R	367
Statistics	040FCh	TSCTFC	TCP Segmentation Context Tx Fail Count	R	367
Diagnostic	02410h	RDFH	Receive Data FIFO Head	R/W	368
Diagnostic	02418h	RDFT	Receive Data FIFO Tail	R/W	368
Diagnostic	02420h	RDFHS	Receive Data FIFO Head Saved Register	R/W	369
Diagnostic	02428h	RDFTS	Receive Data FIFO Tail Saved Register	R/W	369
Diagnostic	02430h	RDFPC	Receive Data FIFO Packet Count	R/W	370
Diagnostic	03410h	TDFH	Transmit Data FIFO Head	R/W	370

Category	Offset	Abbreviation	Name	R/W	Page
Diagnostic	03418h	TDFT	Transmit Data FIFO Tail	R/W	371
Diagnostic	03420h	TDFHS	Transmit Data FIFO Head Saved Register	R/W	371
Diagnostic	03428h	TDFTS	Transmit Data FIFO Tail Saved Register	R/W	372
Diagnostic	03430h	TDFPC	Transmit Data FIFO Packet Count	R/W	372
Diagnostic	10000h-1FFFCh	PBM	Packet Buffer Memory (n)	R/W	373

Note: The PHY registers are accessed indirectly through the MDI/O interface described in Section 8.2.

Category	MDI Register	Abbreviation	Name	R/W	Page
PHY	00d	PCTRL	PHY Control Register	R/W	245
PHY	01d	PSTATUS	PHY Status Register	R	248
PHY	02d	PID	PHY Identifier (LSB)	R	250
PHY	03d	EPID	Extended PHY Identifier (MSB)	R	250
PHY	04d	ANA	Auto-Negotiation Advertisement Register	R/W	251
PHY	05d	LPA	Link Partner Ability Register (Base Page)	R	255
PHY	06d	ANE	Auto-Negotiation Expansion Register	R	258
PHY	07d	NPT	Next Page Transmit Register	R/W	259
PHY	08d	LPN	Link Partner Next Page Register	R	260
PHY	09d	GCON	1000BASE-T Control Register	R/W	261
PHY	10d	GSTATUS	1000BASE-T Status Register	R	262
PHY	15d	EPSTATUS	Extended PHY Status Register	R	263
PHY	16d	PSCON	PHY Specific Control Register	R/W	264
PHY	17d	PSSTAT	PHY Specific Status Register	R	267
PHY	18d	PINTE	PHY Interrupt Enable	R/W	270
PHY	19d	PINTS	PHY Interrupt Status	R	272
PHY	20d	EPSCON1 EPSCON (**82544GC/EI**)	Extended PHY Specific Control 1	R/W	275
PHY	21d	PREC	PHY Receive Error Counter	R	277
PHY	23d[1]	PGSTAT	PHY Global Status SPEED_TEN_LED and LINK_ACT_LED[2]	R	279
PHY	24d[1]	PLED	PHY LED Control SPEED_100_LED and SPEED_1000_LED[2]	R/W	280
PHY	26d	EPSCON2	Extended PHY Specific Status (not applicable to the **82540EP/EM** or **82544GC/EI**)	R/W	281
PHY	27d[3]	EPSSTAT	Extended PHY Specific Status	R/W	281
PHY	29d	R30PS	Register 30 Page Select (not applicable to the **82544GC/EI**, **82541xx**, or **82547GI/EI**)	W	281
PHY	30d	R30AW	Register 30 Access Window (not applicable to the **82544GC/EI**, **82541xx**, or **82547GI/EI**)	R/W	282
PHY	31d	PPAGE	Page Select (**82541xx** and **82547GI/EI** only)	R/W	283

1. Applicable to the **82544GC/EI**, **82541xx**, and **82547GI/EI** only.
2. Applicable to the **82541xx**, and **82547GI/EI** only.
3. Applicable to the **82544GC/EI** only.

13.3 PCI-X Register Access Split[1]

The PCI-X specification states that accesses to internal device memory spaces must complete within a specific target initial latency, or else the device should signal that it completes the transaction later using a split-completion operation. Due to internal access latencies, read accesses to most device registers in the Ethernet controller exceeds target initial-access latencies, and therefore are split.

Once a register read operation has been split, the device may, as part of normal operation, initiate a large inbound or outbound transmit or receive data burst transaction. The split completion for the pending register read might be forced to wait until the data burst completes. Therefore, the read access-delay for most registers can be indeterminate (although is generally bounded by the nature or normal burst transactions).

A small subset of the internal register space has been identified as most critical for high-performance driver execution. The variable completion delay for access to some registers could potentially limit the performance of such critical routines as Interrupt Service Routines (ISRs). To help minimize potential critical routine performance, read accesses to a small subset of internal register space will instead complete without being split. These registers are listed as follows:

Category	Offset	Abbreviation	Name
General	00000h	CTRL	Device Control Register
General	00008h	STATUS	Device Status Register
General	00010h	EECD	EEPROM/Flash Control/Data Register
General	00018h	CTRL_EXT	Extended Device Control Register
General	00020h	MDIC	MDI Control Register
General	00028h	FCAL	Flow Control Address Low
General	0002Ch	FCAH	Flow Control Address High
General	00030h	FCT	Flow Control Type
General	00038h	VET	VLAN Ether Type
General	00170h	FCTTV	Flow Control Transmit Timer Value
General	00178h	TXCW	Transmit Configuration Word
General	00180h	RXCW	Receive Configuration Word
General	01000h	PBA	Packet Buffer Allocation
Interrupt	000C0h	ICR	Interrupt Cause Read
Interrupt	000C8h	ICS	Interrupt Cause Set

1. Not applicable to the **82540EP/EM**, **82541xx**, or **82547GI/EI**.

Category	Offset	Abbreviation	Name
Interrupt	000D0h	IMS	Interrupt Mask Set/Read
Interrupt	000D8h	IMC	Interrupt Mask Clear
Transmit	00400h	TCTL	Transmit Control

The EEPROM configuration bit "Force CSR Read Split" (Initialization Control Word 2, word 0Fh) provides the ability to configure the device to split all internal register accesses, rather than providing non-split behavior for the registers listed.

13.4 Main Register Descriptions

This section contains detailed register descriptions for general purpose, DMA, interrupt, receive, and transmit registers. These registers correspond to the main functions of the Ethernet controller.

13.4.1 Device Control Register

CTRL (00000h; R/W)

This register and the Extended Device Control register (CTRL_EXT) control the major operational modes for the Ethernet controller.

While software writes to this register to control device settings, several bits (such as FD and SPEED) can be overridden depending on other bit settings and the resultant link configuration determined by the PHY's Auto-Negotiation resolution.

Note: TBI Mode is used only by the **82544GC/EI** Ethernet controller. Internal SerDes mode is used only by the **82546GB/EB** and **82545GM/EM** Ethernet controllers.

Table 13-3. CTRL Register Bit Description

31	0
Device Control Bits	

Field	Bit(s)	Initial Value	Description
FD	0	1b 0b[1]	Full-Duplex Enables software to override the hardware Auto-Negotiation function. The FD sets the duplex mode only if CTRL.FRCDPLX is set. When cleared, the Ethernet controller operates in half-duplex; when set, the Ethernet controller operates in full-duplex. When the Ethernet controller operates in TBI mode/internal SerDes mode, and the AN Hardware is enabled, this bit is ignored. When the Ethernet controller operates in TBI mode/internal SerDes, and the AN Hardware is disabled, or the link is forced, this bit should be set by software. When the Ethernet controller operates in internal PHY mode, the FD bit is set by software based on AN and data rate resolution. Configurable through the EEPROM.
Reserved	2:1	0b	These bits are reserved and should be set to 00b.
LRST	3	1b	Link Reset (not applicable to the **82540EP/EM**, **82541xx**, or **82547GI/EI**) 0b = Normal; 1b = Link Reset Applicable only in TBI mode/internal SerDes of operation. Used to reset the link control logic and restart the Auto-Negotiation process, when TXCW.ANE is set and TBI mode/internal SerDes is enabled. When set, transmission and reception are halted regardless of TBI mode/internal SerDes setting. A transition to 0b initiates the Auto-Negotiation function. Configurable from the EEPROM, allowing initiation of Auto-Negotiation function at power up.
Reserved	4	0b	Reserved Factory use only. Should be written with 0b.
ASDE	5	0b	Auto-Speed Detection Enable. When set, the Ethernet controller automatically detects the resolved speed of the link by sampling the link in internal PHY mode and self-configures the appropriate status and control bits. Software must also set the SLU bit for this operation. This function is ignored in TBI mode/internal Serdes. The ASD feature provides a method of determining the link speed without the need for software accesses to the MII management registers.

Field	Bit(s)	Initial Value	Description
SLU	6	0b	Set Link Up In TBI mode/internal SerDes, provides manual link configuration. When set, the Link Up signal is forced high once receiver synchronization is achieved (LOS not asserted) using CTRL.FD to determine the duplex mode. This operation bypasses the link configuration process. If Auto-Negotiation is enabled (TXCW.ANE equals 1b), then Set Link Up is ignored. In internal PHY mode, this bit must be set to 1b to permit the Ethernet controller to recognize the I_LOS/I_LIND link signal from the PHY. The "Set Link Up" is normally initialized to 0b. However, if either the *APM Enable* or *SMBus Enable* bits are set in the EEPROM then it is initialized to 1b, ensuring MAC/PHY communication during preboot states (for example, the **82547EI** and **82541EI**). Driver software sets this bit when the driver software initializes, therefore LED indications (link, activity, speed) are not active until the software driver loads even though the PHY has auto-negotiated and established link with a partner on the Ethernet. See Section 8.6 for more information about Auto-Negotiation and link configuration in the various modes. Configurable through the EEPROM.
ILOS	7	0b	Invert Loss-of-Signal (LOS). 0b = do not invert (active high input signal); 1b = invert signal (active low input signal). If using the internal PHY, this bit should be set to 0b to ensure proper communication with the MAC. If using an external TBI device, this bit can be set if the Ethernet controller provides a link loss indication with negative polarity. **Note:** This is a reserved bit for the **82541xx** and **82547GI/EI**.
SPEED	9:8	10b	Speed selection. These bits determine the speed configuration and are written by software after reading the PHY configuration through the MDI/O interface. These signals are ignored in TBI mode/internal Serdes or when Auto-Speed Detection (CTRL.ASDE) is enabled. See Section 8.6 for details. 00b 10 Mb/s 01b 100 Mb/s 10b 1000 Mb/s 11b not used
Reserved	10	0b	Reserved Should be written with 0b to ensure future compatibility.
FRCSPD	11	1b	Force Speed When set, the Ethernet controller speed is configured by CTRL.SPEED bits. The PHY device must resolve to the same speed configuration or software must manually set it to the same speed as the Ethernet controller. When cleared, this allows the PHY device or ASD function (CTRL.ASDE is set) to set the Ethernet controller speed. This bit is superseded by the CTRL_EXT.SPD_BYPS bit, which has a similar function. Applicable only in internal PHY mode of operation and is configurable through EEPROM.

Field	Bit(s)	Initial Value	Description
FRCDPLX	12	0b	Force Duplex When set, software can override the duplex indication from the PHY which is in internal PHY mode. When set the CTRL.FD bit sets duplex. When cleared, the CTRL.FD is ignored.
Reserved	17:13	0b	Reserved Should be written with 0b to ensure future compatibility. Read as 0b
SDP0_DATA	18	0b[2]	SDP0 Data Value. Used to read (write) value of software-controllable IO pin SDP0. If SDP0 is configured as an output (SDP0_IODIR=1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP0 is configured as an input, reads return the current value of the pin.
SDP1_DATA	19	0b[2]	SDP1 Data Value. Used to read (write) value of software-controllable IO pin SDP1. If SDP1 is configured as an output (SDP1_IODIR=1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP1 is configured as an input, reads return the current value of the pin.
ADVD3WUC	20	0b[2]	D3Cold Wakeup Capability Advertisement Enable. When set, D3Cold wakeup capability is advertised based on whether the AUX_PWR pin advertises presence of auxiliary power (yes if AUX_PWR is indicated, no otherwise). When 0b, however, D3Cold wakeup capability is not advertised even if AUX_PWR presence is indicated. Formerly used as SDP2 pin data value, initial value is EEPROM-configurable. **Note:** Not applicable to the **82541ER**.
EN_PHY_ PWR_MGMT	21	0b[2] 1b[1]	PHY Power-Management Enable. When set, the PHY is informed of power-state transitions and attempts to auto-negotiate advertising lower line speeds only (10 or 100 Mb/sec) when entering D3 or D0u power states with wakeup or manageability enabled. It again re-negotiates, advertising full speed capabilities (10/100/1000 Mbps) when transitioning back to full D0 operational state. If this bit is clear, the PHY automatic speed/power management capability is disabled, and the PHY remains operational at its current line speed through power-state transitions. Formerly used as SDP3 pin data value, initial value is EEPROM-configurable.
SDP0_IODIR	22	0b[2]	SDP0 Pin Directionality. Controls whether software-controllable pin SDP0 is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
SDP1_IODIR	23	0b[2]	SDP1 Pin Directionality. Controls whether software-controllable pin SDP1 is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
Reserved	25:24	0b	Reserved. Formerly used as SDP2and SDP3 pin input/output direction control.

Field	Bit(s)	Initial Value	Description
RST	26	0b	Device Reset 0b = normal; 1b = reset. Self clearing. When set, it globally resets the entire Ethernet controller with the exception of the PCI configuration registers. All registers (receive, transmit, interrupt, statistics, etc.), and state machines are set to their power-on reset values. This reset is equivalent to a PCI reset, with the one exception being that the PCI configuration registers are not reset. To ensure that global device reset has fully completed and that the Ethernet controller responds to subsequent access, wait approximately 1 µs after setting and before attempting to check to see if the bit has cleared or to access any other device register.
RFCE	27	0b	Receive Flow Control Enable. When set, indicates that the Ethernet controller responds to the reception of flow control packets. Reception and responding to flow control packets requires matching the content of the Ethernet controller's FCAL/H and FCT registers. If Auto-Negotiation is enabled, this bit is set to the negotiated flow control value. See Section 8.6 for more information about Auto-Negotiation.
TFCE	28	0b	Transmit Flow Control Enable. When set, indicates that the Ethernet controller transmits flow control packets (XON and XOFF frames) based on the receive FIFO fullness, or when triggered to do so based on external control pins (XOFF XON pins when FCTRH.XFCE is set). If Auto-Negotiation is enabled, this bit is set to the negotiated flow control value. See *Auto-Negotiation* for more information.
Reserved	29	0b	Reserved. Should be written with 0b to ensure future compatibility. Read as 0b.
VME	30	0b	VLAN Mode Enable When set to 1b, all packets transmitted from the Ethernet controller that have VLE bit set in their descriptor is sent with an 802.1Q header added to the packet. The contents of the header come from the transmit descriptor and from the VLAN type register. On receive, VLAN information is stripped from 802.1Q packets and is loaded to the packet's descriptor. See Section 9.2 for more details. Reserved. Should be written with 0b to ensure future compatibility. **Note:** Not applicable to the **82541ER**.
PHY_RST	31	0b	PHY Reset 0b = Normal. 1b = Assert hardware reset to the internal PHY. The technique is to set the bit, wait approximately 3 µs, then clear the bit. For the **82547GI/82541GI** (B1 stepping), this register must be used instead of a PHY register. **Note:** For the **82546GB**, when resetting the PHY through the MAC, the PHY should be held in reset for a minimum of 10 ms before releasing the reset signal.

1. **82541xx** and **82547GI/EI** only.
2. If the signature bits of the EEPROM's Initialization Control Word 1 match (01b), these bits are read from the EEPROM.

The ADVD3WUC bit (Advertise D3Cold Wakeup Capability Enable control) allows the AUX_PWR pin to determine whether D3Cold support is advertised. If full 1 Gb/s operation in D3 state is desired but the system's power requirements in this mode would exceed the D3Cold Wakeup-Enabled specification limit (375 mA at 3.3 V dc), this bit can be used to prevent the capability from being advertised to the system.

EEPROM settings allow the default PHY behavior to re-negotiate a lower functional link speed in D3 and D0u states, when PHY operation is still needed for manageability or wakeup capability. The EN_PHY_PWR_MGMT bit allows this capability to be disabled, in case full 1Gb/s speed is desired in these states. The PHY is always powered-down in D3 states when unneeded for either manageability or wakeup support.

Table 13-4. Little-Endian Data Ordering

BEM = 0 (64-bit mode; Little-Endian)

63 ... 0

08	07	06	05	04	03	02	01
10	0f	0e	0d	0c	0b	0a	09

13.4.2 Device Status Register

STATUS (00008h; R)

This register provides software status indication about the Ethernet controller's settings and modes of operation.

Note: TBI Mode is used only by the **82544GC/EI** Ethernet controller. Internal SerDes mode is used only by the **82546GB/EB** and **82545GM/EM** Ethernet controllers.

Table 13-5. Status Register Bit Description

31 … 13	12 … 0
Reserved	Status

Field	Bit(s)	Initial Value	Description
FD	0	X	Link Full Duplex configuration Indication When cleared, the Ethernet controller operates in half-duplex; when set, the Ethernet controller operates in Full duplex. The FD provides the duplex setting status of the Ethernet controller as set by either Hardware Auto-Negotiation function, or by software.
LU	1	X	Link Up Indication 0b = no link config; 1b = link config. For TBI mode/internal SerDes operation: If Auto-Negotiation is enabled, this bit is set if a valid link is negotiated. If link is forced through CTRL.SLU, it reflects the status of this control bit. For internal PHY mode of operation: Reflects the status of the internal link signal indicating a transition to a Link Up. See Section 8.6 for more information about Auto-Negotiation.
Function ID	3:2	0b	Function ID. Provides software a mechanism to determine the Ethernet controller function number (LAN identifier) for this MAC. Read as: [0b,0b] LAN A, [0b,1b] LAN B. **Note:** These settings are only applicable to the **82546GB/EB**. For all other Ethernet controllers, set these bits to 0b.
TXOFF	4	X	Transmission Paused When set, Indicates the transmit function is in Pause state due to reception of an XOFF pause frame when symmetrical flow control is enabled. It is cleared upon expiration of the pause timer, or receipt of an XON frame. Applicable only while working in full-duplex flow-control mode of operation.
TBIMODE	5	X	TBI Mode/internal SerDes Indication When set, the Ethernet controller is configured to work in TBI mode/internal SerDes of operation. When clear, the Ethernet controller is configured to work in internal PHY mode. **Note:** For the **82544GC/EI**, reflects the status of the TBI_MODE input pin. For all other Ethernet controllers, set this bit to 0b.

Field	Bit(s)	Initial Value	Description
SPEED	7:6	X	Link speed setting Indicates the configured speed of the link. These bits are either forced by software when forcing the link speed through the CTRL.SPEED control bits, automatically set by hardware when Auto-Speed Detection is enabled or reflect the internal indication inputs from the PHY. When Auto-Speed Detection is enabled, the Ethernet controller's speed is configured only once after the internal link is asserted. Speed indication is mapped as follows: 00b = 10 Mb/s 01b = 100 Mb/s 10b = 1000 Mb/s 11b = 1000 Mb/s These bits are not valid in TBI mode/internal SerDes.
ASDV	9:8	X	Auto Speed Detection Value Indicates the speed sensed by the Ethernet controller from the internal PHY. The ASDV status bits are provided for diagnostics purposes. The ASD function can be initiated by software writing a logic 1b to the CTRL_EXT.ASDCHK bit. The resultant speed detection is reflected in these bits. See Section 13.4.6 for details.
Reserved	10	X	Reserved Reads as 0b.
PCI66	11	X	PCI Bus speed indication When set, indicates that the PCI bus is running at 66 MHz. Reflects the M66EN input pin. **Note:** Not applicable to the **82547GI/EI**.
BUS64[1]	12	X	PCI Bus Width indication When set, indicates that the Ethernet controller is sitting on a 64-bit PCI/PCI-X bus. BUS64 is determined by REQ64# assertion.
PCIX_MODE[1]	13	X	PCI-X Mode indication When set to 1b, the Ethernet controller is operating in PCI-X Mode; otherwise, the Ethernet controller is operating in conventional PCI Mode.
PCIXSPD[1]	15:14	X	PCI-X Bus Speed Indication Attempts to indicate the speed of the bus when operating in a PCI-X bus. Only valid when STATUS.PCIX_Mode = 1b. 00b = 50-66 MHz 01b = 66-100 MHz 10b = 100-133 MHz 11b = Reserved
Reserved	31:16	0b	Reserved Reads as 0b.

1. Not applicable to the **82540EP/EM**, **82541xx**, or **82547GI/EI**.

13.4.3 EEPROM/Flash Control & Data Register

EECD (00010h; R/W)

This register provides a simplified interface for software accesses to the EEPROM. Software controls the EEPROM by successive writes to this register.

Data and address information is clocked into the EEPROM by software toggling the EECD.SK bit (2) of this register with EECD.CS set to 1b.

Data output from the EEPROM is latched into bit 3 of this register and can be accessed by software through reads of this register.

See Section 5 for more detailed EEPROM information.

Table 13-6. EECD Register Bit Description

31 - 10	9	8	7	6	5 4	3	2	1	0
Reserved	SIZE	PRES	GNT	REQ	FWE	DO	DI	CS	SK

82544GC/EI Only

31 - 6	5 4	3	2	1	0
Reserved	FWE	DO	DI	CS	SK

Field	Bit	Initial Value	Description
SK	0	0b	Clock input to the EEPROM The EESK output signal is mapped to this bit and provides the serial clock input to the EEPROM. Software clocks the EEPROM by means of toggling this bit with successive writes to EECD.
CS	1	0b	Chip select input to the EEPROM The EECS output signal is mapped to the chip select of the EEPROM device. Software enables the EEPROM by writing a 1b to this bit.
DI	2	0b	Data input to the EEPROM The EEDI output signal is mapped directly to this bit. Software provides data input to the EEPROM through writes to this bit.
DO	3	X	Data output bit from the EEPROM The EEDO input signal is mapped directly to this bit in the register and contains the EEPROM data output. This bit is read-only from the software perspective – writes to this bit have no effect.
FWE	5:4	01b	Flash Write Enable Control These two bits, control whether writes to Flash memory are allowed. 00b = Not allowed 01b = Flash writes disabled 10b = Flash writes enabled 11b = Not allowed

Field	Bit	Initial Value	Description
EE_REQ	6[1]	0b	Request EEPROM Access The software must write a 1b to this bit to get direct EEPROM access. It has access when EE_GNT is 1b. When the software completes the access it must write a 0b.
EE_GNT	7[1]	0b	Grant EEPROM Access When this bit is 1b the software can access the EEPROM using the SK, CS, DI, and DO bits.
EE_PRES	8[1]	1b 0b[2]	EEPROM Present This bit indicates that an EEPROM is present by monitoring the EEDO input for a active-low acknowledge by the serial EEPROM during initial EEPROM scan. 1b = EEPROM present.
EE_SIZE	9[1]	0b	EEPROM Size 0b = 1024-bit (64 word) NM93C46 compatible EEPROM 1b = 4096-bit (256 word) NM93C66 compatible EEPROM This bit indicates the EEPROM size, based on acknowledges seen during EEPROM scans of different addresses. This bit is read-only. **Note:** This is a reserved bit for the **82541xx** and **82547GI/EI**.
EE_SIZE	10[1]	0b	EEPROM Size (**82541xx** and **82547GI/EI**) For Microwire EEPROMs: 0b = 6-bit addressable (64 words). 1b = 8-bit addressable (256 words). For SPI EEPROMs: 0b = 8-bit addressable. 1b = 16-bit addressable.
Reserved	12:11	00b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.
EE_TYPE	13[1]	1b	EEPROM Type: Reflects the EE_MODE pin. (**82541xx** and **82547GI/EI**) 0b = Microwire. 1b = SPI.
Reserved	31:14	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

1. Not applicable to the **82544GC/EI**.
2. **82541xx** and **82547GI/EI** only.

This register provides software direct access to the EEPROM. Software can control the EEPROM by successive writes to this register. Data & address information is clocked into the EEPROM by software toggling the EESK bit (2) of this register with EECS set to 1b. Data output from the EEPROM is latched into bit 3 of this register via the internal 62.5 MHz clock and can be accessed by software via reads of this register.

Note: Attempts to write to the FLASH device when writes are disabled (FEW = 01b) should not be attempted. Behavior after such an operation is undefined, and can result in component and/or system hangs.

13.4.4 EEPROM Read Register[1]

EERD (00014h; RW)

Table 13-7. EEPROM Read Register Bit Description

31 16	15 8	7 5	4	3 1	0
Data	Address	RSV.	DONE	RSV.	START

Field	Bit(s)	Initial Value	Description
START	0	0b	Start Read Writing a 1b to this bit causes the EEPROM to read a (16-bit) word at the address stored in the EE_ADDR field and then storing the result in the EE_DATA field. This bit is self-clearing.
Reserved	3:1	0b	Reserved. Reads as 0b.
DONE	4	0b	Read Done Set to 1b when the EEPROM read completes. Set to 0b when the EEPROM read is in progress. Writes by software are ignored.
Reserved	7:5	0b	Reserved. Reads as 0b.
ADDR	15:8	X	Read Address This field is written by software along with *Start Read* to indicate the word to read.
DATA	31:16	X	Read Data. Data returned from the EEPROM read.

1. Not applicable to the **82544GC/EI**.

Table 13-8. EEPROM Read Register Bit Description (82541xx and 82547GI/EI)

31 16	15 2	1	0
Data	Address	DONE	START

Field	Bit(s)	Initial Value	Description
START	0	0b	Start Read Writing a 1b to this bit causes the EEPROM to read a (16-bit) word at the address stored in the EE_ADDR field and then storing the result in the EE_DATA field. This bit is self-clearing.
DONE	1	0b	Read Done Set to 1b when the EEPROM read completes. Set to 0b when the EEPROM read is in progress. Writes by software are ignored.
ADDR	15:2	X	Read Address This field is written by software along with *Start Read* to indicate the word to read.
DATA	31:16	X	Read Data. Data returned from the EEPROM read.

This register is used by software to cause the Ethernet controller to read individual words in the EEPROM. To read a word, software writes the address to the *Read Address* field and simultaneously writes a 1b to the *Start Read* field. The Ethernet controller reads the word from the EEPROM and places it in the *Read Data* field, setting the *Read Done* filed to 1b. Software can poll this register, looking for a 1b in the *Read Done* filed, and then using the value in the *Read Data* field.

When this register is used to read a word from the EEPROM, that word is not written to any of Ethernet controller's internal registers even if it is normally a hardware accessed word.

Note: If software has requested direct pin control of the EEPROM using the EEC register, an access through the EERD register mechanism can stall until the EEC control has been released. Software should ensure that EEC.EE_REQ = 0b and that EEC.EE_GNT = 0b as well before attempting to use EERD to access the EEPROM.

13.4.5 Flash Access[1]

FLA (0001Ch; R/W)

This register provides software direct access to the Flash memory. Software can control the Flash device by successive writes to this register. Data and address information is clocked into the Flash memory by software toggling the FL_SCK bit (0) of this register with FL_CE set to 1b. Data output from the Flash memory is latched into bit three of this register via the internal 125 MHz clock and is accessed by software via reads of this register.

Table 13-9. Flash Access – FLA

31	30	29 - 6	5	4	3	2	1	0
FL ER	FL BS	Reserved	FL GNT	FL REQ	FL SO	FL SI	FL CS	FL SCK

Field	Bit(s)	Initial Value	Description
FL_SCK	0	0b	Clock Input to the FLASH. When FL_GNT is 1b, the FL_SCK out signal is mapped to this bit and provides the serial clock input to the Flash device. Software clocks the Flash memory via toggling this bit with successive writes.
FL_CE	1	0b	Chip FL Input to the FLASH. When FL_GNT is 1b, the FL_CS output signal is mapped to the chip select of the device. Software enables the FLASH by writing a 0b to this bit.
FL_SI	2	0b	Data Input to the FLASH. When FL_GNT is 1b, the FL_DI output signal is mapped directly to this bit. Software provides data input to the FLASH via writes to this bit.
FL_SO	3	X	Data Output Bit from the FLASH. The FL_SO input signal is mapped directly to this bit in the register and contains the Flash memory serial data output. This bit is read only from the software perspective — writes to this bit have no effect.
FL_REQ	4	0b	Request FLASH Access. The software must write a 1b to this bit to get direct Flash memory access. It has access when FL_GNT is 1b. When the software completes the access it must write a 0b.
FL_GNT	5	0	Grant FLASH Access. When this bit is 1b, the software can access the Flash memory using the FL_SCK, FL_CE, FL_SI, and FL_SO bits.
Reserved	29:6	0	Reserved. Reads as 0b.
FL_BUSY	30	0	Flash Busy. This bit is set to 1b while a write or an erase to the Flash memory is in progress. While this bit is clear (read as 0b) software can access to write a new byte to the Flash device.
FL_ER	31	0	Flash Erase Command. This command will be sent to the Flash component only if bits 5:4 are also set. This bit is automatically cleared and read as 0b.

1. Applicable to the **82541xx** and **82547GI/EI** only.

13.4.6 Extended Device Control Register

CTRL_EXT (00018h, R/W)

This register and the Device Control register (CTRL) controls the major operational modes for the Ethernet controller. CTRL_EXT provides extended control of the Ethernet controller functionality over the Device Control register (CTRL).

Note: See Table 13-12 and Table 13-13 for the **82544GC/EI**.

Table 13-10. CTRL_EXT Register Bit Description

31 - 24	23 - 16 15 0
Reserved	Extended Device Control Bits

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0b	Reserved bits for the **82541xx** and **82547GI/EI**. Should be written as 0b to ensure future compatibility.
GPI_EN	3:0	0b	General Purpose Interrupt Enables These bits determine whether the upper three software definable pins SDP[7:6] and SDP[4] are mapped to the ICR.GPI interrupt bits. These mappings are enabled only when the SDP[7:6] and SDP[4] pins are configured as inputs through CTRL_EXT.SWDPIOHI. Refer to Table 13-11 for SDP to ICR.GPI bit mapping.
GPI_EN	3:2	0b	General Purpose Interrupt Enables for the **82541xx** and **82547GI/EI**. These bits determine whether the upper software definable pins SDP[3:2] are mapped to the ICR.GPI interrupt bits. These mappings are enabled only when the SDP[3:2] pins are configured as inputs through CTRL_EXT.SWDPIOHI. Refer to Table 13-11 for SDP to ICR.GPI bit mapping.
Reserved	4	0b	Reserved. Formally used as SDP4 pin data value. Reads as 0b.
PHYINT	5	0b	PHY Interrupt Value. When read, returns the current value of the PHY internal interrupt status PHYINT. **Note:** This is a reserved bit for the **82541xx** and **82547GI/EI**.
SDP6_DATA SDP2_DATA (**82541xx** and **82547GI/EI**)	6	0b[1]	SDP6[2] Data Value. Used to read (write) value of software-controllable IO pin SDP6[2]. If SDP6[2] is configured as an output (SDP6[2]_IODIR = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP6[2] is configured as an input, reads return the current value of the pin.
SDP7_DATA SDP3_DATA (**82541xx** and **82547GI/EI**)	7	0b[1]	SDP7[3] Data Value. Used to read (write) value of software-controllable IO pin SDP7[3]. If SDP7[3] is configured as an output (SDP7[3]_IODIR = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP7[3] is configured as an input, reads return the current value of the pin.
Reserved	9:8	01b	Reserved Should be written as 01b to ensure future compatibility.

Field	Bit(s)	Initial Value	Description
SDP6_IODIR SDP2_IODIR (**82541xx** and **82547GI/EI**)	10	0b[1]	SDP6[2] Pin Directionality. Controls whether software-controllable pin SDP6[2] is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
SDP7_IODIR SDP2_IODIR (**82541xx** and **82547GI/EI**)	11	0b[1]	SDP7[2] Pin Directionality. Controls whether software-controllable pin SDP7[2] is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
ASDCHK	12	0b	ASD Check Initiate an Auto-Speed-Detection (ASD) sequence to sense the frequency of the PHY receive clock. The results are reflected in STATUS.ASDV. This bit is self-clearing. This functionality is provided for diagnostic purposes, regardless of whether the Auto Speed Detection feature is enabled. This bit is applicable only for internal PHY mode of operation.
EE_RST	13	0b	EEPROM Reset When set, initiates a reset-like event to the EEPROM function. This causes the EEPROM to be read as if a RST# assertion had occurred. All device functions should be disabled prior to setting this bit. This bit is self-clearing.
Reserved	14	0b[1]	Reserved. Should be set to 0b.
SPD_BYPS	15	0b	Speed Select Bypass When set to 1b, all speed detection mechanisms are bypassed, and the Ethernet controller is immediately set to the speed indicated by CTRL.SPEED. This can be used to override the hardware clock switching circuitry and give full control to software. SPD_BYPS differs from the CTRL.FRCSPD function in that FRCSPD uses the internal clock switching circuitry rather than an immediate forcing function of the speed settings, as does SPD_BYPS.
Reserved	16	0b[1]	Reserved. Should be set to 0b.
RO-DIS	17	0b	Relaxed Ordering Disabled When set to 1b, the Ethernet controller does not request any relaxed ordering transactions in PCI-X mode regardless of the state of bit 1 in the PCI-X command register. When this bit is clear and bit 1 of the PCI-X command register is set, the Ethernet controller requests relaxed ordering transactions. **Note:** This is a reserved bit for the **82540EP/EM**, **82541xx**, and **82547GI/EI**. Set to 0b.
Reserved	20:18	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

Field	Bit(s)	Initial Value	Description
VREG POWER DOWN	21	0b	Voltage Regulator Power Down (**82541xx** and **82547GI/EI** only) 0b = Normal operation. 1b = Voltage regulators power down. This bit is initialized from the EEPROM. **Note:** This is a reserved bit for all remaining Ethernet controllers. Set to 0b.
LINK_MODE	23:22	0b	Link Mode. This controls which interface is used to talk to the link. 00b = Direct copper (1000Base-T) interface (GMII/MII internal PHY mode) 01b = Reserved 10b = Direct Fiber interface (using internal SerDes) 11b = external TBI interface **Note:** These are reserved bits for the **82540EP/EM**, **82541xx**, and **82547GI/EI**. Set to 00b
Reserved	31:24	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

1. These bits are read from the EEPROM

The Ethernet controller allows for up to two externally controlled interrupts. The upper two software-definable pins, SDP[7:6] (SDP[3:2] for the **82541xx** and **82547GI/EI**), can be mapped for use as GPI interrupt bits. These mappings are enabled by the SDPx_GPIEN bits only when these signals are also configured as inputs via SDPx_IODIR. When configured to function as external interrupt pins, a GPI interrupt is generated when the corresponding pin is sampled in an active-high state.

The bit mappings are shown in Table 13-11 for clarity.

Table 13-11. GPI to SDP Bit Mappings

SDP pin to be used as GPI	CTRL_EXT field settings		Resulting ICR bit (GPI)
	Directionality	Enable as GPI interrupt	
7 2[1]	SDP7_IODIR SDP2_IODIR[1]	SDP7_GPIEN SDP2_GPIEN[1]	14
6 3[1]	SDP6_IODIR SDP3_IODIR[1]	SDP6_GPIEN SDP3_GPIEN[1]	13

1. **82541xx** and **82547GI/EI** only

Table 13-12. 82544GC/EI CTRL_EXT Register Bit Description

31 16	15 0
Reserved	Extended Device Control Bits

Field	Bit(s)	Initial Value	Description
GPI_EN	3:0	0	General Purpose Interrupt Enables These bits determine whether the upper three software definable pins SDP[7:6] and SDP[4] are mapped to the ICR.GPI interrupt bits. These mappings are enabled only when the SDP[7:6] and SDP[4] pins are configured as inputs through CTRL_EXT.SWDPIOHI. Refer to Table 13-13 for SDP to ICR.GPI bit mapping.
SWDPINSHI	7:6, 4	0	Software Defined Pins – high nibble These three bits allow direct control of SDP[7:6] and SDP[4]. These pins can be either input pins or output pins as determined by the SWDPIOHI bits. The initial direction of the software defined pins is read out of the EEPROM. **Note:** SDP[5] and its associated bit is not used. It should be programmed to 0b for future compatibility.
SWDPIOHI	11:10, 8	0	Software Defined Pins Input or Output These three bits control whether each of the high nibble software defined pins SDP[7:6] and SDP[4] is used as an input or an output. 0b = inputs; 1b = outputs. This field is not affected by assertion of software reset (CTRL.RST). Configurable through EEPROM. **Note:** SDP[5] and its associated bit is not used. It should be programmed to 0b for future compatibility.
ASDCHK	12	0	ASD Check Initiate an Auto-Speed-Detection (ASD) sequence to sense the frequency of the PHY receive clock. The results are reflected in STATUS.ASDV. This bit is self-clearing. This functionality is provided for diagnostic purposes, regardless of whether the Auto Speed Detection feature is enabled. This bit is applicable only for internal PHY mode of operation.
EE_RST	13	0	EEPROM Reset When set, initiates a “reset-like” event to the EEPROM function. This causes the EEPROM to be read as if a RST# assertion had occurred. All Ethernet controller functions should be disabled prior to setting this bit. This bit is self-clearing.
IPS	14	0	Invert Power State Bit 0 When set to 1b, inverts the assertion polarity of the PWR_STATE bit 0 output. When cleared to 0b, PWR_STATE is logic high in normal operation. Configurable through EEPROM.

Field	Bit(s)	Initial Value	Description
SPD_BYPS	15	0	Speed Select Bypass When set to 1b, all speed detection mechanisms are bypassed, and the Ethernet controller is immediately set to the speed indicated by CTRL.SPEED. This might be used to override the hardware clock switching circuitry and give full control to software. SPD_BYPS differs from the CTRL.FRCSPD function in that FRCSPD uses the internal clock switching circuitry rather than an immediate forcing function of the speed settings, as does SPD_BYPS.
IPS1	16	0	Invert Power State Bit 1 Inverts the polarity of bit 1 of the PWR_STATE signal when set to 1b. Configurable through the EEPROM.
RO-DIS	17	0	Relaxed Ordering Disabled When set to 1b, the Ethernet controller does not request any relaxed ordering transactions in PCI-X mode regardless of the state of bit 1 in the PCI-X command register. When this bit is clear and bit 1 of the PCI-X command register is set, the Ethernet controller requests relaxed ordering transactions as described.
Reserved	31:18	0	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

Table 13-13. 82544GC/EI GPI to SDP Bit Mapping

B_SDP	CTRL_EXT (SWDPINHI)	CTRL_EXT (GPI_EN)	ICR (GPI)
7	7	3	14
6	6	2	13
5	Reserved	Reserved	Reserved
4	4	0	11

13.4.7 MDI Control Register

MDIC (00020h; R/W)

Software uses this register to read or write Management Data Interface (MDI) registers in the internal PHY.

To read a location in the PHY, first perform an MDI write cycle with the following bit settings:

- Ready = 0b
- Interrupt Enable programmed to 1b or 0b
- Opcode = 10b (read)
- PHYADD = PHY address from the MDI register
- REGADD = Register address within the PHY to be read

When the serial data transfer from PHY to MAC is complete, the Ethernet controller issues an interrupt if the MDI interrupt is enabled. The Ethernet controller also sets the Ready bit. This indication tells the system that the read data is available from the 16-bit data field in the MDI Control register. Perform a second read operation to the register at this time to recover the data.

To write a location in the PHY, perform an MDI write cycle with the following bit settings:

- Ready = 0b
- Interrupt Enable programmed to 1b or 0b
- Opcode = 01b (write)
- PHYADD = PHY address from the MDI register
- REGADD = Register address within the PHY to be written
- Data = Specific data for PHY operation

If enabled, the Ethernet controller issues an interrupt when the write completes. The Ethernet controller also sets the Ready bit, denoting that a subsequent operation can be carried out.

Note: The internal PHY register bit descriptions follow Table 13-14.

Table 13-14. MDI Control Register Bit Description

31	30	29	28	27 26	25 21	20 16	15 0
RSV	E	I	R	OP	PHY	REG	DATA

Field	Bit(s)	Initial Value	Description
DATA	15:0	X	Data In a Write command, software places the data bits and the Ethernet controller shifts them out to the PHY. In a Read command, the Ethernet controller reads these bits serially from the PHY and software can read them from this location.
REGADD	20:16	0b	PHY Register Address: Reg. 0, 1, 2, ...31
PHYADD	25:21	0b	PHY Address The Internal PHY's MDI address for each MAC is 0001b
OP	27:26	0b	Opcode 01b = MDI Write 10b = MDI Read All other values are reserved.
R	28	0b	Ready Bit Set to 1b by the Ethernet controller at the end of the MDI transaction (for example, indication of a Read or Write completion). It should be reset to 0b by software at the same time the command is written.
I	29	0b	Interrupt Enable When set to 1b by software, it causes an Interrupt to be asserted to indicate the end of an MDI cycle.
E	30	0b	Error This bit is set to 1b by hardware when it fails to complete an MDI read. Software should make sure this bit is clear (0b) before issuing an MDI read or write command.
Reserved	31	0b	Reserved Reads as 0b.

13.4.7.1 PHY Registers

This document uses a special nomenclature to define the read/write mode of individual bits in each register. See Table 13-15.

For all binary equations appearing in the register map, the symbol “|” is equivalent to a binary OR operation.

Table 13-15. PHY Register Bit Mode Definitions

Register Mode	Description
LH	Latched High. Event is latched and erased when read.
LL	Latched Low. Event is latched and erased when read. For example, Link Loss is latched when the PHY Control Register bit 2 = 0b. After read, if the link is good, the PHY Control Register bit 2 is set to 1b.
RO	Read Only.
R/W	Read and Write.
SC	Self-Clear. The bit is set, automatically executed, and then reset to normal operation.
CR	Clear after Read. For example, 1000BASE-T Status Register bits 7:0 (Idle Error Counter).
Update	Value written to the register bit does not take effect until software PHY reset is executed.

13.4.7.1.1 PHY Control Register PCTRL (00d; R/W)

Table 13-16. PHY Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	5:0	These bits are reserved and should be set to 000000b.	RO RW[1]	Always 000000b	
Speed Selection (MSB)	6	Speed Selection is determined by bits 6 (MSB) and 13 (LSB) as follows. 11b = Reserved 10b = 1000 Mbps 01b = 100 Mbps 00b = 10 Mbps A write to these bits do not take effect until a software reset is asserted, Restart Auto-Negotiation is asserted, or Power Down transitions from power down to normal operation. **82544GC/EI** only: The Speed Selection bits take on the values set by external pins ANEG[3:0] on hardware reset only. Bit 6: ANEG[3] | ANEG[2] | (MODE[3:0] is one of xx01b, 1x00b, 001xb, 0111b). Bit 13: (ANEG[3:1] = 001b) and (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b).	R/W	1b ANEG[3:2] MODE[3:0]	Update
Collision Test	7	1b = Enable COL signal test. 0b = Disable COL signal test.	R/W	0b	0b
Duplex Mode	8	1b = Full Duplex. 0b = Half Duplex. **82544GC/EI** only: The Duplex bit takes on the value set by external pins ANEG[3:2, 0] on hardware reset only. Bit 8: ANEG[3:2,0] = 001 | ANEG[3]. A write to this bit does not take effect until a software reset is asserted, Restart Auto-Negotiation is asserted, or Power Down transitions from power down to normal power.	R/W	1b ANEG[3:2, 0]	Update
Restart Auto-Negotiation	9	1b = Restart Auto-Negotiation Process. 0b = Normal operation. Auto-Negotiation automatically restarts after hardware or software reset regardless of whether or not the restart bit is set.	R/W, SC	0b	Self Clear
Isolate	10	1b = Isolate. 0b = Normal operation.	R/W	0b	0b

Table 13-16. PHY Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Power Down	11	1b = Power down. 0b = Normal operation. Power down shuts down the Ethernet controller except for the MAC interface if the MAC interface power down bit is set to 1b. If it equals 0b, then the MAC interface also shuts down. For the **82544GC/EI**, power down has no effect on the 125CLK output if the Disable 125CLK bit is set to 0b. **NOTE:** Setting this bit to 1b will prevent wakeup by detecting circuitry on the CAT5 cable. To enable wakeup, this bit must be written back to 0b. **82544GC/EI** only: If bit 12 is set to 0b and speed is manually forced to 1000 Mb/s in bits 13 and 6, then Auto-Negotiation is still enabled and only 1000BASE-T full duplex is advertised if bit 8 is set to 1b. 1000BASE-T half duplex is advertised if bit 8 is cleared (0b). Duplex settings in other registers are ignored. Auto-Negotiation is required by IEEE for proper operation in 1000BASE-T.	R/W	0b	0b

Table 13-16. PHY Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Auto-Negotiation Enable	12	1b = Enable Auto-Negotiation Process. 0b = Disable Auto-Negotiation Process. A write to this bit does not take effect until a software reset is asserted, Restart Auto-Negotiation is asserted, or Power Down transitions from power down to normal operation. When the port is switched from power down to normal operation, software reset and restart Auto-Negotiation are performed even if bits Reset and Restart Auto-Negotiation are not set by the programmer. If bit 12 is set to 0b and speed is manually forced to 1000 Mbps in bits 13 and 6, then Auto-Negotiation is still enabled and only 1000BASE-T full duplex is advertised if bit 8 is set to 1b. 1000BASE-T half duplex is advertised if bit 8 is cleared (0b). Duplex settings in other registers are ignored. Auto-Negotiation is required by IEEE for proper operation in 1000BASE-T. **82544GC/EI** only: Auto-Negotiation enable takes on the value set by external pins ANEG[3:0] on hardware reset only. Bit 12: ANEG[3:2] = 11b. If MODE[3:0] equals 001xb or 0111b, where x equals either 0b or 1b, then the ANE bit determines whether 1000BASE-X Auto-Negotiation is on or off. Otherwise ANE determines whether 10/100/1000BASE-T Auto-Negotiation is on or off.	R/W R/W	1b ANEG[3:2] ENA_XC	Update Update
Speed Selection (LSB)	13	See Speed Selection (MSB), bit 6.	R/W	0b ANEG[3:1][2] MODE[3:0]	Update
Loopback	14	1b = Enable loopback. 0b = Disable loopback.	R/W	0b	0b
Reset	15	1b = PHY reset. 0b = Normal operation.	R/W, SC	0b	Self Clear

1. **82541xx** and **82547GI/EI** only.
2. **82544GC/EI** only.

13.4.7.1.2 PHY Status Register PSTATUS (01d; R)

Table 13-17. PHY Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Extended Capability	0	1b = Extended register capabilities.	RO	Always 1b	
Jabber Detect	1	1b = Jabber condition detected. 0b = Jabber condition not detected.	RO,LH	0b	0b
Link Status	2	1b = Link is up. 0b = Link is down. This register indicates whether link was lost after the last read. For the current link status, either read this register back-to-back or read the Link Real Time bit 17 in the PHY Specific Status Register.	RO, LL	0b	0b
Auto-Negotiation Ability	3	1b = PHY able to perform Auto-Negotiation.	RO	Always 1b	
Remote Fault	4	1b = Remote fault condition detected. 0b = Remote fault condition not detected.	RO,LH	0b	0b
Auto-Negotiation Complete	5	1b = Auto-Negotiation process complete. 0b = Auto-Negotiation process not complete.	RO	0b	0b
MF Preamble Suppression	6	1b = PHY accepts management frames with preamble suppressed.	RO	Always 1b (0b for the **82541xx** and **82547GI/EI**)	
Reserved	7	Reserved. Should be set to 0b	RO	Always 0b	
Extended Status	8	1b = Extended status information in the Extended PHY Status Register (15d).	RO	Always 1b	
100BASE-T2 Half Duplex	9	0b = PHY not able to perform half duplex 100BASE-T2.	RO	Always 0b	
100BASE-T2 Full Duplex	10	0b = PHY not able to perform full duplex 100BASE-T2.	RO	Always 0b	
10 Mb/s Half Duplex	11	1b = PHY able to perform half duplex 10BASE-T. 0b = PHY not able to perform half duplex 10BASE-T. **82544GC/EI** only: Bit 14 = Bit 13 = Bit 12 = Bit 11 = (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b).	RO RO	1b MODE[3:0]	

Table 13-17. PHY Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
10 Mb/s Full Duplex	12	1b = PHY able to perform full duplex 10BASE-T. 0b = PHY not able to perform full duplex 10BASE-T. **82544GC/EI** only: Bit 14 = Bit 13 = Bit 12 = Bit 11 = (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b).	RO RO	1b MODE[3:0]	
100BASE-X Half Duplex	13	1b = PHY able to perform half duplex 100BASE-X. 0b = PHY able to perform half duplex 100BASE-X. **82544GC/EI** only: Bit 14 = Bit 13 = Bit 12 = Bit 11 = (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b).	RO RO	1b MODE[3:0]	
100BASE-X Full Duplex	14	1b = PHY able to perform full duplex 100BASE-X. 0b = PHY not able to perform full duplex 100BASE-X. **82544GC/EI** only: Bit 14 = Bit 13 = Bit 12 = Bit 11 = (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b).	RO RO	1b MODE[3:0]	
100BASE-T4	15	0b = PHY not able to perform 100BASE-T4.	RO	Always 0b	

13.4.7.1.3 PHY Identifier Register (LSB) PID (02d; R)

Table 13-18. PHY Identifier Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Organizationally Unique Identifier Bit 18:3[1]	15:0	0000_0001_0100_0001b OUI is 005043h. The PHY identifier composed of bits 3 through 18 of the OUI (Organizationally Unique Identifier)[2]	RO	Always 0141h Always 02A8h[2]	

1. PHY ID number for the **82541xx** and **82547GI/EI** only.
2. **82541xx** and **82547GI/EI** only.

13.4.7.1.4 Extended PHY Identifier Register (MSB) EPID (03d; R)

Table 13-19. Extended PHY Identifier Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Revision Number	3:0	Contains the current revision number. 4 bits containing the manufacturer's revision number.[1]	RO	Always static 00h[1]	
Model Number	9:4	000010b = 10/100/1000 Copper PHY. **82544GC/EI** only: 000011b = 10/100/1000 Copper PHY. 6 bits containing the manufacturer's part number[1]	RO	Always 000010b **82544GC/EI** only: Always 000011b Always 111000b[1]	
Organizationally Unique Identifier Bit 19:24 (PHY ID Number for the **82541xx** and **82547GI/EI**)	15:10	000011b. The PHY identifier composed of bits 19 through 24 of the OUI[1]	RO	Always 000011b 0h[1]	

1. **82541xx** and **82547GI/EI** only.

13.4.7.1.5 Auto-Negotiation Advertisement Register ANA (04d; R/W)

Table 13-20. Auto-Negotiation Advertisement Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Selector Field	4:0	00001b = 802.3 For the **82541xx** and **82547GI/EI**: Other combinations are reserved. Unspecified or reserved combinations should not be transmitted. **Note:** Setting this field to a value other than 00001b can cause auto negotiation to fail.	RO	Always 00001b	
10BASE-TX Half Duplex 10Base-T (**82541xx** and **82547GI/EI**)	5	1b = Advertise. 0b = Not advertised. Values programmed in the Auto-Negotiation Advertisement Register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down. This bit can be overridden by the PHY Control Register. **82544GC/EI** only: Bit 5: (ANEG[3:0] = 0000b | ANEG[3:2] = 11b) and (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b). **82541xx and 82547GI/EI** only: 1b = DTE is 10BASE-T capable. 0b = DTE is not 10BASE-T capable.	R/W	1b ANEG[3:1] MODE[3:0] 1b	Retain
10BASE-TX Full Duplex 10Base-T Full Duplex (**82541xx** and **82547GI/EI**)	6	1b = Advertise. 0b = Not advertised. Values programmed in the Auto-Negotiation advertisement register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down. This bit can be overridden by the PHY Control Register. **82544GC/EI** only: Bit 5: (ANEG[3:0] = 0001b | ANEG[3:2] = 11b) and (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b). **82541xx and 82547GI/EI** only: 1b = DTE is 10BASE-T full duplex capable. 0b = DTE is not 10BASE-T full duplex capable.	R/W	1b ANEG[3:0] MODE[3:0] 1b	Retain

Table 13-20. Auto-Negotiation Advertisement Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
100BASE-TX Half Duplex 100Base-TX (**82541xx** and **82547GI/EI**)	7	1b = Advertise. 0b = Not advertised. Values programmed in the Auto-Negotiation advertisement register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down. This bit can be overridden by the PHY Control Register. **82544GC/EI** only: Bit 5: (ANEG[3:0] = 0010b | ANEG[3:2] = 11b) and (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b). **82541xx and 82547GI/EI** only[1]: 1b = DTE is 100BASE-TX capable. 0b = DTE is not 100BASE-TX capable.	R/W	1b ANEG[3:1] MODE[3:0] 1b	Retain
100BASE-TX Full Duplex	8	1b = Advertise. 0b = Not advertised. Values programmed in the Auto-Negotiation advertisement register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down. This bit can be overridden by the PHY Control Register. **82544GC/EI** only: Bit 8: (ANEG[3:0] = 0011b | ANEG[3:2] = 11b) and (MODE[3:0] is not any of xx01b, 1x00b, 001xb, 0111b). **82541xx and 82547GI/EI** only:[1] 1b = DTE is 100BASE-TX full duplex capable. 0b = DTE is not 100BASE-TX full duplex capable.	R/W	1b ANEG[3:0] MODE[3:0] 1b	Retain
100BASE-T4	9	0b = Not capable of 100BASE-T4.	RO	Always 0b	
PAUSE	10	1b = MAC PAUSE implemented. 0b = MAC PAUSE not implemented. Values programmed in the Auto-Negotiation advertisement register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down. **82541xx and 82547GI/EI** only: Advertise to Partner that Pause operation (as defined in 802.3x) is desired.	R/W	0b 1b for the (**82541xx** and **82547GI/EI**)	Retain

Table 13-20. Auto-Negotiation Advertisement Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Asymmetric Pause ASM_DIR for the (**82541xx** and **82547GI/EI**)	11	1b = Asymmetric Pause. 0b = No asymmetric Pause. Values programmed in the Auto-Negotiation advertisement register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down. **82541xx and 82547GI/EI** only: Advertise Asymmetric Pause direction bit. This bit is used in conjunction with PAUSE.	R/W	0b 1b for the (**82541xx** and **82547GI/EI**)	Retain
Reserved	12	This bit is reserved and equals 0b. Values programmed in the Auto-Negotiation advertisement register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down. Reserved bit is R/W to allow for forward compatibility with future IEEE standards.	R/W	0b	Retain
Remote Fault	13	1b = Set Remote Fault bit. 0b = Do not set Remote Fault bit. Values programmed in the Auto-Negotiation advertisement register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down.	R/W	0b	Retain
Reserved	14	Reserved. Should be set to 0b.	RO	Always 0b	
Next Page	15	1b = Advertise. 0b = Not advertised. Values programmed in the Auto-Negotiation advertisement register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or link goes down. If 1000BASE-T is advertised then the required next pages are automatically transmitted. The Next Page bit should equal 0 if no additional next pages are needed. **82541xx and 82547GI/EI** only: 1b = Manual control of Next Page (Software). 0b = Ethernet controller control of Next Page (Auto).	R/W	0b	Retain

1. For the 82541xx and 82547GI/EI, if EEPROM bit ADV10LU is asserted, then the default is set to 0b. Otherwise, the default is 1b.

82544GC/EI Only:

Table 13-21. Auto-Negotiation Advertisement Register Bit Description (MODE[3:0] is one of 001xb, 0111b)

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	4:0	Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down. Reserved bit is R/W to allow forward compatibility with future IEEE standards.	R/W	00000b	Retain
1000BASE-X Full Duplex	5	1b = Advertise 0b = Not advertised Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down. Bit 5: ANEG[3] = 1b	R/W	ANEG[3]	Retain
1000BASE-X Half Duplex	6	1b = Advertise 0b= Not advertised Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down. Bit 6: ANEG[2] = 1b	R/W	ANEG[2]	Retain
Pause	7	Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down.	R/W	0b	Retain
Asymmetric Pause	8	Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down.	R/W	000b	Retain
Reserved	11:9	Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down. Reserved bit is R/W to allow forward compatibility with future IEEE standards.	R/W	0b	Retain
RF1	12	Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down. Ethernet controller has no ability to detect remote fault.	R/W	0b	Retain
RF2	13	Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down. Ethernet controller has no ability to detect remote fault.	R/W	0b	Retain
Reserved	14	Reserved. Should be set to 0b.	RO	Always 0b	

Table 13-21. Auto-Negotiation Advertisement Register Bit Description (MODE[3:0] is one of 001xb, 0111b)

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Next Page	15	0b = Not advertised Values programmed in this register have no effect unless Auto-Negotiation is restarted (PHY Control Register) or the link goes down. Next Page is not supported in 1000BASE-X mode.	RO	Always 0b	

13.4.7.1.6 Link Partner Ability Register (Base Page) LPA (05d; R)

Table 13-22. Link Partner Ability Register (Base Page) Bit Description[1]

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Selector Field	4:0	Received Code Word Bit 4:0.	RO	00000b	00000b
Technology Ability Field	12:5	Received Code Word Bit 12:5.	RO	00h	00h
Remote Fault	13	Received Code Word Bit 13.	RO	0b	0b
Acknowledge	14	Received Code Word Bit 14.	RO	0b	0b
Next Page	15	Received Code Word Bit 15.	RO	0b	0b

1. (MODE[3:0] is one of xx00b, 1x01b, 101xb, 1111b) for the **82544GC/EI** only.

82544GC/EI Only:

Table 13-23. Link Partner Ability Register (Base Page) Bit Description[1]

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	4:0	Reserved. Should be set to 00000b.	RO	00000b	00000b
10BASE-TX Half Duplex	5	1b = 10 Base-TX half duplex is available. 0b = 10 Base-TX half duplex is not available.	RO	0b	0b
10BASE-TX Full Duplex	6	1b = 10 Base-TX full duplex is available. 0b = 10 Base-TX full duplex is not available.	RO	0b	0b
100BASE-TX Half Duplex	7	1b = 100 Base-TX half duplex is available. 0b = 100 Base-TX half duplex is not available.	RO	0b	0b
100BASE-TX Full Duplex	8	1b = 100 Base-TX full duplex is available. 0b = 100 Base-TX full duplex is not available.	RO	0b	0b
100BASE-T4	9	0b = Not capable of 100BASE-T4.	RO	Always 0b	
Pause	10	1b = Pause operation is available. 0b = Pause operation is not available.	RO	0b	0b
Asymmetric Pause	11	1b = Asymmetric Pause operation is available. 0b = Asymmetric Pause operation is not available.	RO	0b	0b
Reserved	12	Reserved. Should be set to 0b.	RO	0b	0b
Remote Fault	13	Indicates a remote fault.	RO	0b	0b
Reserved	14	Reserved. Should be set to 0b.	RO	0b	0b
Next Page	15	1b = Link partner is Next Pagable. 0b = Link partner is not Next Pagable.	RO	0b	0b

1. (MODE[3:0] is one of 001xb, 0111b).

82541xx and 82547GI/EI Only:

Table 13-24. PHY Link Page Ability Bit Description[1]

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Selector Field [4:0]	4:0	<00001> = IEEE 802.3 Other combinations are reserved. Unspecified or reserved combinations shall not be transmitted. If field does not match PHY Register 4, bits 4:0, the AN process does not complete and no HCD is selected.	RO	N/A	
10BASE-T	5	1b = Link Partner is 10BASE-T capable. 0b = Link Partner is not 10BASE-T capable.	RO	N/A	
10BASE-T Full Duplex	6	1b = Link Partner is 10BASE-T full duplex capable. 0b = Link Partner is not 10BASE-T full duplex capable.	RO	N/A	
100BASE-TX	7	1b = Link Partner is 100BASE-TX capable. 0b = Link Partner is not 100BASE-TX capable.	RO	N/A	
100BASE-TX Full Duplex	8	1b = Link Partner is 100BASE-TX full duplex capable. 0b = Link Partner is not 100BASE-TX full duplex capable.	RO	N/A	
100BASE-T4	9	1b = Link Partner is 100BASE-T4 capable. 0b = Link Partner is not 100BASE-T4 capable.	RO	N/A	
LP Pause	10	Link Partner uses Pause Operation as defined in 802.3x.	RO	N/A	
LP ASM_DIR	11	Asymmetric Pause Direction Bit 1b = Link Partner is capable of asymmetric pause. 0b = Link Partner is not capable of asymmetric pause.	RO	N/A	
Reserved	12	Reserved. Should be set to 0b.	RO	0b	
Remote Fault	13	1b = Remote fault. 0b = No remote fault.	RO	N/A	
Acknowledge	14	1b = Link Partner has received Link Code Word from the PHY. 0b = Link Partner has not received Link Code Word from the PHY.	RO	N/A	
Next Page	15	1b = Link Partner has ability to send multiple pages. 0b = Link Partner has no ability to send multiple pages.	RO	N/A	

1. PHY register 8d stores the Auto-Negotiation Link Partner Received Next Pages. PHY register 5d is not used to store Next Pages. It contains the information from the last Base Page correctly received.

13.4.7.1.7 Auto-Negotiation Expansion Register ANE (06d; R)

Table 13-25. Auto-Negotiation Expansion Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Link Partner Auto-Negotiation Able	0	1b = Link Partner is Auto-Negotiation able. 0b = Link Partner is not Auto-Negotiation able.	RO	0b	0b
Page Received	1	1b = A New Page has been received. 0b = A New Page has not been received. **82541xx** and **82547GI/EI** only: If PHY register 16, bit 1 (Alternate NP Feature) is set, the Page Received bit also clears when mr_page_rx = false or transmit_disable = true.	RO/ LH[1]	0b	0b
Local Next Page Able	2	1b = Local Device is Next Pageable.	RO	0b 1b (**82541xx** and **82547GI/EI**)	
Link Partner Next Page Able	3	1b = Link Partner is Next Page able. 0b = Link Partner is not Next Page able. **82544GC/EI** only: Bit 2 = 1 if MODE[3:0] is not 001xb or 0111b.	RO	0b MODE[3:0]	0b MODE [3:0]
Parallel Detection Fault	4	1b = A fault has been detected via the Parallel Detection function. 0b = A fault has not been detected via the Parallel Detection function.	RO/ LH[1]	0b	0b
Base Page	5	**82541xx** and **82547GI/EI** only: This bit indicates the status of the Auto-Negotiation variable, base page. If flags synchronization with the Auto-Negotiation state diagram enabling detection of interrupted links. This bit is only used if PHY register 16, bit 1 (Alternate NP Feature) is set. 1b = base_page = true. 0b = base_page = false. **Note:** This is a reserved bit for all remaining Ethernet controllers.	RO/ LH[1]	0b	0b
Reserved	15:6	0000000000b	RO	Always 000h	

1. **82541xx** and **82547GI/EI** only.

NOTE: The ANE Register is not valid until the Auto-Negotiation complete bit in the PHY Status Register indicates completion of the Auto-Negotiation process.

13.4.7.1.8 Next Page Transmit Register NPT (07d; R/W)

Table 13-26. Next Page Transmit Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Message/ Unformatted Field	10:0	Transmit Code Word Bit 10:0. **82541xx** and **82547GI/EI** only: 11-bit message code field.	R/W	001h	001h
Toggle	11	Transmit Code Word Bit 11. **82541xx** and **82547GI/EI** only: 1b = Previous value of the transmitted Link Code Word = logical 0. 0b = Previous value of the transmitted Link Code Word = logical 1.	RO	0b	0b
Acknowledge 2	12	Transmit Code Word Bit 12. **82541xx** and **82547GI/EI** only: 1b = Complies with message. 0b = Cannot comply with message.	R/W	0b	0b
Message Page	13	Transmit Code Word Bit 13. **82541xx** and **82547GI/EI** only: 1b = Message page. 0b = Unformatted page.	R/W	1b 0b[1]	1b 0b[1]
Reserved	14	Transmit Code Word Bit 14. **82541xx** and **82547GI/EI** only: Always read as 0b. Write to 0b for normal operation.	RO	0b	0b
Next Page	15	Transmit Code Word Bit 15. A write to the NPT Register implicitly sets a variable in the Auto-Negotiation state machine. **82541xx** and **82547GI/EI** only: 1b = Additional next pages follow. 0b = Last page.	R/W	0b	0b

1. **82541xx** and **82547GI/EI** only.

13.4.7.1.9 Link Partner Next Page Register LPN (08d; R)

Table 13-27. Link Partner Next Page Register Bit Description

Bit(s)	Field	Description	Mode	HW Rst	SW Rst
10:0	Message/ Unformatted Field	Received Code Word Bit 10:0. **82541xx** and **82547GI/EI** only: 11-bit message code field.	RO	000h	000h
11	Toggle	Received Code Word Bit 11. **82541xx** and **82547GI/EI** only: 1b = Previous value of the transmitted Link Code Word = a logic 0. 0b = Previous value of the transmitted Link Code Word = a logic 1.	RO	0b	0b
12	Acknowledge 2	Received Code Word Bit 12. **82541xx** and **82547GI/EI** only: 1b = Link Partner complies with the message. 0b = Link Partner cannot comply with the message.	RO	0b	0b
13	Message Page	Received Code Word Bit 13. **82541xx** and **82547GI/EI** only: 1b = Page sent by the Link Partner is a Message Page. 0b = Page sent by the Link Partner is an Unformatted Page.	RO	0b	0b
14	Acknowledge	Received Code Word Bit 14. **82541xx** and **82547GI/EI** only: 1b = Link Partner has received Link Code Word from the PHY. 0b = Link Partner has not received Link Code Word from the PHY.	RO	0b	0b
15	Next Page	Received Code Word Bit 15. **82541xx** and **82547GI/EI** only: 1b = Link Partner has additional next pages to send. 0b = Link Partner has no additional next pages to send.	RO	0b	0b

13.4.7.1.10 1000BASE-T Control Register GCON (09d; R/W)

Table 13-28. 1000BASE-T Control Register Bit Description

Bit(s)	Field	Description	Mode	HW Rst	SW Rst
7:0	Reserved	Reserved. Should be set to 00000000b.	R/W	0b	0b
8	1000BASE-T Half Duplex	1b = Advertise. 0b = Not advertised. **82544GC/EI** only: Bit 8 = ANEG[2]. **82541xx** and **82547GI/EI** only: 1b = DTE is 1000BASE-T capable. 0b = DTE is not 1000BASE-T capable. This bit is used by Smart Negotiation.	R/W	0b ANEG[2]	Retain
9	1000BASE-T Full Duplex	1b = Advertise. 0b = Not advertised. **82544GC/EI** only: Bit 9 = ANEG[3]. **82541xx** and **82547GI/EI** only: 1b = DTE is 1000BASE-T full duplex capable. 0b = DTE is not 1000BASE-T full duplex capable. This bit is used by Smart Negotiation (see Note 3).	R/W	1b ANEG[3]	Retain
10	Port Type	1b = Prefer multi-port device (MASTER). 0b = Prefer single port device (SLAVE)[1]. **82544GC/EI** only: Bit 10 =!ANEG[0]. Bit 10 is ignored if bit 12 = 1b.	R/W	1b[2] ANEG[0]	Retain
11	MASTER/SLAVE Configuration Value	1b = Manual configure as MASTER[3]. 0b = Manual configure as SLAVE[3]. **82544GC/EI** only: Bit 11 =!ANEG[0]. Bit 11 is ignored if bit 12 = 0b.	R/W	1b[2] ANEG[0]	Retain
12	MASTER/SLAVE Manual Configuration Enable	1b = Manual MASTER/SLAVE configuration. 0b = Automatic MASTER/SLAVE configuration. **82544GC/EI** only: Bit 12 =!ANEG[1].	R/W	1b[2] ANEG[1]	Retain

Table 13-28. 1000BASE-T Control Register Bit Description

Bit(s)	Field	Description	Mode	HW Rst	SW Rst
15:13	Test mode	000b = Normal Mode. 001b = Test Mode 1 - Transmit Waveform Test. 010b = Test Mode 2 - Transmit Jitter Test (MASTER mode). 011b = Test Mode 3 - Transmit Jitter Test (SLAVE mode). 100b = Test Mode 4 - Transmit Distortion Test. 101b, 110b, 111b = Reserved.	R/W	000b	000b

1. For the **82541xx** and **82547GI/EI**, only when PHY register 9, bit 12 is set to a logical 0.
2. 0b for the **82541xx** and **82547GI/EI** only
3. For the **82541xx** and **82547GI/EI**, only when PHY register 9, bit 12 is set to a logical 1.

NOTES:

1. Values programmed in bits 12:8 of the 1000BASE-T Control Register have no effect unless Auto-Negotiation is restarted (PHY Control Register, bit 9) or the link goes down. These bits can also be overridden by the PHY Control Register.
2. The symbol "!" is equivalent to logical "not."
3. For the **82541xx** and **82547GI/EI**, the default for bit 9 is affected by configuration bits in the EEPROM. If EEPROM ANI1000DIS is asserted, then the default is set to 0b. If EEPROM ADV10LU is asserted, then the default is set to 0b.

13.4.7.1.11 1000BASE-T Status Register GSTATUS (10d; R)

Table 14-29. 1000BASE-T Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Idle Error Count	7:0	Idle Error Counter. The counter stops at 1111b 1111b and does not roll over.	RO, SC	0000b 0000b	0000b 0000b
Reserved	9:8	Reserved. Should be set to 00b	RO	00b	00b
Link Partner 1000BASE-T Half Duplex Capability	10	1b = Link Partner is capable of 1000BASE-T half duplex. 0b = Link Partner is not capable of 1000BASE-T half duplex. Values in bits 11:10 are not valid until the ANE Register Page Received bit equals 1b.	RO	0b	0b
Link Partner 1000BASE-T Full Duplex Capability	11	1b = Link Partner is capable of 1000BASE-T full duplex. 0b = Link Partner is not capable of 1000BASE-T full duplex. Values in bits 11:10 are not valid until the ANE Register Page Received bit equals 1b.	RO	0b	0b
Remote Receiver Status	12	1b = Remote Receiver OK. 0 b = Remote Receiver Not OK.	RO	0b	0b

Table 14-29. 1000BASE-T Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Local Receiver Status	13	1b = Local Receiver OK. 0b = Local Receiver Not OK.	RO	0b	0b
MASTER/SLAVE Configuration Resolution	14	1b = Local PHY configuration resolved to MASTER. 0b = Local PHY configuration resolved to SLAVE.	RO	0b	0b
MASTER/SLAVE Configuration Fault	15	1b = MASTER/SLAVE configuration fault detected. 0b = No MASTER/SLAVE configuration fault detected.	RO, LH, SC	0b	0b

13.4.7.1.12 Extended PHY Status Register EPSTATUS (15d; R)

Table 13-30. Extended PHY Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	11:0	Reserved. Should be set to 000000000000b.	RO	000h	000h
1000BASE-T Half Duplex	12	1b = 1000BASE-T half duplex capable. 0b = not 1000BASE-T half duplex capable. **Note:** 1000Mb-Half duplex is NOT supported.	RO	1b[1]	1b[1]
1000BASE-T Full Duplex	13	1b = 1000BASE-T full duplex capable. 0b = Not 1000BASE-T full duplex capable.	RO	1b[1]	1b[1]
1000BASE-X Half Duplex	14	1b =1000BASE-X half duplex capable. 0b = Not 1000BASE-X half duplex capable. **Note:** 1000Mb-Half Duplex is NOT supported.	RO	0b[1]	0b[1]
1000BASE-X Full Duplex	15	1b =1000BASE-X full duplex capable. 0b = Not 1000BASE-X full duplex capable.	RO	0b[1]	0b[1]

1. MODE[3:0] **82544GC/EI** only.

NOTES:
1. 1000BASE-X Half Duplex only applicable to the **82544GC/EI**.
2. Bit 12 = bit 13 = 1b if MODE[3:0] does not = 001xb or 0111b.
3. Bit 14 = bit 15 = 1b if MODE[3:0] = 001xb or 0111b.

13.4.7.1.13 PHY Specific Control Register PSCON (16d; R/W)

Table 13-31. PHY Specific Control Register Bit Description

Field	Bit(s)	Description: 1000BASE-T	Description: 10/100BASE-T	Mode	HW Rst	SW Rst
Disable Jabber	0	1b = Disable jabber function. 0b = Enable jabber function. Jabber has effect only in 10BASE-T half duplex mode.		R/W	0b	Retain
Polarity Reversal	1	1b = Polarity Reversal Disabled. 0b = Polarity Reversal Enabled. If polarity is disabled, then the polarity is forced to be normal in 10BASE-T.		R/W	0b	Retain
SQE Test	2	1b = SQE test enabled. 0b = SQE test disabled. Jabber has effect only in 10BASE-T half duplex mode.		R/W	0b	Retain
MAC Interface Power Down	3	1b = Always power up. 0b = Can power down. This bit determines whether the MAC interface powers down when register PHY Control Register bit 11 is used to power down the Ethernet controller or when the PHY enters the energy detect state.		R/W	1b	Update
Disable 125CLK[1] Reserved	4	1b = 125CLK Low. 0b = 125CLK Toggling. Bit 4 = ENA_XC. This bit is reserved for all Ethernet controllers except the **82544GC/EI**. Should be set to 0b.		R/W	DIS_ 125CLK[1] 0b	Update
MDI Crossover Mode	6:5	00b = Manual MDI configuration. 01b = Manual MDI-X configuration. 10b = Reserved. 11b = Enable automatic crossover for all modes. **82544GC/EI** only: Bit 6 = DIS_125. Bit 5 = ENA_XC.		R/W	11b	Update
Enable Extended Distance	7	1b = Lower 10BASE-T receive threshold. 0b = Normal 10BASE-T receive threshold. When a cable longer than 100 m is used, the 10BASE-T receive threshold must be lowered in order to detect incoming signals.		R/W	0b	Retain

Table 13-31. PHY Specific Control Register Bit Description

Field	Bit(s)	Description		Mode	HW Rst	SW Rst
		1000BASE-T	10/100BASE-T			
Energy Detect	9:8	Energy Detect. 0xb = Off. 10b = Sense only on receive. 11b = Sense and periodically transmit NLP.		R/W	0b	Retain
Force Link Good	10	1b = Force link good. 0 b = Normal operation. If link is forced to be good, the link state machine is bypassed and the link is always up. In 1000BASE-T mode this has no effect.		R/W	0b	Retain
Assert CRS on Transmit	11	1b = Assert on transmit. 0b = Never assert on transmit. This bit has no effect in full duplex.		R/W	0b	Retain
Receive FIFO depth	13:12	00b = ± 16 Bits 01b = ± 24 Bits 10b = ± 32 Bits 11b = ± 40 Bits	00b = ± 8 Bits 01b = ± 12 Bits 10b = ± 16 Bits 11b = ± 20 Bits	R/W	00b	Retain
		Receive FIFO is enabled in internal SerDes mode[2] or serial interface mode.				
Transmit FIFO depth	15:14	00b = ± 16 Bits 01b = ± 24 Bits 10b = ± 32 Bits 11b = ± 40 Bits	00 b= ± 8 Bits 01b = ± 12 Bits 10b = ± 16 Bits 11b = ± 20 Bits	R/W	00b	Retain
		Transmit FIFO is enabled in 1000BASE-T mode or serial interface mode.				

1. **82544GC/EI** only.
2. TBI Mode for the **82544GC/EI**.

PHY Port Configuration Register (82541xx and 82547GI/EI Only) PPCONF (16d; R/W)

Table 13-32. PHY Port Configuration Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	0	Write to 0b for normal operation.	R/W	0b	0b
Alternate NP Feature	1	1b = Enable alternate Auto-Negotiate next page feature. 0b = Disable alternate Auto-Negotiate next page feature. If polarity is disabled, then the polarity is forced to be normal in 10BASE-T.	R/W	0b	0b
Reserved	3:2	Write to 0b for normal operation.	R/W	0b	0b

Table 13-32. PHY Port Configuration Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Auto MDIX Parallel Detect Bypass	4	Auto_MDIX Parallel Detect Bypass. Bypasses the fix to IEEE auto-MDIX algorithm for the case where the PHY is in forced-speed mode and the link partner is auto-negotiating. 1b = Strict 802.3 Auto-MDIX algorithm. 0b = Auto-MDIX algorithm handles Auto-Negotiation disabled modes. This is accomplished by lengthening the auto-MDIX switch timer before attempting to swap pairs on the first time out.	R/W	0b	0b
PRE_EN	5	Preamble Enable 0b = Set RX_DV high coincident with SFD. 1b = Set RX_DV high and RXD = preamble (after CRS is asserted).	R/W	1b	1b
Reserved	6	Write to 0b for normal operation.	R/W	0b	0b
Smart Speed	7	1b = Smart Speed selection enabled. 0b = Smart Speed selection disabled.	R/W	0b	0b[1]
TP Loopback (10BASE-T)	8	1b = Disable TP loopback during half-duplex operation. 0b = Normal operation.	R/W	1b	1b
Reserved	9	Write to 0b for normal operation.	R/W	0b	0b
Jabber (10BASE-T)	10	1b = Disable jabber. 0b = Enable jabber.	R/W	0b	0b
Bypass 4B5B (100BASE-TX)	11	1b = Bypass4B5B encoder and decoder. 0b = Normal operation.	R/W	0b	0b
Bypass Scramble (100BASE-TX)	12	1b = Bypass scrambler and descrambler. 0b = Normal operation.	R/W	0b	0b
Transmit Disable	13	1b = Disable twisted-pair transmitter. 0b = Normal operation.	R/W	0b	0b
Link Disable	14	1b = Force link pass 0b = Normal operation For 10BASE-T, this bit forces the link signals to be active. In 100BASE-T mode, setting this bit should force the Link Monitor into it's LINKGOOD state. For Gigabit operation, this merely bypasses Auto-Negotiation—the link signals still correctly indicate the appropriate status.	R/W	0b	0b
Reserved	16:15	Always read as 0b. Write 0b for normal operation.	R/W	0b	0b

1. The default for this bit is determined by EEPROM bit SSPEED.

13.4.7.1.14 PHY Specific Status Register PSSTAT (17d; R)

Table 13-33. PHY Specific Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Jabber (real time)	0	1b = Jabber. 0 = No jabber.	RO	0b	Retain
Polarity (real time)	1	1b = Reversed. 0b = Normal.	RO	0b	0b
Receive Pause Enable	2	1b = Receive pause enabled. 0b = Receive pause disabled. The Receive Pause Enable bit is valid only after the Speed and Duplex Resolved bit (11) is set. This occurs when Auto-Negotiation is completed or Auto-Negotiation is disabled. This is a reflection of the MAC pause resolution.	RO	0b	0b
Transmit Pause Enabled	3	1b = Transmit pause enabled. 0b = Transmit pause disabled. The Transmit Pause Enable bit is valid only after the Speed and Duplex Resolved bit (11) is set. This occurs when Auto-Negotiation is completed or Auto-Negotiation is disabled.	RO	0b	0b
Energy Detect Status	4	0b = Active 1b = Sleep	RO	0b	0b
Downshift Status	5	1b = Downshift. 0b = No Downshift.	RO	0b	0b
MDI Crossover Status	6	1b = MDI-X. 0b = MDI. The MDI Crossover Status bit is valid only after the Speed and Duplex Resolved bit (11) is set. This occurs when Auto-Negotiation is completed or Auto-Negotiation is disabled.	RO	0b	0b
Cable Length (100/1000 modes only)	9:7	000b = < 50 m. 001b = 50 – 80 m. 010b = 80 – 110 m. 011b = 110 – 140 m. 100b = >140 m. Cable length measurement is only a rough estimate. The actual value depends on the attenuation of the cable, output levels of the remote transceiver, connector impedance, etc.	RO	000b	000b
Link (real time)	10	1b = Link up. 0b = Link down.	RO	0b	0b

Table 13-33. PHY Specific Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Speed and Duplex Resolved	11	1b = Resolved. 0b = Not resolved. Speed, Duplex, MDI Crossover Status, Transmit Pause Enable, and Receive Pause Enable bits are valid only after the Speed and Duplex Resolved bit (11) is set. This occurs when Auto-Negotiation is completed or Auto-Negotiation is disabled.	RO	0b	0b
Page Received	12	1b = Page received. 0b = Page not received.	RO, LH	0b	0b
Duplex	13	1b = Full duplex. 0b = Half duplex. The Duplex bit is valid only after the Speed and Duplex Resolved bit (11) is set. This occurs when Auto-Negotiation is completed or Auto-Negotiation is disabled.	RO	0b	Retain
Speed	15:14	11b = Reserved. 10b = 1000 Mb/s. 01b = 100 Mb/s. 00b = 10 Mb/s. The Speed bit is valid only after the Speed and Duplex Resolved bit (11) is set. This occurs when Auto-Negotiation is completed or Auto-Negotiation is disabled.	RO	00b	Retain

PHY Port Status 1 Register (82541xx and 82547GI/EI Only)
PPSTAT (17d; R)

Table 13-34. PHY Status 1 Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
LFIT Indicator	0	Status bit indicating the Auto-Negotiation Link Fail Inhibit Timer has expired. This indicates that the Auto-Negotiation process completed page exchanges but was unable to bring up the selected MAU's link. 1b = Auto-Negotiation has aborted Link establishment following normal page exchange. 0b = Auto-Negotiation has either completed normally, or is still in progress. This bit is cleared when read or when one of the following occurs: Link comes up (PHY register 17, bit 10 = 1b). Auto-Negotiation is disabled (PHY register 0, bit 12 = 0b). Auto-Negotiation is restarted (PHY register 0, bit 9 = 1b).	RO/ LH/ SC	0b	0b
Polarity Status	1	1b = 10BASE-T polarity is reversed. 0b = 10BASE-T polarity is normal.	RO	0b	0b
Reserved	8:2	Write to 0b for normal operation.	RO	0b	0b
Duplex Mode	9	1b = Full duplex. 0b = Half duplex.	RO	0b	0b
Link	10	Indicates the current status of the link. Differs from PHY register 1, bit 2 in that this bit changes anytime the link status changes. PHY register 1, bit 2 latches low and stays low until read regardless of link status. 1b = Link is currently up. 0b = Link is currently down.	RO	0b	0b
MDI-X Status	11	Status indicator of the current MDI/MDI-X state of the twisted pair interface. This status bit is valid regardless of the MAU selected. 1b = PHY has selected MDI-X (crossed over). 0b = PHY has selected MDI (NOT crossed over).	RO	0b	0b
Receive Status	12	1b = PHY currently receiving a packet. 0b = PHY receiver is IDLE. When in loopback, this bit reads as 0b.	RO	0b	0b

Table 13-34. PHY Status 1 Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Transmit Status	13	1b = PHY currently transmitting a packet. 0b = PHY transmitter is IDLE. When in loopback, this bit reads as 0b.	RO	0b	0b
Data Rate	15:14	00b = Reserved. 01b = PHY operating in 10BASE-T mode. 10b = PHY operating in 100BASE-TX mode. 11b = PHY operating in 1000BASE-T mode.	RO	0b	0b

13.4.7.1.15 PHY Interrupt Enable Register PINTE (18d; R/W)

Table 13-35. PHY Interrupt Enable Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Jabber Interrupt Enable	0	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
Polarity Changed Interrupt Enable	1	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
Reserved	3:2	Always write 00b.	R/W	0b	Retain
Energy Detect	4	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
Downshift Interrupt Enable	5	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
MDI Crossover Changed Interrupt Enable	6	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
FIFO Over/Underflow Interrupt Enable	7	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
False Carrier Interrupt Enable	8	1b = Interrupt enable. 0 b= Interrupt disable.	R/W	0b	Retain
Symbol Error Interrupt Enable	9	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
Link Status Changed Interrupt Enable	10	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
Auto-Negotiation Completed Interrupt Enable	11	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain

Table 13-35. PHY Interrupt Enable Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Page Received Interrupt Enable	12	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
Duplex Changed Interrupt Enable	13	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
Speed Changed Interrupt Enable	14	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain
Auto-Negotiation Error Interrupt Enable	15	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain

PHY Port Control Register (82541xx and 82547GI/EI Only) PPCONT (18d; R/W)

Table 13-36. PHY Port Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	3:0	Always read as 0b. Write to 0b for normal operation.	R/W	0b	0b
TP Loopback	4	Allow gigabit loopback on twisted pairs.	R/W	0b	0b
Fast Downshift Enable	5	**82541EI/82547GI** (B0 stepping): Reserved, write to 0b. **82541/GI/ER** and **82547GI** (B1 stepping): Fast 1000 Mb to 100 Mb downshift enable. 0b = Downshift after 16 seconds. 1b = Downshift after 10 seconds.	R/W	0b	0b
Reserved	8:6	Always read as 0b. Write to 0b for normal operation.	R/W	0b	0b
Non-Compliant Scrambler Compensation	9	1b = Detect and correct for non-compliant scrambler. 0b = Detect and report non-compliant scrambler.	R/W	0b	0b[1]
TEN_CRS_Select	10	1b = Extend CRS to cover GMII latency and RX_DV. 0b = Do not extend CRS (RX_DV can continue past CRS).	R/W	1b	1b
Reserved	11	Always write as 1b for normal operation.	R/W	1b	1b
Auto-MDI-X	12	Auto-MDI-X algorithm enable. 1b = Enable Auto-MDI-X mode. 0b = Disable Auto-MDI-X mode (manual mode). **Note:** When forcing speed to 10Base-T or 100Base-T, use manual mode. Clear the bit and set PHY register 18, bit 13 according to the required MDI-X mode.	R/W	1b	1b

Table 13-36. PHY Port Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
MDI-X Mode	13	Force MDI-X mode. Valid only when operating in manual mode. (PHY register 18, bit 12 = 0b. 1b = MDI-X (cross over). 0b = MDI (no cross over).	R/W	0b	0b
Reserved	14	Always read as 0b. Write to 0b for normal operation.	R/W	0b	0b
Jitter Test Clock	15	This configuration bit is used to enable the Ethernet controller to drive its differential transmit clock out through the appropriate Analog Test (ATEST+/-) output pads. This feature is required in order to demonstrate conformance to the IEEE Clause 40 jitter specification. When high, it sends Jitter Test Clock out. This bit works in conjunction with internal PHY register 18, bit 15. In order to have the clock probed out, it is required to perform the following write sequence: PHY register 18, bit15 = 1b PHY register 31 = 4000h (page select) PHY register 17 = 0080h PHY register 31 = 0000h (page select)	R/W	0b	0b

1. The default for this bit is determined by EEPROM configuration bits. If EEPROM bit NCSCRAMB is asserted, then the default is set to 1b.

13.4.7.1.16 PHY Interrupt Status Register PINTS (19d; R)

Table 13-37. PHY Interrupt Status Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Jabber	0	1b = Jabber. 0b = No jabber.	RO, LH	0b	0b
Polarity Changed	1	1b = Polarity Changed. 0b = Polarity not changed.	RO, LH	0b	0b
Reserved	3:2	Reserved. Should be set to 00b.	RO	Always 00b	
Energy Detect	4	1b = Energy Detect state changed 0b = No state change detected	RO, LH	0b	0b
Downshift Detected	5	1b = Downshift detected. 0b = No down shift.	RO, LH	0b	0b
MDI Crossover Changed	6	1b = Crossover changed. 0b = Crossover not changed.	RO, LH	0b	0b

Table 13-37. PHY Interrupt Status Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
FIFO Over/Underflow	7	1b = Over/Underflow Error. 0b = No FIFO Error.	RO, LH	0b	0b
False Carrier	8	1b = False carrier. 0b = No false carrier.	RO, LH	0b	0b
Symbol Error	9	1b = Symbol error. 0b = No symbol error.	RO, LH	0b	0b
Link Status Changed	10	1b = Link status changed. 0b = Link status not changed.	RO, LH	0b	0b
Auto-Negotiation Completed	11	1b = Auto-Negotiation completed. 0b = Auto-Negotiation not completed.	RO, LH	0b	0b
Page Received	12	1b = Page received. 0b = Page not received.	RO, LH	0b	0b
Duplex Changed	13	1b = Duplex changed. 0b = Duplex not changed.	RO, LH	0b	0b
Speed Changed	14	1b = Speed changed. 0b = Speed not changed.	RO, LH	0b	0b
Auto-Negotiation Error	15	1b = Auto-Negotiation Error. 0b = No Auto-Negotiation Error. An error occurs if MASTER/SLAVE does not resolve, parallel detect fault, no common HCD, or link does not validate after negotiation has completed.	RO, LH	0b	0b

PHY Link Health Register (82541xx and 82547GI/EI Only) PLINK (19d; R)

Table 13-38. PHY Link Health Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Valid Channel A	0	The channel A DSP had converged to incoming data.	RO	0b	0b
Valid Channel B	1	The channel B DSP had converged to incoming data.	RO	0b	0b
Valid Channel C	2	The channel C DSP had converged to incoming data.	RO	0b	0b
Valid Channel D	3	The channel D DSP had converged to incoming data.	RO	0b	0b
Auto-Negotiation Active	4	Auto-Negotiate is actively deciding HCD.	RO	0b	0b
Reserved	5	Always read as 0b.	RO	0b	0b

Table 13-38. PHY Link Health Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Auto-Negotiation Fault	6	Auto-Negotiate Fault: This is the logical OR of PHY register 1, bit 4, PHY register 6, bit 4, and PHY register 10, bit 15.	RO	0b	0b
Reserved	7	Always read as 0b.	RO	0b	0b
Data Err[0]	8	Mode: 10: 10 Mbps polarity error. 100: Symbol error. 1000: Gig idle error.	LH	0b	0b
Data Err[1]	9	Mode: 10: Reserved. 100: Scrambler unlocked. 1000: Local receiver not OK.	RO/ LH	0b	0b
Count Overflow	10	The idle error counter has overflowed.	RO/ LH	0b	0b
Gigabit Rem Rcvr NOK	11	Gig has detected a remote receiver status error. This is a latched high version of PHY register 10, bit 12.	RO/ LH	0b	0b
Gigabit Master Resolution	12	Gig has resolved to master. This is a duplicate of PHY register 10, bit 14. Programmers must read PHY register 10, bit 14 to clear this bit.	RO	0b	0b
Gigabit Master Fault	13	A fault has occurred with the gig master/ slave resolution process. This is a copy of PHY register 10, bit 15. Programmers must read PHY register 10, bit 15 to clear this bit.	RO	0b	0b
Gigabit Scrambler Error	14	1b indicates that the PHY has detected gigabit connection errors that are most likely due to a non-IEEE compliant scrambler in the link partner. 0b = Normal scrambled data. Definition is: If an_enable is true and in Gigabit mode, on the rising edge of internal signal link_fail_inibit timer_done, the dsp_lock is true but loc_rcvr_OK is false.	RO	0b	0b
SS Downgrade	15	Smart Speed has downgraded the link speed from the maximum advertised.	RO/ LH	0b	0b

13.4.7.1.17 Extended PHY Specific Control Register 1[1] EPSCON1 (20d; R/W)

Table 13-39. Extended PHY Specific Control 1 Bit Description[1]

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	1:0	00b	R/W	00b	Retain
Reserved	3	0b	R/W	0b	0
Reserved	6:4	Reserved. Should be set to 0b. Changes to this bit are disruptive to the normal operation; any change to this register must be followed by software reset to take effect.	R/W	110b	Update
Reserved	7	Reserved. Should be set to 0b. Changes to this bit are disruptive to the normal operation; any change to this register must be followed by software reset to take effect.	R/W	0b	Update
Slave downshift counter	9:8	00b = disable. (10/100 downshift) 01b = 1x. 10b = 2x. 11b = 3x. Changes to this bit are disruptive to the normal operation; any change to this register must be followed by software reset to take effect. Bits 11:10 have no effect unless bits 1:0 are set to their default values.	R/W	01b	Update
Master downshift counter	11:10	00b = 1x. 01b = 2x. 10b = 3x. 11b = 4x. Number of times that the PHY attempts to achieve gigabit link before downshifting to the next speed in Master Mode. Changes to this bit are disruptive to the normal operation; any change to this register must be followed by software reset to take effect. Bits 11:10 have no effect unless bits 1:0 are set to their default values.	R/W	11b	Update
Reserved	12	Reserved. Should be set to 0b.	R/W	0b	0b
Reserved	13	Reserved. Should be set to 0b.	R/W	0b	0b
Reserved	14	Reserved. Should be set to 0b.	R/W	0b	0b
Reserved	15	Reserved. Should be set to 0b.	R/W	0b	0b

1. Extended PHY Specific Control Register - EPSCON for the **82544GC/EI** only.

GMII FIFO Register (82541xx and 82547GI/EI Only)
PFIFO (20d; R/W)

Table 13-40. GMII FIFO Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Buffer Size	3:0	An unsigned integer that stipulates the number of write clocks to delay the read controller after internal GMII's tx_en is first asserted. This "buffer" protects from underflow at the expense of latency. The maximum value that can be set is 13d or Dh.	R/W	0101b	0101b
Enable Speed-Up Upon Cable Reconnect	4	When set, the PHY advertises higher speed than 10Base-T after reconnect of the cable, even if the software advertised only 10Base-T speed.	R/W	Note 2	Note 2
Power Down On Link Disconnect	5	When set, the PHY optimizes the power consumption when the cable is disconnected. The PHY gets back to normal operation reconnect of the cable, supporting Auto-Negotiation and parallel detection.	R/W	Note 1	Note 1
Reserved	7:6	Always read as 0b. Write to 0b for normal operation.	R/W	00b	00b
FIFO Out Steering	9:8	00b, 01b: Enable the output data bus from GMII FIFO to transmitters, drives zeros on the output loop-back bus from GMII FIFO to external application and to DSP RX-FIFOs in test mode. 10b: Drive zeros on output bus from GMII FIFO to transmitters, enable data on the output loop-back bus from GMII FIFO to external application and to DSP RX-FIFOs in test mode. 11b: Enable the output data bus from GMII FIFO to both transmitters and loop-back bus.	R/W	00b	00b
Disable Error Out	10	When set, disables the addition of under/overflow errors to the output data stream on internal GMII's tx_error.	R/W	0b	0b
Reserved	13:11	Always read as 0b. Write to 0b for normal operation.	R/W	0b	0b
FIFO Overflow	14	Status bit set when read clock that is faster than internal GMII's gtx_clk empties the FIFO mid packet. Increase the buffer size.	RO/ LH	0b	0b
FIFO Underflow	15	Status bit set when read clock that is slower than internal GMII's gtx_clk has allowed the FIFO to fill to capacity mid packet. Decrease buffer size.	RO/ LH	0b	0b

NOTES:
1. The default is determined by EEPROM bit SPD_EN.
2. The default is determined by EEPROM bit ADV10LU.

13.4.7.1.18 PHY Receive Error Counter PREC (21d; R)

Table 13-41. PHY Receive Error Counter Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW RST
Receive Error Count	15:0	Error Count.	RO,SC	0000h	0000h

NOTE: The counter stops at FFFFh and does not roll over.

PHY Channel Quality Register (82541xx and 82547GI/EI Only) PCHAN (21d; R)

Table 13-42. PHY Channel Quality Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
MSE D	3:0	The converged mean square error for Channel D.	RO	0b	0b
MSE C	7:4	The converged mean square error for Channel C.	RO	0b	0b
MSE B	11:8	The converged mean square error for Channel B.	RO	0b	0b
MSE A	15:12	The converged mean square error for Channel A. This field is only meaningful in gigabit, or in 100BASE-TX if this is the receive pair. Use of this field is complex and needs interpretation based on the chosen threshold value.	RO	0b	0b

13.4.7.1.19 SPEED_TEN_LED and LINK_ACT_LED Control (82541xx and 82547GI/EI Only) (23d; R/W)

Table 13-43. SPEED_TEN_LED and LINK_ACT_LED Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
LED Source Select	3:0	MUX the designated input to SPEED_TEN_LED.	R/W	0000b	0000b
LED Blink Disable	4	Disable the SPEED_TEN_LED Blink Logic. 0b = Enable logic. 1b = Disable logic.	R/W	0b	0b

Table 13-43. SPEED_TEN_LED and LINK_ACT_LED Bit Description

LED Stretch Disable	5	Disable the SPEED_TEN_LED Extension Logic. 0b = Enable logic. 1b = Disable logic. **Note:** Only when both the stretch and blink are disabled the input bypasses the blink logic and is muxed out with no sampling (only combinational logic).	R/W	1b	1b
LED Source Select	9:6	Mux the designated input to LED_ACT_LED.	R/W	0001b	0001b
LED Blink Disable	10	Disable the LINK_ACT_LED Blink Logic. 0b = Enable logic. 1b = Disable logic.	R/W	1b	1b
LED Stretch Disable	11	Disable the LINK_ACT_LED Extension Logic. 0b = Enable logic. 1b = Disable logic. **Note:** Only when both the stretch and blink are disabled the input bypasses the blink logic and is muxed out with no sampling (only combinational logic).	R/W	0b	0b
Invert Select	12	When set to 1b, all LED outputs are inverted.	R/W	0b	0b
Reserved	14:13	Always read as 0b. Write to 0b for normal operation	R/W	00b	00b
Disable 10 Power Saving	15	This bit is used to disable special power saving in 10BASE-T mode and parallel detection. When set to 1b, power reduction features of 10BASE-10 are disabled (reserved for customers).	R/W	0b	0b

13.4.7.1.20 PHY Global Status (82544GC/EI Only) PGSTAT (23d; R)

Table 13-44. PHY Global Status Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Port 0 Interrupt	0	0b = No Interrupt on Port. 1b = Interrupt on Port.	RO	0b	0b
Port 1 Interrupt	1	0b = No Interrupt on Port. 1b = Interrupt on Port.	RO	0b	0b
Port 2 Interrupt	2	0b = No Interrupt on Port. 1b = Interrupt on Port.	RO	0b	0b
Port 3 Interrupt	3	0b = No Interrupt on Port. 1b = Interrupt on Port.	RO	0b	0b
Reserved	15:4	Reserved. Should be set to 0b.	RO	0b	0b

NOTE: Bits 3:0 remain high until the active corresponding interrupt bits are cleared on a read of the PHY Interrupt Status Register.

13.4.7.1.21 SPEED_100_LED and SPEED_1000_LED Control (82541xx and 82547GI/EI Only) (24d; R/W)

Table 13-45. SPEED_100_LED and SPEED_1000_LED Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
LED Source Select	3:0	MUX the designated input to SPEED_100_LED.	R/W	0011b	0011b
LED Blink Disable	4	Disable the SPEED_100_LED Blink Logic. 0b = Enable logic. 1b = Disable logic.	R/W	0b	0b
LED Stretch Disable	5	Disable the SPEED_100_LED Extension Logic. 0b = Enable logic. 1b = Disable logic. **Note:** Only when both the stretch and blink are disabled the input bypasses the blink logic and is muxed out with no sampling (only combinational logic).	R/W	1b	1b
LED Source Select	9:6	Mux the designated input to SPEED_1000_LED.	R/W	0100b	0100b
LED Blink Disable	10	Disable the SPEED_1000_LED Blink Logic. 0b = Enable logic. 1b = Disable logic.	R/W	0b	0b

Table 13-45. SPEED_100_LED and SPEED_1000_LED Bit Description

LED Stretch Disable	11	Disable the SPEED_1000_LED Extension Logic. 0b = Enable logic. 1b = Disable logic. **Note:** Only when both the stretch and blink are disabled the input bypasses the blink logic and is muxed out with no sampling (only combinational logic).	R/W	1b	1b
Reserved	15:12	Always read as 0b. Write to 0b for normal operation	R/W	00b	00b

13.4.7.1.22 PHY LED Control Register (82544GC/EI Only) PLED (24d; R/W)

Table 13-46. PHY LED Control Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
LED_TX control	0	1b = Activity/Link. 0b = Transmit activity.	R/W	0b	Retain
LED_RX control	1	1b = Receive activity/Link. 0b = Receive activity.	R/W	0b	Retain
Reserved	2	Reserved. Should be set to 0b.	R/W	0b	Retain
LED_LINK control	4:3	1xb = Link[2:1], Link. 01b = Link, Speed[1:0]. 00b = Link[2:0].	R/W	00b	Retain
Reserved	7:5	Reserved. Should be set to 000b.	R/W	000b	Retain
Blink Rate	10:8	000b = 42 ms. 001b = 84 ms. 010b = 170 ms. 011b = 340 ms. 100b = 670 ms. 101b to 111b = Reserved.	R/W	001b	Retain
Force INT# to Assert	11	0b = Do not force INT# assertion. 1b = Force INT# assertion.	R/W	0b	Retain
Pulse stretch duration	14:12	000b = no pulse stretching. 001b = 21 ms to 42 ms. 010b = 42 ms to 84 ms. 01b1 = 84 ms to 170 ms. 100b = 170 ms to 340 ms. 101b = 340 ms to 670 ms. 110b = 670 ms to 1.3 s. 111b = 1.3s to 2.7 s.	R/W	100b	Retain
Disable LED	15	0b = Enable. 1b = Disable.	R/W	0b	Retain

13.4.7.1.23 Extended PHY Specific Control Register 2 EPSCON2 (26d; R/W)

Table 13-47. Extended PHY Specific Control Register 2 Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Fiber Output Amplitude	2:0	111b = 1.2, 100b = 0.9, 001b = 0.6 110b = 1.1, 011b = 0.8, 000b = 0.5 101b = 1.0, 010b = 0.7	R/W	100b	Retain
Reserved	3	Reserved. Should be set to 1b.	R/W	1b	1b
Reserved	4	Reserved. Should be set to 0b.	R/W	0b	0b
Fiber Output Impedance	5	1b= 75 ohm. 0b = 50 ohm.	R/W	0b	Update
Fiber Input Impedance	6	1b = 75 ohm. 0b = 50 ohm.	R/W	0b	Update
Reserved	15:7	Reserved. Should be set to 0b.	R/W	000h	000h

NOTE: Not applicable to the **82540EP/EM**, **82544GC/EI**, **82541xx**, or **82547GI/EI**.

13.4.7.1.24 Extended PHY Specific Status Register (82544GC/EI Only) EPSSTAT (27d; R)

Table 13-48. Extended PHY Specific Status Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
MODE[3:0]	27.3:0	MODE[3:0].	RO	MODE[3:0]	Retain
Reserved	27.15:4	Reserved. Should be set to 000000000000b.	RO	0b	0b

13.4.7.1.25 MDI Register 30 Page Select[1] R30PS (29d; WO)

Table 13-49. MDI Register 30 Page Select Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Register 30 Page Select	15:0	Selects the register accessible via the “window” at MDI register 30.	R/W	0000h	0000h

1. Not applicable to the **82544GC/EI**, **82541xx**, or **82547GI/EI**.

13.4.7.1.26 MDI Register 30 Access Window[1] R30AW (30d; R/W)

Table 13-50. MDI Register 30 Page Select Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Register 30 Access	15:0	Provides read/write capability for register selected via MDI register 29.	R/W	0000h	0000h

13.4.7.1.27 Documented MDI Register 30 Operations[1]

Unless otherwise specified, no reset operations are required in order for the following operations to take effect.

Table 13-51. MDI Register 30 Operations

To Perform	Operation	MDI Read/Write Sequence
Power down SerDes (optimize power for copper PHY operation)[1]	30_31.15:0 <= x2001h	• Write MDI register 29 <= 31d • Write MDI register 30 <= 2001h
Tune VCO on SerDes Rx for optimal Bit Error Ratio (BER)[1]	30_5.8 <= 0b	• Write MDI register 29 <= 5d • Read MDI register 30 • Change bit 8 to 0 • Write result to MDI register 30
	30_4.11 <= 1b	• Write MDI register 29 <= 4d • Read MDI register 30 • Change bit 11 to 1 • Write result to MDI register 30
Set PHY output drivers into Class A mode (Class AB is default after reset)	30_11.15:0 = 8004h	• Write MDI register 29 = 11d • Write MDI register 30 = 8004h

1. Not applicable to **82540EP/EM**

NOTE: Any time the PHY is reset it returns to Class AB drive mode.

1. Not applicable to the **82544GC/EI**, **82541xx**, or **82547GI/EI**.

13.4.7.1.28 PHY Page Select Register (82541xx and 82547GI/EI Only) PPAGE (31d; R/W)

Table 13-52. PHY Page Select Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
PAGE_SEL	15:0	This register is used to swap out the Base Page containing the IEEE registers for Intel reserved test and debug pages residing within the Extended Address space.	WO	0b	0b

13.4.8 Flow Control Address Low

FCAL (00028h; R/W)

Flow control packets are defined by IEEE 802.3x to be either a unique multicast address or the station address with the EtherType field indicating PAUSE. The FCAL, FCAH registers provide the value hardware compares incoming packets against to determine that it should PAUSE its output, and hardware use when transmit PAUSE packets to its remote node when flow control is activated.

The FCAL register contains the lower bits of the internal 48-bit Flow Control Ethernet address. All 32 bits are valid. Software can access the High and Low registers as a register pair if it can perform a 64-bit access to the PCI bus. This register should be programmed with 00_C2_80_01h. The complete flow control multicast address is: 01_80_C2_00_00_01h; where 01h is the first byte on the wire, 80h is the second, etc.

Table 13-53. FCAL Register Bit Description

31 ... 0
FCAL

Field	Bit(s)	Initial Value	Description
FCAL	31:0	X	Flow Control Address Low Should be programmed with 00_C2_80_01h

13.4.9 Flow Control Address High

FCAH (0002Ch; R/W)

This register contains the upper bits of the 48-bit Flow Control Ethernet address. Only the lower 16 bits of this register have meaning. The complete Flow Control address is {FCAH, FCAL}. This register should be programmed with 01_00h. The complete flow control multicast address is: 01_80_C2_00_00_01h; where 01h is the first byte on the wire, 80h is the second, etc.

Table 13-54. FCAH Register Bit Description

31 ... 16	15 ... 0
Reserved	FCAH

Field	Bit(s)	Initial Value	Description
FCAH	15:0	X	Flow Control Address High Should be programmed with 0100h.
Reserved	31:16	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

13.4.10 Flow Control Type

FCT (00030h; R/W)

This register contains the type field that hardware matches to recognize a flow control packet and that hardware uses when transmitting a PAUSE packet to its remote node. Only the lower 16 bits of this register have meaning. This register should be programmed with 88_08h. The upper byte is first on the wire FCT[15:8].

Table 13-55. FCT Register Bit Description

31 ... 16	15 ... 0
Reserved	FCT

Field	Bit(s)	Initial Value	Description
FCT	15:0	X	Flow Control Type Should be programmed with 88_08h.
Reserved	31:16	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

13.4.11 VLAN Ether Type

VET (00038h; R/W)

This register contains the type field hardware matches against to recognize an 802.1Q (VLAN) Ethernet packet and uses when add and transmit VLAN Ethernet packets. To be compliant with the 802.3ac standard, this register should be programmed with the value 8100h. For VLAN transmission the upper byte is first on the wire (VET[15:8]).

Table 13-56. VET Register Bit Description

31 ... 16	15 ... 0
Reserved	VET

Field	Bit(s)	Initial Value	Description
VET	15:0	X	VLAN EtherType Should be programmed with 8100h.
Reserved	31:16	0b	Reserved Reads as 0b.

13.4.12 Flow Control Transmit Timer Value

FCTTV (00170h; R/W)

Provides the Pause slot time value to be included in the transmitted XOFF Pause packets.

The slot time value that is used is a fixed slot of 64-byte time.

Table 13-57. FCTTV Register Bit Description

31 ... 16	15 ... 0
Reserved	TTV

Field	Bit(s)	Initial Value	Description
TTV	15:0	X	Transmit Timer Value Slot time value (Slot time value is 64-byte time) to be inserted into the transmitted Pause frame. If software wishes to send XON frame, it should set the FCTTV. TTV value to 0b prior to initializing the Pause frame.
Reserved	31:16	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

13.4.13 Transmit Configuration Word Register[1]

TXCW (00178h; R/W)

This register is applicable to the TBI mode/internal SerDes mode of operation. For internal PHY operation, program the register to 0000h. For example, clear this register in MMI mode.

This register has two meanings, depending on the state of Auto-Negotiation: one as the "AN advertise register" defined by IEEE 802.3z, and the other as a register for software control of the Auto-Negotiation process.

When performing hardware Auto-Negotiation, it fulfills the function defined by sub-clause 37.3.6.1.3 of IEEE 802.3z.

Table 13-58. TXCW Register Bit Description

31	30	29 16	15 0
A	Tx	Reserved	TxConfigWord

Field	Bit(s)	Initial Value	Description
TxConfigWord	15:0	0b	Data transmitted during Auto-Negotiation process. When performing hardware Auto-Negotiation (TXCW.ANE is set), the value of TxConfigWord is encoded as two 10-bit symbols and sent as the "config_word" field for the /C/ ordered set. When the Ethernet controller performs software Auto-Negotiation, TxConfig and TxConfigWord are used to negotiate with the link partner. Data stored in TxConfigWord is transmitted during the Auto-Negotiation process. Software should not read back the contents of this field as content might change during the software Auto-Negotiation process. In the absence of loss of synchronization, the value read back is stable and equal to the value written. The mapping of the TxConfigWord is as follows: • TxConfigWord[15] Next page request • TxConfigWord[14] Reserved (write as 0b; ignore on read) • TxConfigWord[13:12] Remote fault indication • TxConfigWord[11:9] Reserved (write as 0b; ignore on read) • TxConfigWord[8:7] Pause • TxConfigWord[6] Half-duplex • TxConfigWord[5] Full-duplex • TxConfigWord[4:0] Reserved (write as 0b; ignore on read) Bits 5,7 & 8 of TxConfigWord are loadable from the EEPROM upon power-up, or chip reset.
Reserved	29:16	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

1. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

Field	Bit(s)	Initial Value	Description
TxConfig	30	0b	Transmit Config Control bit 0b = Transmit data/idle 1b = Transmit /C/ ordered sets Setting the TxConfig bit causes transmission of /C/ ordered set in a software controlled Auto-Negotiation process (TXCW.ANE=0b).
ANE	31	0b	Auto-Negotiation Enable. 1b = Enable the hardware Auto-Negotiation state machine. 0b = Disable the hardware Auto-Negotiation state machine. This bit has the same function as bit 0.12 defined in sub-clause 22.2.4.1.4 of the 802.3z standard. Since this bit is a “static” value, a pulse is generated by hardware in response to writing this bit with a 1b. This pulse is used to restart the Auto-Negotiation state machine. When ANE is set, a transition from loss of synchronization to synchronized state restarts the Auto-Negotiation as well. If ANE is cleared, then the Ethernet controller is performing software Auto-Negotiation. In that case TxConfig and TxConfigWord are used to negotiate with the link partner. The ANE is loadable from the EEPROM upon power up or chip reset.

Note: Careful attention to the IEEE 802.3z standard is required in order to meet specified timing requirements for timing during a software negotiated link.

13.4.14 Receive Configuration Word Register[1]

RXCW (00180h; R)

This register has meaning only in TBI/internal SerDes mode of operation. The RXCW register records the partner abilities and provides indications about its Auto-Negotiation status.

The contents of this register depend on the state of TXCW.ANE. If ANE is set, then this register records the 16-bit defined in IEEE 802.3z. When performing software Auto-Negotiation, software should look for RXCW.ANC. When RXCW.ANC is set, the contents of RXCW.RxConfigWord are valid, when RXCW.ANC is cleared, then the content of this register is undefined.

Note: While in internal SerDes mode (**82546GB/EB** and **82545GM/EM** only), software might be required to inspect or monitor the results of RXCW to generate a link up/down indication.

1. Not applicable to the **82541xx**, **82547GI/EI**, or **82540EP/EM**.

Table 13-59. RXCW Register Bit Description

Field	Bit(s)	Initial Value	Description
RxConfigWord	15:0	X	Data received during Auto-Negotiation process. When performing hardware Auto-Negotiation (TXCW.ANE = 1b), the “AN link partner ability base page register” is recorded in the RxConfigWord. When TXCW.ANE is clear, then this register is used by software to perform software based Auto-Negotiation. In that capacity, RxConfigWord records the raw values returned from the Auto-Negotiation process.
Reserved	25:16	0b	Reserved Reads as 0b. Should be written with 0b for future compatibility.
RxConfigNoCarrier	26	0b	Carrier Sense Indication 0b = Ethernet controller not receiving idle characters (Carrier sense is true) 1b = Ethernet controller receiving idle characters (Carrier sense is false) Hardware sets RxConfigNoCarrier when it is receiving idle characters (and when receiving link configuration information or port status). Software can use this bit to determine that idles have been seen prior to exiting the link start-up procedure. **82544GC/EI** only: Valid only in software Auto-Negotiation mode (TXCW.ANE = 0b).
RxConfigInvalid	27	0b	Invalid Symbol during configuration process 0b = Have not received an invalid symbol during Auto-Negotiation process. 1b = Have received an invalid symbol during Auto-Negotiation process (bit is LH). If the Ethernet controller detects an invalid symbol at any time, it sets the RxConfigInvalid bit. The bit is latched high until read by software. Software is expected to restart the configuration process when the Ethernet controller receives an invalid symbol at any time during the Auto-Negotiation process. **82544GC/EI** only: Valid only in software Auto-Negotiation mode (TXCW.ANE = 0b).
RxConfigChange	28	0b	Change to the RxConfigWord indication 0b = RxConfigWord has changed since last read. 1b = RxConfig is unchanged since last read (LH) Indicates that the Ethernet controller interface has seen a change to the RxConfigWord. This bit is latched high until read by software. **82544GC/EI** only: Valid only in software Auto-Negotiation mode (TXCW.ANE = 0b).

Field	Bit(s)	Initial Value	Description
RxConfig	29	0b	/C/ order set reception indication 0b = Receive idle/data stream. 1b = Receiving /C/ order sets. Provides an indication as to whether the interface is receiving /C/ order set, or normal idle/data stream. **82544GC/EI** only: Valid only in software Auto-Negotiation mode (TXCW.ANE = 0b).
RxSynchronize	30	0b	Lost bit synchronization indication 0b = Lost synchronization. 1b = Bit synchronization (bit is LL). Used to qualify all other bits in the register (when in software Auto-Negotiation). Each time the Ethernet controller loses bit synchronization, this bit becomes ‘0b’ and stays ‘0b’ until read by software. **82544GC/EI** only: Valid only in software Auto-Negotiation mode (TXCW.ANE = 0b).
ANC	31	0b	Auto Negotiation Complete 0b = The contents of the register are undefined in hardware Auto-Negotiation mode (TXCW.ANE = 1b) 1b = The contents of the register are valid in hardware Auto-Negotiation mode. Reports the status as required in sub-clause 22.2.4.2.11. This bit remains cleared from the time Auto-Negotiation is reset until Auto-Negotiation reaches the “LINK_OK” state. It remains set until Auto-Negotiation is disabled or restarted. This bit has meaning only if TXCW.ANE = 1b. If software Auto-Negotiation is used, it always reads 0b and should be ignored.

13.4.15 LED Control[1]

LEDCTL (00E00h; RW)

31 - 24	23 - 16	15 - 8	7 - 0
LED3 (LINK1000#)	LED2 (LINK100#)	LED1 ACTIVITY#)	LED0 (LINK_UP#)

1. Not applicable to the **82544GC/EI**.

Table 13-60. LED Control Bit Description[1]

Field	Bit	Initial Value	Description
LED0_MODE	3:0	0010b[1]	**LED0/LINK# Mode**. This field specifies the control source for the LED0 output. An initial value of 0010b selects LINK_UP# indication.
Reserved	5:4	00b	**Reserved**. Read-only as 0b. Write as 0b for future compatibility.
LED0_IVRT	6	0b[1]	**LED0/LINK# Invert**. This field specifies the polarity/ inversion of the LED source prior to output or blink control. 0b = do not invert LED source. 1b = invert LED source.
LED0_BLINK	7	0b[1]	**LED0/LINK# Blink.** This field specifies whether to apply blink logic to the (possibly inverted) LED control source prior to the LED output. 0b = do not blink asserted LED output. 1b = blink asserted LED output.
LED1_MODE	11:8	0011b	**LED1/ACTIVITY# Mode**. This field specifies the control source for the LED1 output. An initial value of 0011b selects ACTIVITY# indication.
Reserved	13:12	00b	**Reserved**. Read-only as 0b. Write as 0b for future compatibility.
LED1_IVRT	14	0b	**LED1/ACTIVITY# Invert**.
LED1_BLINK	15	1b	**LED1/ACTIVITY# Blink.**
LED2_MODE	19:16	0110b[1]	**LED2/LINK100# Mode**. This field specifies the control source for the LED2 output. An initial value of 0011b selects LINK100# indication.
Reserved	21:20	00b	**Reserved**. Read-only as 0b. Write as 0b for future compatibility.
LED2_IVRT	22	0b[1]	**LED2/LINK100# Invert**.
LED2_BLINK	23	0b[1]	**LED2/LINK100# Blink.**
LED3_MODE	27:24	0111b	**LED3/LINK1000# Mode**. This field specifies the control source for the LED3 output. An initial value of 0111b selects LINK1000# indication.
Reserved	29:28	00b	**Reserved**. Read-only as 0b. Write as 0b for future compatibility.
LED3_IVRT	30	0b	**LED3/LINK1000# Invert**.
LED3_BLINK	31	0b	**LED3/LINK1000# Blink.**

1. These bits are read from the EEPROM.

13.4.15.1 MODE Encodings for LED Outputs[1]

The Table 13-61 lists the MODE encodings used to select the desired LED signal source for each LED output. Refer to Section 10.1.1 to ensure proper understanding of expression polarity and resulting LED output polarity.

Note: All 16 modes listed are functional.

1. Not applicable to the **82544GC/EI**.

Table 13-61. Mode Encodings for LED Outputs[1]

Mode	Pneumonic	State / Event Indicated
0000b	LINK_10/1000	Asserted when either 10 or 1000 Mbps link is established and maintained.
0001b	LINK_100/1000	Asserted when either 100 or 1000 Mbps link is established and maintained.
0010b	LINK_UP	Asserted when any speed link is established and maintained.
0011b	ACTIVITY	Asserted when link is established and packets are being transmitted or receive activity that passes filtering.
0100b	LINK/ACTIVITY	Asserted when link is established and when there is no transmit or receive activity that passes filtering.
0101b	LINK_10	Asserted when a 10 Mbps link is established and maintained.
0110b	LINK_100	Asserted when a 100 Mbps link is established and maintained.
0111b	LINK_1000	Asserted when a 1000 Mbps link is established and maintained.
1000b	PCIX_MODE	Asserted when Ethernet controller is in PCI-X mode (deasserted in PCI mode).
1001b	FULL_DUPLEX	Asserted when the link is configured for full duplex operation (deasserted in half-duplex).
1010b	COLLISION	Asserted when a collision is observed.
1011b	BUS_SPEED	Asserted when the Ethernet controller is operating in a PCI 66 MHz or a PCI-X 133 MHz configuration (high-speed PCI operation), deasserted for 33 MHz PCI and 66 MHz PCI-X (as determined by pins sampled at PCI reset).
1100b	BUS_SIZE Reserved for the **82547GI/EI** only)	Asserted when the Ethernet controller is operating as a 64-bit PCI or PCI-X device, deasserted for 32-bit configuration.
1101b	PAUSED	Asserted when the Ethernet controller's transmitter is flow controlled.
1110b	VCC/LED_ON	Always high. Assuming no optional inversion selected, causes output pin high / LED ON for typical LED circuit.

1. Not applicable to the **82544GC/EI**.

Mode	Pneumonic	State / Event Indicated
1111b	GND/LED_OFF	Always low. Assuming no optional inversion selected, causes output pin low / LED OFF for typical LED circuit.

13.4.16 Packet Buffer Allocation

PBA (01000H; R/W)

This register sets the on-chip receive and transmit storage allocation ratio. The receive allocation value is read/write for the lower seven bits. The receive allocation value must be a multiple of eight (multiple of two for the **82547GI/EI** B1 stepping). The transmit allocation is read-only and is calculated based on PBA.RXA.

Table 13-62. PBA Register Bit Description

31 ... 16	15 ... 0
TXA	RXA

Field	Bit(s)	Initial Value	Description
RXA	6:0 15:0 (**82541xx** and **82547GI/EI**)	0030h[1] 0016h (**82547EI** A0-B0 steppings) 001Eh (**82547GI** B1 stepping)	Receive Packet Buffer Allocation in KB Sets the size of the receive packet buffer. The value of this field must be a multiple of eight. The upper nine bits are read only as 0b. Default is 48 KB. For the **82541xx** and **82547GI/EI**, the upper unused bits are read only as 0b. The default is 48 KB for the **82541xx**, 24 KB for the **82547EI**, and 20 KB for the **82547GI**.
RXA_R[2]	15:7	0b	Receive Packet Buffer Allocation – Upper Bits Provides the upper nine bits of the receive packet buffer allocation. Read only bits - Read as 0b.
TXA	31:16 15:00 (**82541xx** and **82547GI/EI**)	0010h 0012h (**82547EI** A0-B0 steppings) 000Ah (**82547GI** B1 stepping)	Transmit Packet Buffer Allocation Provides the size of the transmit packet buffer. The value is in units of KB. These bits are read only. TXA is calculated based on RXA value: TXA = 64 – RXA. For the **82547GI/EI**, TXA is calculated based on RXA value: TXA = 40 – RXA.

1. Not applicable to the **82547GI/EI**.
2. Not applicable to the **82541xx** or **82547GI/EI**.

13.4.17 Interrupt Cause Read Register

ICR (000C0H; R)

This register contains all interrupt conditions for the Ethernet controller. Each time an interrupt causing event occurs, the corresponding interrupt bit is set in this register. A PCI interrupt is generated each time one of the bits in this register is set, and the corresponding interrupt is enabled through the Interrupt Mask Set/Read IMS Register (see Section 13.4.20).

All register bits are cleared upon read. As a result, reading this register implicitly acknowledges any pending interrupt events. Writing a 1b to any bit in the register also clears that bit. Writing a 0b to any bit has no effect on that bit.

Table 13-63. ICR Register Bit Description

31 - 17	16 14 ... 0
Reserved	ICR Bits

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Transmit Descriptor Written Back Set when hardware processes a transmit descriptor with the RS bit set (and possibly IDE set). If using delayed interrupts (IDE set), the interrupt occurs after the timer expires.
TXQE	1	0b	Transmit Queue Empty Set when the last descriptor block for a transmit queue has been used.
LSC	2	0b	Link Status Change This bit is set each time the link status changes (either from up to down, or from down to up). This bit is affected by the internal link indication when configured for internal PHY mode.
RXSEQ	3	0b	Receive Sequence Error In TBI mode/internal SerDes[1], incoming packets with a bad delimiter sequence set this bit. In other 802.3 implementations, this would be classified as a framing error. A valid sequence consists of: idle → SOF → data → pad (opt) EOF → fill (opt) → idle. This is a reserved bit for the **82541xx**, **82547GI/EI**, and **82540EP/EM**. Set to 0b.
RXDMT0	4	0b	Receive Descriptor Minimum Threshold Reached Indicates that the minimum number of receive descriptors are available and software should load more receive descriptors.
Reserved	5	0b	Reserved Reads as 0b.
RXO	6	0b	Receiver Overrun Set on receive data FIFO overrun. Could be caused either because there are no available receive buffers or because PCI receive bandwidth is inadequate.

Field	Bit(s)	Initial Value	Description
RXT0	7	0b	Receiver Timer Interrupt Set when the receiver timer expires. The receiver timer is used for receiver descriptor packing. Timer expiration flushes any accumulated descriptors and sets an interrupt event when enabled.
Reserved	8	0b	Reserved Reads as 0b.
MDAC	9	0b	MDI/O Access Complete This bit is set when the MDI/O access is completed.
RXCFG	10	0b	Receiving /C/ ordered sets Mapped to RXCW.RxConfig. Sets when the hardware receives configuration symbols (/C/ codes). Software should enable this interrupt when forcing link. When the link is forced, the link partner can begin to Auto-Negotiate based, due to a reset or enabling of Auto-Negotiation. The reception of /C/ codes causes an interrupt to software and the proper hardware configuration might be set. See Section 13.4.14 for details. Only valid in internal SerDes mode (TBI mode for the **82544GC/EI**). This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
Reserved	11	0b	Reserved. Set this bit to 0b. Not applicable to the **82544GC/EI**.
PHYINT	12	0b	PHY Interrupt (not applicable to the **82544GC/EI**) Set when the PHY generates an interrupt. If bit 1 (PHYINT_EN) of the CTRL_EXT register (00018h) is set, then this bit gets set. This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
GPI_SDP6 GPI_SDP2 (**82541xx** and **82547GI/EI**)	13	0b	General Purpose Interrupt on SDP6[2]. If GPI interrupt detection is enabled on this pin (via CTRL_EXT), this interrupt cause is set when the SDP6[2] is sampled high. Not applicable to the **82544GC/EI**.
GPI_SDP7 GPI_SDP3 (**82541xx** and **82547GI/EI**)	14	0b	General Purpose Interrupt on SDP7[3]. If GPI interrupt detection is enabled on this pin (via CTRL_EXT), this interrupt cause is set when the SDP7[3] is sampled high. Not applicable to the **82544GC/EI**.
GPI	14:13, 11	0b	General Purpose Interrupts (**82544GC/EI** only) These bits are mapped to the upper three SDP pins when they are configured as inputs. Refer to Section 13.4.6.
TXD_LOW[2]	15	0b	Transmit Descriptor Low Threshold hit. Indicates that the descriptor ring has reached the threshold specified in the Transmit Descriptor Control register.
SRPD[2]	16	0b	Small Receive Packet Detected. Indicates that a packet of size ≤ RSRPD.SIZE register has been detected and transferred to host memory. The interrupt is only asserted if RSRPD.SIZE register has a non-zero value.

Field	Bit(s)	Initial Value	Description
Reserved	31:17	0b	Reserved Reads as 0b.

1. The **82540EP/EM**, **82541xx**, or **82547GI/EI** do not support SerDes functionality.
2. Not applicable to the **82544GC/EI**.

Note: The **82547GI/EI** signals interrupts over the CSA port, not a dedicated interrupt pin.

13.4.18 Interrupt Throttling Register[1]

ITR (000C4h; R/W)

31 - 16	15 - 0
Reserved	INTERVAL

Field	Bit(s)	Initial Value	Description
INTERVAL	15:0	0b	Minimum inter-interrupt interval. The interval is specified in 256 ns increments. Setting this bit to 0b disables interrupt throttling logic.
Reserved	31:16	X	Reserved. Should be written with 0b to ensure future compatibility.

Software can use this register to pace (or even out) the delivery of interrupts to the host CPU. This register provides a guaranteed inter-interrupt delay between interrupts asserted by the Ethernet controller, regardless of network traffic conditions. To independently validate configuration settings, software can use the following algorithm to convert the inter-interrupt interval value to the common interrupts/sec performance metric:

$$interrupts/\sec ond = (256 \times 10^{-9} \times interval)^{-1}$$

For example, if the interval is programmed to 500d, the Ethernet controller guarantees the CPU is not interrupted by the Ethernet controller for 128 µsec from the last interrupt. The maximum observable interrupt rate from the Ethernet controller must never exceed 7813 interrupts/sec.

Inversely, inter-interrupt interval value can be calculated as:

$$inter\text{-}interrupt\ interval = (256 \times 10^{-9} \times interrupts/\sec)^{-1}$$

The optimal performance setting for this register is very system and configuration specific. A initial suggested range is 651-5580 (28Bh - 15CCh).

1. Not applicable to the **82544GC/EI**.

13.4.19 Interrupt Cause Set Register

ICS (000C8h; W)

Software uses this register to set an interrupt condition. Any bit written with a 1b sets the corresponding interrupt. This results in the corresponding bit being set in the Interrupt Cause Read Register (see Section 13.4.17). A PCI interrupt is generated if one of the bits in this register is set and the corresponding interrupt is enabled through the Interrupt Mask Set/Read Register (see Section 13.4.20).

Bits written with 0b are unchanged.

Table 13-64. ICS Register Bit Description

31 – 17	16 – 0
Reserved	ICS Bits

Field	Bit(s)	Initial Value	Description
TXDW	0	X	Sets Transmit Descriptor Written Back Interrupt.
TXQE	1	X	Sets Transmit Queue Empty Interrupt.
LSC	2	X	Sets Link Status Change Interrupt.
RXSEQ	3	X	Sets Receive Sequence Error Interrupt. This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
RXDMT0	4	X	Sets Receive Descriptor Minimum Threshold Reached Interrupt.
Reserved	5	X	Reserved Should be written with 0b to ensure future compatibility.
RXO	6	X	Sets on Receiver FIFO Overrun Interrupt.
RXT0	7	X	Sets Receiver Timer Interrupt.
Reserved	8	X	Reserved Should be written with 0b to ensure future compatibility.
MDAC	9	X	Sets MDI/O Access Complete Interrupt.
RXCFG	10	X	Sets Receiving /C/ ordered sets Interrupt. This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
Reserved	11	X	Reserved Should be written with 0b to ensure future compatibility. Not applicable to the **82544GC/EI**.
PHYINT	12	X	Sets PHY interrupt. Not applicable to the **82544GC/EI**. This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.

Field	Bit(s)	Initial Value	Description
GPI	14:11	X	Sets General Purpose Interrupts (**82544GC/EI** only).
GPI	14:13	X	Sets General Purpose Interrupts.
TXD_LOW	15	X	Transmit Descriptor Low Threshold Hit. Not applicable to the **82544GC/EI**.
SRPD	16	X	Small Receive Packet Detected and Transferred. Not applicable to the **82544GC/EI**.
Reserved	31:17	X	Reserved Should be written with 0b to ensure future compatibility.

13.4.20 Interrupt Mask Set/Read Register

IMS (000D0h; R/W)

An interrupt is enabled if its corresponding mask bit is set to 1b, and disabled if its corresponding mask bit is set to 0b. A PCI interrupt is generated each time one of the bits in this register is set and the corresponding interrupt condition occurs. The occurrence of an interrupt condition is reflected by having a bit set in the Interrupt Cause Read Register (see Section 13.4.17).

A particular interrupt can be enabled by writing a 1b to the corresponding mask bit in this register. Any bits written with a 0b are unchanged. As a result, if software desires to disable a particular interrupt condition that had been previously enabled, it must write to the Interrupt Mask Clear Register (see Section 13.4.21) rather than writing a 0b to a bit in this register.

Reading this register returns bits that have an interrupt mask set.

Note: For the **82547GI/EI**, programmers need to first write (clear) the IMS and IMC registers due to a Hub Link bus being occupied. This results in an interrupt de-assertion message that can't to be sent out. When a future interrupt assertion message is generated, two messages are re-ordered and sent out. This signals APIC that the **82547GI/EI** is in a de-asserted state when it is actually in an asserted state, which causes a system dead lock. To avoid a system dead lock, first clear the IMS and IMC registers by writing FFFFh and then re-assert IRQ enable.

Table 13-65. IMS Register Bit Description

31 ... 17	16 ... 0
Reserved	IMS Bits

Field	Bit(s)	Initial Value	Description
TXDW	0	X	Sets mask for Transmit Descriptor Written Back.
TXQE	1	X	Sets mask for Transmit Queue Empty.
LSC	2	X	Sets mask for Link Status Change.

Field	Bit(s)	Initial Value	Description
RXSEQ	3	X	Sets mask for Receive Sequence Error. This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
RXDMT0	4	X	Sets mask for Receive Descriptor Minimum Threshold hit.
Reserved	5	X	Reserved Should be written with 0b to ensure future compatibility.
RXO	6	X	Sets mask for on Receiver FIFO Overrun.
RXT0	7	X	Sets mask for Receiver Timer Interrupt.
Reserved	8	X	Reserved Should be written with 0b to ensure future compatibility.
MDAC	9	X	Sets mask for MDI/O Access Complete Interrupt.
RXCFG	10	X	Sets mask for Receiving /C/ ordered sets. This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
Reserved	11	X	Reserved Should be written with 0b to ensure future compatibility (not applicable to the **82544GC/EI**).
PHYINT	12	X	Sets mask for PHY Interrupt (not applicable to the **82544GC/EI**). This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
GPI	14:11	X	Sets mask for General Purpose Interrupts (**82544GC/EI** only).
GPI	14:13	X	Sets mask for General Purpose Interrupts.
TXD_LOW	15	X	Sets the mask for Transmit Descriptor Low Threshold hit (not applicable to the **82544GC/EI**).
SRPD	16	X	Sets mask for Small Receive Packet Detection (not applicable to the **82544GC/EI**).
Reserved	31:17	0b	Reserved Should be written with 0b to ensure future compatibility.

13.4.21 Interrupt Mask Clear Register

IMC (000D8h; W)

Software uses this register to disable an interrupt. Interrupts are presented to the bus interface only when the mask bit is set to 1b and the cause bit set to 1b. The status of the mask bit is reflected in the Interrupt Mask Set/Read Register (see Section 13.4.20), and the status of the cause bit is reflected in the Interrupt Cause Read Register (see Section 13.4.17).

Software blocks interrupts by clearing the corresponding mask bit. This is accomplished by writing a 1b to the corresponding bit in this register. Bits written with 0b are unchanged (their mask status does not change).

Software should write a 1b to the reserved bits to ensure future compatibility. Since this register masks interrupts when 1b is written to the corresponding (defined) bits, then writing 1b to the reserved bits ensures that the software is never called to handle an interrupt that the software is not aware exists.

Note: For the **82547GI/EI**, programmers need to first write (clear) the IMS and IMC registers due to a Hub Link bus being occupied. This results in an interrupt de-assertion message that can't to be sent out. When a future interrupt assertion message is generated, two messages are re-ordered and sent out. This signals APIC that the **82547GI/EI** is in a de-asserted state when it is actually in an asserted state, which causes a system dead lock. To avoid a system dead lock, first clear the IMS and IMC registers by writing FFFFh and then re-assert IRQ enable.

Table 13-66. IMC Register Bit Description

31 17	16 0
Reserved	IMC Bits

Field	Bit(s)	Initial Value	Description
TXDW	0	X	Clears mask for Transmit Descriptor Written Back.
TXQE	1	X	Clears mask for Transmit Queue Empty.
LSC	2	X	Clears mask for Link Status Change.
RXSEQ	3	X	Clears mask for Receive Sequence Error. This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
RXDMT0	4	X	Sets mask for Receive Descriptor Minimum Threshold hit.
Reserved	5	X	Reserved: Should be written with 1b to ensure future compatibility.
RXO	6	X	Clears mask for on Receiver FIFO Overrun.
RXT0	7	X	Clears mask for Receiver Timer Interrupt.
Reserved	8	X	Reserved: Should be written with 1b to ensure future compatibility.
MDAC	9	X	Clears mask for MDI/O Access Complete Interrupt.
RXCFG	10	X	Clears mask for Receiving /C/ ordered sets. This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
Reserved	11	X	Reserved: Should be written with 1b to ensure future compatibility (not applicable to the **82544GC/EI**).
PHYINT	12	X	Clears PHY Interrupts (not applicable to the **82544GC/EI**). This is a reserved bit for the **82541xx** and **82547GI/EI**. Set to 0b.
GPI	14:11	X	Clears General Purpose Interrupts (**82544GC/EI** only).
GPI	14:13	X	Clears General Purpose Interrupts.

Field	Bit(s)	Initial Value	Description
TXD_LOW	15	X	Clears the mask for Transmit Descriptor Low Threshold hit (not applicable to the **82544GC/EI**).
SRPD	16	X	Clears mask for Small Receive Packet Detect Interrupt (not applicable to the **82544GC/EI**).
Reserved	31:17	X	Reserved Should be written with 1b to ensure future compatibility.

13.4.22 Receive Control Register

RCTL (00100h; R/W)

This register controls all Ethernet controller receiver functions.

Table 13-67. RCTL Register Bit Description

31 27	26 0
Reserved	Receive Control Bits

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved Write to 0b for future compatibility.
EN	1	0b	Receiver Enable The receiver is enabled when this bit is 1b. Writing this bit to 0b stops reception after receipt of any in-progress packets. Data remains in the receive FIFO until the device is re–enabled. Disabling or re-enabling the receiver does not reinitialize the packet filter logic that demarcates packet start and end locations in the FIFO; Therefore the receiver must be reset before re-enabling it.
SBP	2	0b	Store Bad Packets 0b = do not store. 1b = store bad packets. When set, the Ethernet controller stores bad packets (CRC error, symbol error, sequence error, length error, alignment error, short packets or where carrier extension or RX_ERR errors) that pass the filter function in host memory. When the Ethernet controller is in promiscuous mode, and SBP is set, it might possibly store all packets.
UPE	3	0b	Unicast Promiscuous Enabled 0b = Disabled. 1b = Enabled. When set, passes without filtering out all received unicast packets. Otherwise, the Ethernet controller accepts or rejects unicast packets based on the received packet destination address match with 1 of the 16 stored addresses.

Field	Bit(s)	Initial Value	Description
MPE	4	0b	Multicast Promiscuous Enabled 0b = Disabled. 1b = Enabled. When set, passes without filtering out all received multicast packets. Otherwise, the Ethernet controller accepts or rejects a multicast packet based on its 4096-bit vector multicast filtering table.
LPE	5	0b	Long Packet Reception Enable 0b = Disabled. 1b = Enabled. LPE controls whether long packet reception is permitted. When LPE is cleared, the Ethernet controller discards packets longer than 1522 bytes. When LPE is set, the Ethernet controller discards packets that are longer than 16384 bytes. For the **82541xx** and **82547GI/EI**, packets larger than 2 KB require full duplex operation.
LBM	7:6	0b	Loopback mode. Controls the loopback mode of the Ethernet controller. 00b = No loopback. 01b = Undefined. 10b = Undefined. 11b = PHY or external SerDes loopback. All loopback modes are only allowed when the Ethernet controller is configured for full-duplex operation. Receive data from transmit data looped back internally to the SerDes or internal PHY. In TBI mode (**82544GC/EI**), the EWRAP signal is asserted. **Note:** The **82540EP/EM**, **82541xx**, and **82547GI/EI** do not support SerDes functionality.
RDMTS	9:8	0b	Receive Descriptor Minimum Threshold Size The corresponding interrupt ICR.RXDMT0 is set each time the fractional number of free descriptors becomes equal to RDMTS. The following table lists which fractional values correspond to RDMTS values. The size of the total receiver circular descriptor buffer is set by RDLEN. See Section 13.4.27 for details regarding RDLEN. 00b = Free Buffer threshold is set to 1/2 of RDLEN. 01b = Free Buffer threshold is set to 1/4 of RDLEN. 10b = Free Buffer threshold is set to 1/8 of RDLEN. 11b = Reserved.
Reserved	11:10	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.
MO	13:12	0b	Multicast Offset The Ethernet controller is capable of filtering multicast packets based on 4096-bit vector multicast filtering table. The MO determines which bits of the incoming multicast address are used in looking up the 4096-bit vector. 00b = bits [47:36] of received destination multicast address. 01b = bits [46:35] of received destination multicast address. 10b = bits [45:34] of received destination multicast address. 11b = bits [43:32] of received destination multicast address.
Reserved	14	0b	Reserved Should be written with 0 to ensure future compatibility Reads as 0

Field	Bit(s)	Initial Value	Description
BAM	15	0b	Broadcast Accept Mode. 0 = ignore broadcast; 1 = accept broadcast packets. When set, passes and does not filter out all received broadcast packets. Otherwise, the Ethernet controller accepts, or rejects a broadcast packet only if it matches through perfect or imperfect filters.
BSIZE	17:16	0b	Receive Buffer Size Controls the size of the receive buffers, allowing the software to trade off between system performance and storage space. Small buffers maximize memory efficiency at the cost of multiple descriptors for bigger packets. RCTL.BSEX = 0b: 00b = 2048 Bytes. 01b = 1024 Bytes. 10b = 512 Bytes. 1b1 = 256 Bytes. RCTL.BSEX = 1b: 00b = Reserved; software should not program this value. 01b = 16384 Bytes. 10b = 8192 Bytes. 11b = 4096 Bytes.
VFE	18	0b	VLAN Filter Enable[1] 0b = Disabled (filter table does not decide packet acceptance). 1b = Enabled (filter table decides packet acceptance for 802.1Q packets). Three bits control the VLAN filter table. RCTL.VFE determines whether the VLAN filter table participates in the packet acceptance criteria. RCTL.CFIEN and RCTL.CFI are used to decide whether the CFI bit found in the 802.1Q packet's tag should be used as part of the acceptance criteria.
CFIEN	19	0b	Canonical Form Indicator Enable 0b = Disabled (CFI bit found in received 802.1Q packet's tag is not compared to decide packet acceptance). 1b = Enabled (CFI bit found in received 802.1Q packet's tag must match RCTL.CFI to accept 802.1Q type packet.
CFI	20	0b	Canonical Form Indicator bit value If RCTL.CFIEN is set, then 802.1Q packets with CFI equal to this field is accepted; otherwise, the 802.1Q packet is discarded.
Reserved	21	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.
DPF	22	0b	Discard Pause Frames 0 = incoming pause frames subject to filter comparison. 1 = incoming pause frames are filtered out even if they match filter registers. DPF controls the DMA function of flow control PAUSE packets addressed to the station address (RAH/L[0]). If a packet is a valid flow control packet and is addressed to the station's address, it is not transferred to host memory if RCTL.DPF = 1b. However, it is transferred when DPF is set to 0b.

Field	Bit(s)	Initial Value	Description
PMCF	23	0b	Pass MAC Control Frames 0b = Do not (specially) pass MAC control frames. 1b = Pass any MAC control frame (type field value of 8808h) that does not contain the pause opcode of 0001h. PMCF controls the DMA function of MAC control frames (other than flow control). A MAC control frame in this context must be addressed to either the MAC control frame multicast address or the station address, match the type field and NOT match the PAUSE opcode of 0001h. If PMCF = 1b then frames meeting this criteria are transferred to host memory. Otherwise, they are filtered out.
Reserved	24	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.
BSEX	25	0b	Buffer Size Extension When set to one, the original BSIZE values are multiplied by 16. Refer to the RCTL.BSIZE bit description.
SECRC	26	0b	Strip Ethernet CRC from incoming packet 0b = Do not strip CRC field. 1b = Strip CRC field. Controls whether the hardware strips the Ethernet CRC from the received packet. This stripping occurs prior to any checksum calculations. The stripped CRC is not transferred to host memory and is not included in the length reported in the descriptor.
Reserved	31:27	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

1. Not applicable to the **82541ER**.

13.4.23 Flow Control Receive Threshold Low

FCRTL (02160h; R/W)

This register contains the receive threshold used to determine when to send an XON packet. It counts in units of bytes. Each time the receive FIFO crosses the receive high threshold FCRTH.RTH (filling up), and then crosses the receive low threshold FCRTL.RTL, with FCRTL.XONE enabled, hardware transmits an XON frame.

Flow control reception/transmission are negotiated capabilities by the Auto-Negotiation process. When the Ethernet controller is manually configured, flow control operation is determined by the CTRL.RFCE and CTRL.TFCE bits.

Table 13-68. FCRTL Register Bit Description

31	30 … 16	15 … 3	2 … 0
XONE[1]	Reserved	RTL	0

1. **82544GC/EI**, **82541xx**, and **82547GI/EI** only.

Field	Bit(s)	Initial Value	Description
Reserved	2:0	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.
RTL	15:3	0b	Receive Threshold Low. FIFO low water mark for flow control transmission. Each time the receive FIFO crosses the receive high threshold FCRTH.RTH and later crosses the receive low threshold FCRTL.RTL with FCRTL.XONE enabled, hardware transmits an XON frame (a PAUSE frame with a timer value of 0b). RTL is provided in units of 8 bytes.
Reserved	31:16	0b	Reserved Should be written with 0b for future compatibility. Reads as 0b.
XONE	31	0b	XON Enable (**82544GC/EI**, **82541xx**, and **82547GI/EI** only) 0b = Disabled. 1b = Enabled. When set, enables the Ethernet controller to transmit XON packets based on receive FIFO crosses FCRTL.RTL threshold value, or based on external pins XOFF and XON.

13.4.24 Flow Control Receive Threshold High

FCRTH (02168h; R/W)

This register contains the receive threshold used to determine when to send an XOFF packet. It counts in units of bytes. Each time the receive FIFO reaches the fullness indicated by FCRTH, hardware transmits a PAUSE frame if the transmission of flow control frames is enabled (CTRL.TFCE).

Flow control reception/transmission are negotiated capabilities by the Auto-Negotiation process. When the Ethernet controller is manually configured, flow control operation is determined by the CTRL.RFCE & CTRL.TFCE bits.

Table 13-69. FCRTH Register Bit Description

31	30 16	15 3	2 0
XFCE[1]	Reserved	RTH	0

1. **82544GC/EI** only.

Field	Bit(s)	Initial Value	Description
Reserved	2:0	0b	Reserved Should be written with 0 for future compatibility Reads as 0
RTH	15:3	0b	Receive Threshold High. FIFO high water mark for flow control transmission. Each time the receive FIFO reaches the fullness indicated by RTH, the Ethernet controller transmits a Pause packet if enabled to do so.
Reserved	31:16	0b	Reserved Should be written with 0b for future compatibility. Reads as 0b.
XFCE	31	0	External Flow Control Enabled (**82544GC/EI** only) 0b = Disabled. 1b = Enabled. Allows the Ethernet controller to send XOFF and XON frames based on external pins XOFF and XON. The transmission of pause frames must be also enabled through the CTRL.TFCE control bit. When the XOFF signal is asserted high, the Ethernet controller transmits a single XOFF frame. The assertion of XON (after deassertion of XOFF) initiates an XON frame transmission, if enabled by FCRTL.XONE. The assertion/deassertion of XON is required between assertions of XOFF in order to send another XOFF frame. This behavior also provides a built-in hysteresis mechanism.

13.4.25 Receive Descriptor Base Address Low

RDBAL (02800h;R/W)

This register contains the lower bits of the 64-bit descriptor base address. The four low-order register bits are always ignored. The Receive Descriptor Base Address must point to a 16-byte aligned block of data.

Table 13-70. RDBAL Register Bit Description

31 ... 4	3 ... 0
RDBAL	0

Field	Bit(s)	Initial Value	Description
0	3:0	0b	Ignored on writes. Returns 0b on reads.
RDBAL	31:4	X	Receive Descriptor Base Address Low.

13.4.26 Receive Descriptor Base Address High

RDBAH (02804h; R/W)

This register contains the upper 32 bits of the 64-bit Descriptor Base Address.

Table 13-71. RDBAH Register Bit Description

31 ... 0
RDBAH

Field	Bit(s)	Initial Value	Description
RDBAH	31:0	X	Receive Descriptor Base Address [63:32]

13.4.27 Receive Descriptor Length

RDLEN (02808h; R/W)

This register determines the number of bytes allocated to the circular receive descriptor buffer. This value must be 128-byte aligned (the maximum cache line size). Since each descriptor is 16 bytes in length, the total number of receive descriptors is always a multiple of eight.

Table 13-72. RDLEN Register Bit Description

31 20	19 7	6 0
Reserved	LEN	0

Field	Bit(s)	Initial Value	Description
Zero	6:0	0b	Zero value Ignore on write. Reads back as 0b.
LEN	19:7	0b	Receive Descriptor length Provides the number of receive descriptors (in a multiple of eight).
Reserved	31:20	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

13.4.28 Receive Descriptor Head

RDH (02810h; R/W)

This register contains the head pointer for the receive descriptor buffer. The register points to a 16-byte datum. Hardware controls the pointer. The only time that software should write to this register is after a reset (RCTL.RST or CTRL.RST) and before enabling the receiver function (RCTL.EN). If software were to write to this register while the receive function was enabled, the on-chip descriptor buffers can be invalidated and other indeterminate operations might result. Reading the descriptor head to determine which buffers are finished is not reliable.

Table 13-73. RDH Register Bit Description

31 16	15 0
Reserved	RDH

Field	Bit(s)	Initial Value	Description
RDH	15:0	0b	Receive Descriptor Head.
Reserved	31:16	0b	Reserved. Should be written with 0b for future compatibility. Reads as 0b.

13.4.29 Receive Descriptor Tail

RDT (02818h;R/W)

This register contains the tail pointers for the receive descriptor buffer. The register points to a 16-byte datum. Software writes the tail register to add receive descriptors to the hardware free list for the ring.

Table 13-74. RDT Register Bit Description

31 ... 16	15 ... 0
Reserved	RDT

Field	Bit(s)	Initial Value	Description
RDT	15:0	0b	Receive Descriptor Tail.
Reserved	31:16	0b	Reserved Reads as 0b. Should be written with 0b for future compatibility.

13.4.30 Receive Delay Timer Register

RDTR (02820h; R/W)

This register is used to delay interrupt notification for the receive descriptor ring. Delaying interrupt notification helps maximize the number of receive packets serviced by a single interrupt.

Warning: It is strongly recommended that the *Delay Timer* field of this register not be used. For any application requiring an interrupt moderation mechanism, it is recommended that the Interrupt Throttling Register (ITR) be used instead. ITR provides a more direct interrupt solution than RDTR. In addition, Intel software device drivers use ITR instead of RDTR. Refer to the *82546EB Gigabit Ethernet Controller Specification Update* or *82546GB Gigabit Ethernet Controller Specification Update* for additional details.

Table 13-75. RDTR Register Bit Description

31	30 ... 16	15 ... 0
FPD	Reserved	Delay Timer

Field	Bit(s)	Initial Value	Description
Delay Timer	15:0	0b	Receive delay timer measured in increments of 1.024 μs.
Reserved	30:16	0b	Reserved. Reads as 0b.
FPD	31	0b	Flush partial descriptor block when set to 1b; ignore otherwise. Reads 0b (self-clearing).

This feature operates by initiating a countdown timer upon successfully receiving each packet to system memory. If a subsequent packet is received BEFORE the timer expires, the timer is re-initialized to the programmed value and re-starts its countdown. If the timer expires due to NOT having received a subsequent packet within the programmed interval, pending receive descriptor writebacks are flushed and a receive timer interrupt is generated.

Setting the value to 0b represents no delay from a receive packet to the interrupt notification, and results in immediate interrupt notification for each received packet.

Writing this register with FPD set initiates an immediate expiration of the timer, causing a writeback of any consumed receive descriptors pending writeback, and results in a receive timer interrupt in the ICR.

Receive interrupts due to a Receive Absolute Timer (RADV) expiration cancels a pending RDTR interrupt. The RDTR countdown timer is reloaded but halted, so as to avoid generation of a spurious second interrupt after the RADV has been noted, but might be restarted by a subsequent received packet.

13.4.31 Receive Interrupt Absolute Delay Timer[1]

RADV (0282Ch; RW)

Warning: It is strongly recommended that the *Delay Timer* field of this register not be used. For any application requiring an interrupt moderation mechanism, it is recommended that the Interrupt Throttling Register (ITR) be used instead. ITR provides a more direct interrupt solution than RADV. In addition, Intel software device drivers use ITR instead of RADV. Refer to the *82546EB Gigabit Ethernet Controller Specification Update* or *82546GB Gigabit Ethernet Controller Specification Update* for additional details.

31 30 16	15 0
Reserved	Delay Timer

Field	Bit(s)	Initial Value	Description
Delay Timer	15:0	0b	Receive absolute delay timer measured in increments of 1.024 µs (0b = disabled).
Reserved	31:16	0b	Reserved. Reads as 0b.

If the packet delay timer is used to coalesce receive interrupts, the Ethernet controller ensures that when receive traffic abates, an interrupt is generated within a specified interval of no receives. During times when receive traffic is continuous, it may be necessary to ensure that no receive remains unnoticed for too long an interval. This register can be used to ENSURE that a receive interrupt occurs at some predefined interval after the first packet is received.

1. Not applicable to the **82544GC/EI**.

When this timer is enabled, a separate absolute countdown timer is initiated upon successfully receiving each packet to system memory. When this absolute timer expires, pending receive descriptor writebacks are flushed and a receive timer interrupt is generated.

Setting this register to 0b disables the absolute timer mechanism (the RDTR register should be used with a value of 0b to cause immediate interrupts for all receive packets).

Receive interrupts due to a Receive Packet Timer (RDTR) expiration cancels a pending RADV interrupt. If enabled, the RADV countdown timer is reloaded but halted, so as to avoid generation of a spurious second interrupt after the RDTR has been noted.

13.4.32 Receive Small Packet Detect Interrupt[1]

RSRPD (02C00h; R/W)

31 12	11 0
Reserved	SIZE

Field	Bit(s)	Initial Value	Description
SIZE	11:0	0b	If the interrupt is enabled, any receive packet of size ≤ SIZE asserts an Interrupt. SIZE is specified in bytes and includes the headers and the CRC. It does not include the VLAN header in size calculation if it is stripped.
Reserved	31:12	X	Reserved. Reads as 0b.

13.4.33 Transmit Control Register

TCTL (00400h;R/W)

This register controls all transmit functions for the Ethernet controller.

1. Not applicable to the **82544GC/EI**.

Table 13-76. TCTL Register Bit Description

31 … 26	25 … 22	21 … 12	11 … 4	3 … 0
Reserved	CNTL Bits	COLD	CT	CNTL Bits

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved Write as 0b for future compatibility.
EN	1	0b	Transmit Enable The transmitter is enabled when this bit is set to 1b. Writing 0b to this bit stops transmission after any in progress packets are sent. Data remains in the transmit FIFO until the device is re-enabled. Software should combine this operation with reset if the packets in the TX FIFO should be flushed.
Reserved	2	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.
PSP	3	0b	Pad Short Packets 0b = Do not pad. 1b = Pad short packets. Padding makes the packet 64 bytes long. The padding content is data. When the Pad Short Packet feature is disabled, the minimum packet size the Ethernet controller can transfer to the host is 32 bytes long. This feature is not the same as Minimum Collision Distance (TCTL.COLD).
CT	11:4	0b	Collision Threshold This determines the number of attempts at re-transmission prior to giving up on the packet. The Ethernet back–off algorithm is implemented and clamps to the maximum value after 16 retries. It only has meaning in half-duplex operation. Recommended value – 0Fh.
COLD	21:12	0b	Collision Distance Specifies the minimum number of byte times that must elapse for proper CSMA/CD operation. Packets are padded with special symbols, not valid data bytes. Hardware checks this value and padded packets even in full-duplex operation. Recommended value: Half-Duplex – 512-byte time (200h) Full-Duplex – 64-byte time (40h) **Note:** 10/100 half-duplex - 64 - 68 (40h to 44h) byte times for the **82541xx** and **82547GI/EI** only.
SWXOFF	22	0b	Software XOFF Transmission When set to 1b, the Ethernet controller schedules the transmission of an XOFF (PAUSE) frame using the current value of the PAUSE timer (FCTTV.TTV). This bit self-clears upon transmission of the XOFF frame. This bit is valid only in Full-Duplex mode of operation. Software should not set this bit while the Ethernet controller is configured for half-duplex operation.

Field	Bit(s)	Initial Value	Description
Reserved	23	0b	Reserved Read as 0b. Should be written with 0b for future compatibility.
RTLC	24	0b	Re-transmit on Late Collision When set, enables the Ethernet controller to re-transmit on a late collision event. The collision window is speed dependent. For example, 64 bytes for 10/100 Mb/s and 512 bytes for 1000Mb/s operation. If a late collision is detected when this bit is disabled, the transmit function assumes the packet is successfully transmitted. The RTLC bit is ignored in full-duplex mode.
NRTU[1] Reserved	25	0b	No Re-transmit on underrun (**82544GC/EI** only) If this bit is set, the **82544GC/EI** does not re-transmit packets that initially had an underrun. This function is accomplished by waiting for the entire packet to be buffered in the transmit FIFO before the controller attempts to re-transmit a packet that previously encountered an underrun. This operation guarantees only one underrun can occur per packet. This is a reserved bit for all other Ethernet controllers and should be written with 0b for future compatibility.
Reserved	31:26	0b	Reserved Read as 0. Should be written with 0b for future compatibility.

1. **82544GC/EI** only.

For the **82541xx** and **82547GI/EI**, carrier extension (through the $TCTL_{COLD}$ field) provides a method to increase the duration of the carrier event to a minimum usable duration in order to meet a 200 m collision domain objective, even though half-duplex operation is impractical at Gigabit. Packets that are signaled from the CSMA/CD layer that do not meet the minimum slot time of 512 bytes have extension bytes appended to them in order to meet this minimum slot time requirement. The extension bytes are defined within the context of the frame encapsulation discussion of the 802.3z standard and are recognized by 802.3z compliant devices. Refer to Figure 13-1

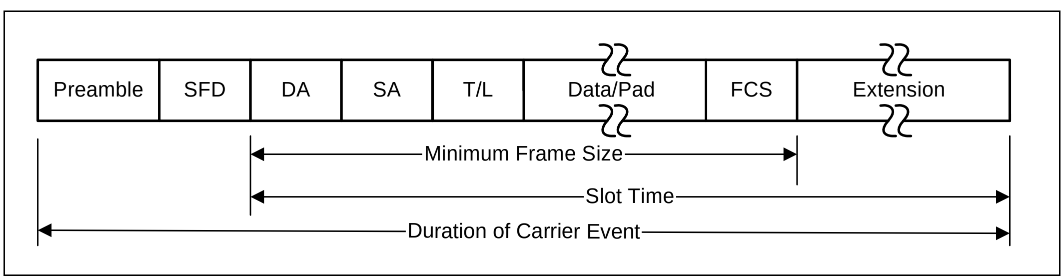


Figure 13-1. Carrier Extended Frame Format (82541xx and 82547GI/EI)

13.4.34 Transmit IPG Register

TIPG (00410;R/W)

This register controls the IPG (Inter Packet Gap) timer for the Ethernet controller.

Table 13-77. TIPG Register Bit Description

31 30	29 20	19 10	9 0
Reserved	IPGR2	IPGR1	IPGT

Field	Bit(s)	Initial Value	Description
IPGT	9:0	X	IPG Transmit Time Specifies the IPG time for back-to-back packet transmissions Measured in increments of the MAC clock: • 8 ns MAC clock when operating @ 1 Gbps. • 80 ns MAC clock when operating @ 100 Mbps. • 800 ns MAC clock when operating @ 10 Mbps. To calculate the actual IPG value for TBI applications, a value of 10 (6 for the **82544GC/EI**) should be added to the IPGT value. For the IEEE 802.3 standard IPG value of 96-bit time, the value that should be programmed into IPGT is 10 (6 for the **82544GC/EI**). To calculate the IPG value for 10/100/1000BASE-T applications, a value of four should be added to the IPGT value as four clocks are used by the MAC as internal overhead. The value that should be programmed into IPGT is 10 (8 for the **82544GC/EI**). These values are recommended to assure that the minimum IPG gap is met under all synchronization conditions.
IPGR1	19:10	X	IPG Receive Time 1 Specifies the length of the first part of the IPG time for non back-to-back transmissions. During this time, the internal IPG counter restarts if any carrier event occurs. Once the time specified in IPGR1 has elapsed, carrier sense does not affect the IPG counter. According to the IEEE802.3 standard, IPGR1 should be 2/3 of IPGR2 value. Measured in increments of the MAC clock: • 8 ns MAC clock when operating @ 1 Gbps (**82544GC/EI** only). • 80 ns MAC clock when operating @ 100 Mbps • 800 ns MAC clock when operating @ 10 Mbps. For IEEE 802.3 minimum IPG value of 96-bit time, the value that should be programmed into IPGR1 is eight. IPGR1 is significant only in half-duplex mode of operation.

Field	Bit(s)	Initial Value	Description
IPGR2	29:20	X	IPG Receive Time 2 Specifies the total length of the IPG time for non back-to-back transmissions. Measured in increments of the MAC clock: • 8 ns MAC clock when operating @ 1 Gbps (**82544GC/EI** only). • 80 ns MAC clock when operating @ 100 Mbps • 800 ns MAC clock when operating @ 10 Mbps. In order to calculate the actual IPG value, a value of six should be added to the IPGR2 value as six MAC clocks are used by the MAC for synchronization and internal engines. For the IEEE 802.3 standard IPG value of 96-bit time, the value that should be programmed into IPGR2 is six (total IPG delay of 12 MAC clock cycles) According to the IEEE802.3 standard, IPGR1 should be 2/3 of IPGR2 value. IPGR2 is significant only in half-duplex mode of operation.
Reserved	31:30	X	Reserved Read as 0b. Should be written with 0b for future compatibility.

13.4.35 Adaptive IFS Throttle - AIT

AIFS (00458;R/W)

This register throttles back-to-back transmissions in the transmit packet buffer and delays their transfer to the CSMA/CD transmit function. As a result, it can be used to delay the transmission of back-to-back packets on the wire.

For the **82544GC/EI**, this register can be used to increase the IPG value between transmitting back-to-back packets on the wire and between frames while in half-duplex mode.

Table 13-78. AIFS Register Bit Description

31 ... 16	15 ... 0
Reserved	Adaptive IFS

Field	Bit(s)	Initial Value	Description
AIFS	15:0	0b	Adaptive IFS Value (**82544GC/EI** only) Adaptive IFS throttles back-to-back transmissions in the transmit packet buffer and delays their transfer to the CSMA/CD transmit function. Normally, this register should be set to 0b. However, if additional delay is desired between back-to-back transmit packets, then this register can be set with a value greater than zero (0). This feature can be helpful in high collision half-duplex environments. In order for AIFS to take effect it should be larger than the minimum IFS value defined in IEEE 802.3 standard. AIFS has no effect on transmissions that occur immediately after receives or transmissions that are not back-to-back. In addition, it has no effect on re-transmission timing (re-transmission after collisions). The AIFS programming value is in units of 8 ns (TX_CLK), and is 16 bits wide, thus providing greater flexibility and maximum delay value comparing TIPG.IPGT. The AIFS value is additive to the TIPG.IPGT value. Adaptive IFS Value (**all remaining Ethernet controllers**) This value is in units of 8 ns.
Reserved	31:16	0b	Reserved Should be written with 0b.

13.4.36 Transmit Descriptor Base Address Low

TDBAL (03800h; R/W)

This register contains the lower bits of the 64-bit transmit Descriptor base address. The base register indicates the start of the circular transmit descriptor queue. Since each descriptor is 16 bits in length, the lower four bits are ignored as the Transmit Descriptor Base Address must point to a 16-byte aligned block of data.

Table 13-79. TDBAL Register Bit Description

31 ... 4	3 ... 0
TDBAL	0

Field	Bit(s)	Initial Value	Description
ZERO	3:0	0b	Zero Value This field is ignored on write and reads as 0b.
TDBAL	31:4	X	Transmit Descriptor Base Address Low [31:4] This register indicates lower 32 bits of the start address for the transmit descriptor ring buffer.

13.4.37 Transmit Descriptor Base Address High

TDBAH (03804h; R/W)

This register contains the upper 32 bits of the 64-bit transmit Descriptor base address.

Table 13-80. TDBAH Register Bit Description

31 ... 0
TDBAH

Field	Bit(s)	Initial Value	Description
TDBAH	31:0	X	Transmit Descriptor Base Address [63:32] This register indicates upper 32 bits of the start address for the transmit descriptor ring buffer.

13.4.38 Transmit Descriptor Length

TDLEN (03808h; R/W)

This register determines the number of bytes allocated to the transmit descriptor circular buffer. This value must be a multiple of 128 bytes (the maximum cache line size). Since each descriptor is 16 bits in length, the total number of receive descriptors is always a multiple of eight.

Table 13-81. TDLEN Register Bit Description

31 ... 20	19 ... 7	6 ... 0
Reserved	LEN	0

Field	Bit(s)	Initial Value	Description
ZERO	6:0	0b	Ignore on write. Reads back as 0b.

LEN	19:7	0b	Descriptor Length. Number of bytes allocated to the transmit descriptor circular buffer.
Reserved	31:20	0b	Reserved Reads as 0b. Should be written with 0b for future compatibility.

13.4.39 Transmit Descriptor Head

TDH (03810h; R/W)

This register contains the head pointer for the transmit descriptor ring. It holds a value that is an offset from the base, and indicates the in–progress descriptor. It points to a 16-byte datum. Hardware controls this pointer. The only time that software should write to this register is after a reset (TCTL.RST or CTRL.RST) and before enabling the transmit function (TCTL.EN). If software were to write to this register while the transmit function was enabled, the on-chip descriptor buffers can be invalidated and indeterminate operation can result. Reading the transmit descriptor head to determine which buffers have been used (and can be returned to the memory pool) is not reliable.

Table 13-82. TDH Register Bit Description

31 … 16	15 … 0
Reserved	TDH

Field	Bit(s)	Initial Value	Description
TDH	15:0	0b	Transmit Descriptor Head
Reserved	31:16	0b	Reserved Reads as 0b. Should be written with 0b for future compatibility.

13.4.40 Transmit Descriptor Tail

TDT (03818h; R/W)

This register contains the tail pointer for the transmit descriptor ring. It holds a value that is an offset from the base, and indicates the location beyond the last descriptor hardware can process.

This is the location where software writes the first new descriptor. It points to a 16-byte datum. Software writes the tail pointer to add more descriptors to the transmit ready queue. Hardware attempts to transmit all packets referenced by descriptors between head and tail.

Table 13-83. TDT Register Bit Description

31 — 16	15 — 0
Reserved	TDT

Field	Bit(s)	Initial Value	Description
TDT	15:0	0b	Transmit Descriptor Tail
Reserved	31:16	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

13.4.41 Transmit Interrupt Delay Value

TIDV (03820h; R/W)

This register contains the transmit interrupt delay value. It determines the amount of time that elapses between writing back a descriptor that has Report Status (RS) [or Report Packet Sent (RPS) for the **82544GC/EI**] and Interrupt Delay Enable (IDE) set in the transmit descriptor and when the IMS.TXDW (Transmit Descriptor Written Back) interrupt bit is set. Counts are in units of 1.024 µs. A value of 0b is not allowed. The TIDV value is used to force timely write–back of descriptors as well. The first packet after timer initialization starts the timer. Timer expiration flushes any accumulated descriptors and sets an interrupt event (TXDW).

Table 13-84. TIDV Register Bit Description

31 … 16	15 … 0
Reserved	IDV

Field	Bit(s)	Initial Value	Description
IDV	15:0	X	Interrupt Delay Value Counts in units of 1.024 µs. A value of 0bis not allowed.
Reserved	31:16	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

13.4.42 TX DMA Control (82544GC/EI only)

TXDMAC (03000h; R/W)

This register controls the transmit DMA pre-fetching and preemption abilities.

Table 11-85. TXDMAC Register Bit Description

31 … 1	0
Reserved	DPP

Field	Bit(s)	Initial Value	Description
DPP	0	1	Disable packet prefetching When set, prevents the Ethernet controller from starting a transmit descriptor data fetch before it has finished processing the previous descriptor. In general, performance increases when this bit is set to 0b.
Reserved	31:1	0	Reserved Reads as 0b. Should be written to 0b for future compatibility.

13.4.43 Transmit Descriptor Control

TXDCTL (03828h; R/W)

This register controls the fetching and write back of transmit descriptors. The three threshold values provided are used to determine when descriptors are read from and written to host memory. The values can be in units of cache lines or descriptors (each descriptor is 16 bytes).

Table 13-86. TXDCTL Register Bit Description

31 25	24	23 22	21 16	15 14	13 8	7 6	5 0
LWTHRESH RSV[1]	GRAN	RSV	WTHRESH	RSV	HTHRESH	RSV	PTHRESH

1. **82544GC/EI** only.

Field	Bit(s)	Initial Value	Description
PTHRESH	5:0	0b	Prefetch Threshold Used to control when a pre-fetch of descriptors is considered. This threshold refers to the number of valid, unprocessed transmit descriptors the Ethernet controller has in its on-chip buffer. If this number drops below PTHRESH, the algorithm considers prefetching descriptors from host memory. This fetch does not happen unless there are at least TXDCTL.HTHRESH valid descriptors in host memory to fetch. Value of PTHRESH can be in either cache line units, or based on number of descriptors based on TXDCTL.GRAN.
Reserved	7:6	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility.
HTHRESH	15:8	0b	Host Threshold Provides the threshold of the valid descriptors in host memory A descriptor prefetch is performed each time enough valid descriptors (TXDCTL.HTHRESH) are available in host memory, no other DMA activity of greater priority is pending (descriptor fetches and write backs or packet data transfers) and the number of transmit descriptors the Ethernet controller has on its on-chip buffers drops below TXDCTL.PTHRESH. The value of HTHRESH can be in either cache line units, or based on number of descriptors based on TXDCTL.GRAN.
Reserved	15:14	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility.
WTHRESH	21:16	0b	Write Back Threshold WTHRESH controls the write back of processed transmit descriptors. This threshold refers to the number of transmit descriptors in the Ethernet controller's on-chip buffer which are ready to be written back to host memory. In the absence of external events (explicit flushes), the write back occurs only after more than WTHRESH descriptors are available for write back. WTHRESH must contain a non-zero value to take advantage of the write back bursting capabilities of the Ethernet controller. A value of 0b causes the descriptors to be written back as soon as they are processed. The value of WTHRESH can be in either cache line units, or based on number of descriptors based on RXDCTL.GRAN.

Field	Bit(s)	Initial Value	Description
Reserved	23:22	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility.
GRAN	24	0b	Granularity Set the values of PTHRESH, HTHRESH and WTHRESH in units of cache lines or descriptors (each descriptor is 16 bytes) 1b = Descriptor granularity. 0b = Cache line granularity.
LWTHRESH[1]	31:25	0h	Transmit descriptor Low Threshold Interrupt asserted when the number of descriptors pending service in the transmit descriptor queue (processing distance from the TDT) drops below this threshold.

1. Not applicable to the **82544GC/EI**.

Since write back of transmit descriptors is optional (under the control of RS bit in the descriptor), not all processed descriptors are counted with respect to WTHRESH. Descriptors start accumulating after a descriptor with RS (or RPS for the **82544GC/EI**) is set. Furthermore, with transmit descriptor bursting enabled, some descriptors are written back that did not have RS (or RPS for the **82544GC/EI**) set in their respective descriptors.

LWTHRESH (not applicable to the **82544GC/EI**) controls the number of pre-fetched transmit descriptors at which a transmit descriptor-low interrupt (ICR.TXD_LOW) is reported. This can enable software to operate more efficiently by maintaining a continuous addition of transmit work, interrupting only when the hardware nears completion of all submitted work. LWTHRESH specifies a multiple of eight descriptors. An interrupt is asserted when the number of descriptors available transitions from (threshold level=8*LWTHRESH)+1 to (threshold level=8*LWTHRESH). Setting this value to 0b causes this interrupt to be generated only when the transmit descriptor cache becomes completely empty.

13.4.44 Transmit Absolute Interrupt Delay Value[1]

TADV (0382Ch; RW)

31 16	15 0
Reserved	IDV

Field	Bit(s)	Initial Value	Description
IDV	15:0	0b	Interrupt Delay Value Counts in units of 1.024 μs. (0b = disabled)
Reserved	31:16	0b	Reads as 0b. Should be written to 0b for future compatibility.

1. Not applicable to the **82544GC/EI**.

The transmit interrupt delay timer (TIDV) can be used to coalesce transmit interrupts. However, it might be necessary to ensure that no completed transmit remains unnoticed for too long an interval in order ensure timely release of transmit buffers. This register can be used to ENSURE that a transmit interrupt occurs at some predefined interval after a transmit is completed. Like the delayed-transmit timer, the absolute transmit timer ONLY applies to transmit descriptor operations where (a) interrupt-based reporting is requested (RS set) and (b) the use of the timer function is requested (IDE is set).

This feature operates by initiating a countdown timer upon successfully transmitting the buffer. When the timer expires, a transmit-complete interrupt (ICR.TXDW) is generated. The occurrence of either an immediate (non-scheduled) or delayed transmit timer (TIDV) expiration interrupt halts the TADV timer and eliminates any spurious second interrupts.

Setting the value to 0b disables the transmit absolute delay function. If an immediate (non-scheduled) interrupt is desired for any transmit descriptor, the descriptor IDE should be set to 0b.

13.4.45 TCP Segmentation Pad And Minimum Threshold

TSPMT (03830h; RW)

This register specifies fields affecting the Ethernet controller behavior during TCP Segmentation operations. Values are specified in bytes. For normal (non TCP Segmentation) operations, the Ethernet controller's transmit DMA never begins servicing an individual data descriptor unless the transmit Packet Buffer has sufficient room to accept all of the data associated with the descriptor. However, for TCP Segmentation operations, it might be desirable to use a data descriptor that refers to a larger contiguous buffer in host memory than is actually allocated for the transmit Packet Buffer. For TCP segmentation, then, the transmit DMA is able to initiate smaller transfers than the entire descriptor's data length field.

31 16	15 0
TSPBP	TSMT

Field	Bit(s)	Initial Value	Description
TSMT	15:0	0400h	TCP Segmentation Minimum Transfer
TSPBP	31:16	0100h	TCP Segmentation Packet Buffer Padding

When performing TCP segmentation, the packet prototype header initially transferred by DMA is stored internally and updated as each packet of the TCP segmentation operation is composed. As data for subsequent TCP segments is DMA'd into the Ethernet controller, the frame header for each segment is dynamically inserted in front of the frame payload data stream prior to being written to the packet buffer. In order to obtain the most efficient use of burst DMA operations, the transmit DMA attempts to fetch as much data from a descriptor as possible, rather than limiting itself to bursting each data segment individually. However, to do this, sufficient packet buffer space must be reserved to account for all headers are inserted into the fetched data stream, as the burst might span multiple data segments. The calculation of how much packet buffer space should be reserved is dependent on the MSS being used, the maximum-sized data buffer pointed to by a descriptor, and the current header size. This calculation is left to software to pre-calculate for the worst-case usage. The TSPBP register allows configuration of this buffer space that must be reserved as "pad" for worst-case header insertion.   To ensure that this value does not prevent descriptors from being serviced at all, it is necessary that the transmit packet buffer allocation should be larger than the sum of (maximum TCP HDRLEN + maximum MSS + TMPBP + 80 bytes).

Because the DMA attempts to issue burst fetches for a much data as possible, it is possible for the transmit DMA to cause the transmit packet buffer to approach fullness (less the pad specified). However, if the packet buffer empties slightly, the transmit DMA could initiate a series of small transfers. To further optimize the efficiency of the transmit DMA during TCP segmentation operation, the TSMT register allows configuration of the minimum number of bytes that the DMA should attempt to transfer in a single burst operation. The transmit DMA uses this value to refrain from issuing a burst read until at least TSMT bytes of data from the current data descriptor can be stored in the packet buffer. This check is ignored if, after a series of DMA operations, the descriptor contains a smaller number of unfetched data bytes. To ensure that this minimum threshold does not prevent descriptors from being serviced at all, it is necessary that the transmit packet buffer allocation should be larger than the sum of (TSMT + TSPBP + 80 bytes). To ensure compliance with PCI-X specifications, this value should not be programmed to exceed 4 KB (the largest single-burst transfer allowed by PCI-X).

The transmit DMA further refrains from initiating service of a new data descriptor unless sufficient packet buffer space exists to at least fetch a full data segment or complete a partially-fetched segment.

13.4.46 Receive Descriptor Control

RXDCTL (02828h; R/W)

This register controls the fetching and write-back of receive descriptors. The three threshold values are used to determine when descriptors are read from and written to host memory. The values can be in units of cache lines or descriptors (each descriptor is 16 bytes) based on the GRAN flag. If GRAN = 0b (specifications are in cache-line granularity), the thresholds specified (based on the cacheline size specified in the PCI header CLS field) must not represent greater than 31 descriptors.

Table 13-87. RXDCTL Register Bit Description

31 25	24	23 22	21 16	15 14	13 8	7 6	5 0
Reserved	GRAN	RSV	WTHRESH	RSV	HTHRESH	RSV	PTHRESH

Field	Bit(s)	Initial Value	Description
PTHRESH	5:0	0b	Prefetch Threshold Used to control when a prefetch of descriptors is considered. This threshold refers to the number of valid, unprocessed receive descriptors the Ethernet controller has in its on-chip buffer. If this number drops below PTHRESH, the algorithm considers prefetching descriptors from host memory. This fetch does not happen unless there are at least RXDCTL.HTHRESH valid descriptors in host memory to fetch. Value of PTHRESH can be in either cache line units, or based on number of descriptors based on RXDCTL.GRAN.
RSV	7:6	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility.
HTHRESH	13:8	0b	Host Threshold Provides the threshold of the valid descriptors in host memory. A descriptors prefetch is performed each time enough valid descriptors (TXDCTL.HTHRESH) are available in host memory, no other DMA activity of greater priority is pending (descriptor fetches and write backs or packet data transfers) and the number of receive descriptors the Ethernet controller has on its on-chip buffers drops below RXDCTL.PTHRESH. Value of HTHRESH can be in either cache line units, or based on number of descriptors based on RXDCTL.GRAN.
RSV	15:14	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility.

Field	Bit(s)	Initial Value	Description
WTHRESH	21:16	1b	Write Back Threshold WTHRESH controls the write back of processed receive descriptors. This threshold refers to the number of receive descriptors in the Ethernet controller's on-chip buffer which are ready to be written back to host memory. In the absence of external events (explicit flushes), the write back occurs only after more than WTHRESH descriptors are available for write back. WTHRESH must contain a non-zero value to take advantage of the write back bursting capabilities of the Ethernet controller. A value of 1b causes the descriptors to be written back as soon as one cache line is available. A value of WTHRESH can be in either cache line units, or based on number of descriptors based on RXDCTL.GRAN.
RSV	23:22	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility.
GRAN	24	1b	Granularity Set the values of PTHRESH, HTHRESH and WTHRESH in units of cache lines or descriptors (each descriptor is 16 bytes) 0b = Cache line granularity. 1b = Descriptor granularity.
Reserved	31:25	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility.

13.4.47 Receive Checksum Control

RXCSUM (05000h; R/W)

The Receive Checksum Control register controls the receive checksum offloading features of the Ethernet controller. The Ethernet controller supports the offloading of three receive checksum calculations: the Packet Checksum, the IP Header Checksum, and the TCP/UDP Checksum.

The frame types that are supported:

- Ethernet II
- Ethernet SNAP

Table 13-88. RXCSUM Register Bit Description

31:11	10	9	8	7 0
RSV	IPV6OFL[1]	TUOFLD	IPOFLD	PCSS

1. Not applicable to the **82544GC/EI**.

Field	Bit(s)	Initial Value	Description
PCSS	7:0	0b	Packet Checksum Start Controls the starting byte for the Packet Checksum calculation. The Packet Checksum is the one's complement over the receive packet, starting from the byte indicated by RXCSUM.PCSS (0b corresponds to the first byte of the packet), after stripping. For example, for an Ethernet II frame encapsulated as an 802.3ac VLAN[1] packet and with RXCSUM.PCSS set to 14, the packet checksum would include the entire encapsulated frame, excluding the 14-byte Ethernet header (DA,SA,Type/Length) and the 4-byte VLAN tag. The Packet Checksum does not include the Ethernet CRC if the RCTL.SECRC bit is set. Software must make the required offsetting computation (to back out the bytes that should not have been included and to include the pseudo-header) prior to comparing the Packet Checksum against the TCP checksum stored in the packet.
IPOFLD	8	0b	IP Checksum Off-load Enable RXCSUM.IPOFLD is used to enable the IP Checksum offloading feature. If RXCSUM.IPOFLD is set to 1b, the Ethernet controller calculates the IP checksum and indicates a pass/fail indication to software through the Checksum Error bit (CSE) in the ERROR field to the receive descriptor. If both RXCSUM.IPOFLD and RXCSUM.TUOFLD are set, the Checksum Error bit (CSE) is set if either checksum was incorrect. If neither RSCSUM.IPOFLD nor RXCSUM.TUOFLD is set, the Checksum Error bit (CSE) is be 0b for all packets.
TUOFLD	9	0b	TCP/UDP Checksum Off-load Enable RXCSUM.TUOFL is used to enable the TCP/UDP Checksum off-loading feature. When set to 1b, the Ethernet controller calculates the TCP or UDP checksum and indicate a pass/fail indication to software through the Checksum Error bit (CSE). If both RXCSUM.TUOFLD and RXCSUM.TUOFLD are set, the Checksum Error bit (CSE) is set if either checksum was incorrect. If neither RXCSUM.IPOFLD nor RXCSUM.TUOFLD is set, the Checksum Error bit (CSE) is 0b for all packets.
IPV6OFL[2]	10	0b	IPv6 Checksum Offload Enable If IPV6OFL is set to 1b, hardware parses IPv6 headers when parsing a receive packet. This applies to checksum offloading only.
Reserved	31:11	0b	Reserved Reads as 0b. Should be written with 0b for future compatibility.

1. Not applicable to the **82541ER**.
2. Not applicable to the **82544GC/EI**.

13.5 Filter Registers

This section contains detailed descriptions for those registers associated with the Ethernet controller's address filter capabilities.

13.5.1 Multicast Table Array

MTA[127:0] (05200h-053FCh; R/W)

The multicast table array is a way to extend address filtering beyond the 16 perfect in the Receive Address Register (RAR). Note that the MTA is an imperfect filter that allows you to filter on 4096 similar addresses using a much smaller data structure than would be required to store all 4096 addresses in a linear table such as a perfect filter.

The Ethernet controller provides a 4096-bit vector multicast table array that is used when all the 16 perfect filters in the Receive Address Registers (RAR) are used. There is one register per 32 bits of the Multicast Address Table for a total of 128 registers (thus the MTA[127:0] designation). The size of the word array depends on the number of bits implemented in the multicast address table. Software must mask to the desired bit on reads and supply a 32-bit word on writes. Accesses to this table must be 32-bit.

Table 13-89. MTA Register Bit Description

Field	Bit(s)	Initial Value	Description
MC Bit Vector	31:0	X	Multicast bit vector specifying 32 bits in the multicast address filter table.

The operating system provides a list of addresses that it would like to respond to. The driver fills in the Receive Address Registers (RAR) first, as these are exact matching addresses. If the OS provides more than the 16 addresses available in RARs, the overflow is put into the MTA. The MTA does not match the exact address, but a subset of the address. Each address filtered on is represented by a single bit within the MTA table. Software needs to do the same calculations that hardware does when checking against the MTA, so it can program the appropriate bit in the MTA. When the hardware receives an address, it goes through the RARs, and if it does not find a match, it does the same calculations that are described below on the address that it was given and only checks one bit in the MTA. If that bit is set, it allows the packet to pass. If that bit is not set, it drops the packet.

The calculation to find that bit is as follows (using the example of 12:34:56:78:9A:BCh):

Check the RCTL bits 13:12 to see what they are set to. In this example it is 00h that means that we only look at bits 47:36. This corresponds to 0BC9h in the example address (assuming that in your example 12 is the least significant byte and 0BCh is the most significant byte). The way the address is stored in memory is the same that it would be going out on the wire, which is the least significant byte is the first on the wire, so it looks like this:

BC:9A:78:56:34:12h so that the LSB (12) goes on the wire first. Breaking 0BC9h down into a word:

0BC9h = 0000_1011_1100_1001b

Of the 16 bits, look at bits 11:5, starting from zero. These seven bits corresponds to the row within the MTA table (the MTA has 128 rows which require seven bits to define). In the example, bits 11:5 are 1011110b. This corresponds to row 94.

Of these 16 bits, count out the first five bits, again starting from bit zero. These first five bits correspond to the bit within the row (the MTA is 32 bits wide which require five bits to define). In the example this is 01001b. This corresponds to bit nine. This is the offset within the row.

Therefore, software needs to set bit nine of row 94 in the MTA. If the OS removes this address from the filter list, software would need to clear this bit. This is the same bit that the hardware would check if it received a packet with an address of xx:xx:xx:xx:9x:BCh.

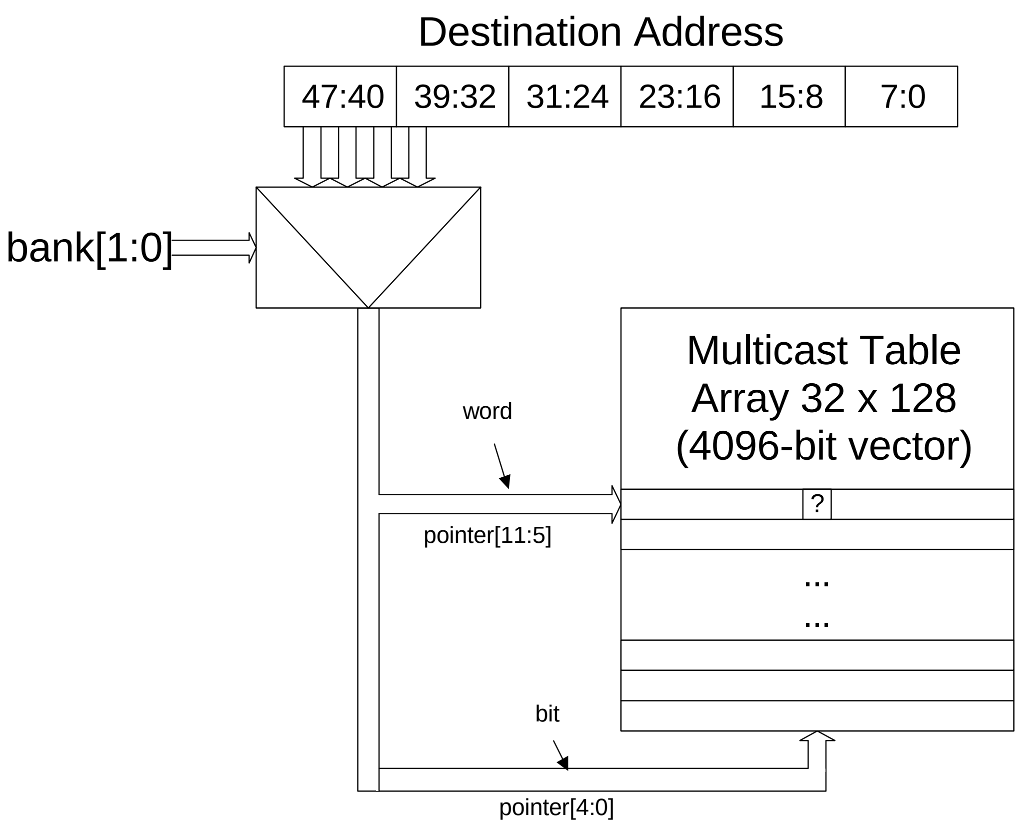


Figure 13-2. Multicast Table Array

13.5.2 Receive Address Low

RAL (05400h + 8*n; R/W)

16 registers contain the lower bits of the 48-bit Ethernet address. All 32 bits are valid. Software can access the High and Low registers as a register pair if it can perform a 64-bit access to the PCI bus. The addresses stored in these registers are used for unicast/multicast address filtering.

The first receive address register (RAL0, RAH0) is also used for exact match PAUSE frame checking (Valid PAUSE packet that is addressed to the station's address). Therefore, RAL0 and RAH0 always should be used to store the individual Ethernet MAC address of the Ethernet controller.

Table 13-90. RAL Register Bit Description

31 ... 0
RAL

Field	Bit(s)	Initial Value	Description
RAL	31:0	X	Receive address low Contains the lower 32-bit of the 48-bit Ethernet address. RAL0 should be used to store the lower 32-bit of the Ethernet controller's Ethernet MAC address.

13.5.3 Receive Address High

RAH (05404h + 8∗n; R/W)

16 registers contain the upper bits of the 48-bit Ethernet address. The complete address is {RAH, RAL}. Software can access the High and Low registers as a register pair if it can perform a 64-bit access to the PCI bus. The addresses stored in these registers are used for unicast/multicast address filtering.

The first receive address register (RAL0, RAH0) is also used for exact match Pause frame checking (Valid Pause packet that is addressed to the station's address). Therefore, RAL0 and RAH0 always should be used to store the individual Ethernet MAC address of the Ethernet controller.

Note: When writing to this register, always write low-to-high. When clearing this register, always clear high-to-low.

Table 13-91. RAH Register Bit Description

31	30 … 18	17 … 16	15 … 0
AV	Reserved	AS	RAH

Field	Bit(s)	Initial Value	Description
RAH	15:0	X	Receive address High Contains the upper 16 bits of the 48-bit Ethernet address. RAH0 should be used to store the upper 16-bit of the Ethernet controller's Ethernet MAC address.
AS	17:16	X	Address Select Selects how the address is to be used in the address filtering. 00b = Destination address (required for normal mode) 01b = Source address 10b = Reserved 11b = Reserved
Reserved	30:18	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.
AV	31	0b	Address Valid Determines whether this address is compared against the incoming packet. When set, the address is valid and is compared against the incoming packet. When cleared, the address is invalid and is not compared against the received packet. AV is only cleared by a PCI reset or software reset. This bit is unchanged by rx_reset.

13.5.4 VLAN Filter Table Array[1]

VFTA[127:0] (05600h – 057FCh; R/W)

The Ethernet controller provides a 4096-bit vector VLAN Filter table array. There is one register per 32 bits of the VLAN Filter Table, for a total of 128 registers (thus the VFTA[127:0] designation). The size of the word array depends on the number of bits implemented in the VLAN Filter table. Software must mask to the desired bit on reads, and supply a 32-bit word on writes. Accesses to this table must be 32-bit.

The algorithm for VLAN filtering using the VFTA is identical to that used for the Multicast Table Array. Refer to Section 13.5.1 for a block diagram of the algorithm. If VLANs are not used, there is no need to initialize the VFTA.

1. Not applicable to the **82541ER**.

Table 13-92. VFTA[127:0] Bit Description

31 ... 0
VLAN Filter Bit Vector

Field	Bit(s)	Initial Value	Description
Bit Vector	31:0	X	Double-word wide bit vector specifying 32 bits in the VLAN Filter table.

13.6 Wakeup Registers

13.6.1 Wakeup Control Register

WUC (05800h; R/W)

This register is reset any time LAN_PWR_GOOD is set to 0b. When AUX_POWER equals 0b, this register is also reset by de-asserting (rising edge) RST#.

31 ... 4	3	2	1	0
Reserved	APMPME	PME_S	PME_EN	APME

Field	Bit(s)	Initial Value	Description
APME	0	0b	Advance Power Management Enable If set to 1b, APM Wakeup is enabled. **Note:** Always 0b for the **82541ER**.
PME_En	1	0b	PME_En This read/write bit is used by the driver to access the PME_En bit of the Power Management Control / Status Register (PMCSR) without writing to the PCI configuration space. **Note:** Do not set this bit for the **82541ER**.
PME_Status	2	0b	PME_Status This bit is set when the Ethernet controller receives a wakeup event. It is the same as the PME_Status bit in the Power Management Control / Status Register (PMCSR). Writing a 1b to this bit clears the PME_Status bit in the PMCSR.

Field	Bit(s)	Initial Value	Description
APMPME	3	0b	Assert PME On APM Wakeup If set to 1b, the Ethernet controller sets the PME_Status bit in the Power Management Control / Status Register (PMCSR) and asserts PME# when APM Wakeup is enabled and the Ethernet controller receives a matching Magic Packet. This field value is loaded from the EEPROM. Note: Not applicable to the **82541ER**.
Reserved	27:4	0b	Reserved Reads as 0b.
Dynamic Powerdown[1]	28	0b	Dynamic Powerdown Mode When programmed to 1b, enables dynamic powerdown operation.
Auto Freq Select[1]	30:29	0b[2]	Automatic Frequency Select Determines automatic reduction of MAC frequency in **82541xx** only. Reserved for the **82547GI/EI**. Bit 29 controls MAC speed at 1000 MB. When cleared, enables the MAC to run at full speed. When set and PCI is configured for 33 MHz (**82541xx**), allows the MAC to run at half speed. Bit 30 controls MAC speed at other Ethernet rates. When cleared, allows the MAC to run at full speed. When set, it allows the MAC to run at quarter speed. If both bits are cleared, the MAC frequency select bits control MAC frequency.
SPM[1]	31	0b	Smart Powerdown MAC When programmed to 0b, the MAC operates normally. When programmed to 1b, the MAC enters smart powerdown mode.
Reserved[3]	31:28	0b	Reserved Reads as 0b.

1. **82541xx** and **82547GI/EI** only.
2. Loaded from the EEPROM.
3. Not applicable to the **82541xx** and **82547GI/EI**.

13.6.2 Wakeup Filter Control Register

WUFC (05808h; R/W)

This register is used to enable each of the pre-defined and flexible filters for wakeup support. A value of 1b means the filter is turned on, and a value of 0b means the filter is turned off.

This register is reset any time LAN_PWR_GOOD is 0b. When AUX_POWER equals 0b, this register is also reset by deasserting (rising edge) RST#.

31 20	19	18	17	16	15	14 8	7	6	5	4	3	2	1	0
Reserved	FLX3	FLX2	FLX1	FLX0	ITCO[1]	Reserved	IPv6[2]	IPv4[3]	ARP	BC	MC	EX	MAG	LNKC

1. **82541xx** and **82547GI/EI** only.
2. Not applicable to the **82544GC/EI**.
3. IP for the **82544GC/EI**.

Field	Bit(s)	Initial Value	Description
LNKC	0	0b	Link Status Change Wakeup Enable.
MAG	1	0b	Magic Packet Wakeup Enable.
EX	2	0b	Directed Exact Wakeup Enable.
MC	3	0b	Directed Multicast Wakeup Enable.
BC	4	0b	Broadcast Wakeup Enable.
ARP	5	0b	ARP Request Packet Wakeup Enable.
IPv4[1]	6	0b	Directed IPv4 Packet Wakeup Enable.
IPv6[2]	7	0b	Directed IPv6 Packet Wakeup Enable.
Reserved	14:8	0b	Reserved. Set these bits to 0b.
ITCO[3]	15	0	Ignore TCO/management packets for wakeup.
FLX0	16	0b	Flexible Filter 0 Enable.
FLX1	17	0b	Flexible Filter 1 Enable.
FLX2	18	0b	Flexible Filter 2 Enable.
FLX3	19	0b	Flexible Filter 3 Enable.

1. IP for the **82544GC/EI**.
2. Not applicable to the **82544GC/EI**.
3. **82541xx** and **82547GI/EI** only.

13.6.3 Wakeup Status Register

WUS (05810h; R)

This register is used to record statistics about all wakeup packets received. If a packet matches multiple criteria then multiple bits could be set. Writing a 1b to any bit clears that bit.

This register is not cleared when RST# is asserted. It is only cleared when LAN_PWR_GOOD is de-asserted or when cleared by the driver software.

31 20	19	18	17	16	15 8	7	6	5	4	3	2	1	0
Reserved	FLX3	FLX2	FLX1	FLX0	Reserved	IPv6[1]	IPv4[2]	ARP	BC	MC	EX	MAG	LNKC

1. Not applicable to the **82544GC/EI**.
2. IP for the **82544GC/EI**.

Field	Bit(s)	Initial Value	Description
LNKC	0	0b	Link Status Change.
MAG	1	0b	Magic Packet Received.
EX	2	0b	Directed Exact Packet Received The packet's address matched one of the 16 pre-programmed exact values in the Receive Address registers.
MC	3	0b	Directed Multicast Packet Received The packet was a multicast packet whose hashed to a value that corresponded to a 1 bit in the Multicast Table Array.
BC	4	0b	Broadcast Packet Received.
ARP	5	0b	ARP Request Packet Received.
IPv4[1]	6	0b	Directed IPv4 Packet Received.
IPv6[2]	7	0b	Directed IPv6 Packet Received.
FLX0	16	0b	Flexible Filter 0 Match.
FLX1	17	0b	Flexible Filter 1 Match.
FLX2	18	0b	Flexible Filter 2 Match.
FLX3	19	0b	Flexible Filter 3 Match.

1. IP for the **82544GC/EI**.
2. Not applicable to the **82544GC/EI**.

13.6.4 IP Address Valid

IPAV (5838h; R/W)

The IP Address Valid indicates whether the IP addresses in the IP Address Table are valid.

The valid bits are reset any time LAN_PWR_GOOD is 0b. When AUX_POWER equals 0b, the valid bits are also reset by deasserting (rising edge) RST#.

31 … 17	16	15 … 4	3	2	1	0
Reserved	v60[1]	Reserved	V43 V3[2]	V42 V2[2]	V41 V1[2]	V40 V0[2]

1. Not applicable to the **82544GC/EI**.
2. **82544GC/EI** only.

Field	Bit(s)	Initial Value	Description
V40 V0	0	0b[1]	IPv4 Address 0 Valid IP Address 0 Valid[2]
V41 V1	1	0b	IPv4 Address 1 Valid IP Address 1 Valid[2]
V42 V2	2	0b	IPv4 Address 2 Valid IP Address 2 Valid[2]
V43 V3	3	0b	IPv4 Address 3 Valid IP Address 3 Valid[2]
V60[3]	16	0b	IPv6 Address 0 Valid

1. The initial value is loaded from the IP Address Valid bit of the EEPROM's Management Control Register.
2. **82544GC/EI** only.
3. Not applicable to the **82544GC/EI**.

13.6.5 IPv4 Address Table[1]

IP4AT (05840h - 05858h; R/W)[2]

The IPv4 Address Table is used to store the four IP addresses for ARP Request packet and Directed IP packet wakeup for IPv4.

Note: This table is not cleared by any reset.

DWORD#	Address	31 0
0	5840h	IPV4ADDR0[1]
2	5848h	IPV4ADDR1
3	5850h	IPV4ADDR2
4	5858h	IPV4ADDR3

1. IPA for the **82544GC/EI**.

Field	Dword #	Address	Bit(s)	Initial Value	Description
IPV4ADDR0[1]	0	5840h	31:0	X[2]	IPv4 Address 0
IPV4ADDR1	2	5848h	31:0	X	IPv4 Address 1
IPV4ADDR2	4	5850h	31:0	X	IPv4 Address 2
IPV4ADDR3	6	5858h	31:0	X	IPv4 Address 3

1. IPA for the **82544GC/EI**.
2. The first entry is loaded from the EEPROM if the IP Address Valid field of the EEPROM's Management Control word is 1b and the IP Address Type field is 0b (IPv4). Otherwise, the value of this register is undefined after reset.

1. IP Address Table for the **82544GC/EI**.
2. IPAT for the **82544GC/EI**.

13.6.6 IPv6 Address Table[1]

IP6AT (05880h - 0588Ch; R/W)

The IPv6 Address Table is used to store the IPv6 addresses for ARP Request packet and Directed IP packet wakeup for IPv6.

Note: This table is not cleared by any reset.

DWORD#	Address	31 0
0	5880h	IPV6ADDR0
1	5884h	
2	5888h	
3	588Ch	

Field	Dword #	Address	Bit(s)	Initial Value[1]	Description
IPV6ADDR0	0	5880h	31:0	X	IPv6 Address 0, bytes 1-4
	1	5884h	31:0	X	IPv6 Address 0, bytes 5-8
	2	5888h	31:0	X	IPv6 Address 0, bytes 9-12
	3	588Ch	31:0	X	IPv6 Address 0, bytes 16-13

1. This table is loaded from the EEPROM if the IP Address Valid field of the EEPROM's Management Control word is 1b and the IP Address Type field is 0b (IPv4). Otherwise, the value of this register is undefined after reset.

1. Not applicable to the **82544GC/EI**.

13.6.7 Wakeup Packet Length

WUPL (05900h; R/W)

This register indicates the length of the first wakeup packet received. It is valid if one of the bits in the Wakeup Status Register (WUSR) is set. It can be written for diagnostic purposes and is not cleared by any reset.

31 ... 12	11 ... 0
Reserved	Length

Field	Bit(s)	Initial Value	Description
LEN	11:0	X	Length of wakeup packet. (If jumbo frames is enabled and the packet is longer than 2047 bytes this field is 2047.)

13.6.8 Wakeup Packet Memory (128 Bytes)

WUPM (05A00h - 05A7Ch; R/W)

This register is read-only and it is used to store the first 128 bytes of the wakeup packet for software retrieval after system wakeup. It can be written for diagnostic purposes and is not cleared by any reset.

31 ... 0
WUPD

Field	Bit(s)	Initial Value	Description
WUPD	31:0	X	Wakeup Packet Data

13.6.9 Flexible Filter Length Table

FFLT (05F00h - 05F18h; R/W)

The Flexible Filter Length Table stores the minimum packet lengths required to pass each of the Flexible Filters. Any packets that are shorter than the programmed length won’t pass that filter. Each Flexible Filter considers a packet that doesn’t have any mismatches up to that point to have passed the Flexible Filter when it reaches the required length. It does not check any bytes past that point.

All reserved fields read as 0b’s and ignore writes.

Before writing to the Flexible Filter Length Table the driver must first disable the flexible filters by writing 0b's to the Flexible Filter Enable bits of the Wakeup Filter Control Register (WUFC.FLXn).

31 0	31 11	10 0
Reserved	Reserved	Length 0
Reserved	Reserved	Length 1
Reserved	Reserved	Length 2
Reserved	Reserved	Length 3

Field	Dword #	Address	Bit(s)	Initial Value	Description
LEN0	0	5F00h	10:0	0b	Minimum Length for Flexible Filter 0
LEN1	2	5F08h	10:0	0b	Minimum Length for Flexible Filter 1
LEN2	4	5F10h	10:0	0b	Minimum Length for Flexible Filter 2
LEN3	6	5F18h	10:0	0b	Minimum Length for Flexible Filter 3

13.6.10 Flexible Filter Mask Table

FFMT (09000h - 093F8h; R/W)

The Flexible Filter Mask and Table is used to store the four 1-bit masks for each of the first 128 data bytes in a packet, one for each Flexible Filter. If the mask bit is set to 1b, the corresponding Flexible Filter compares the incoming data byte at the index of the mask bit to the data byte stored in the Flexible Filter Value Table.

Before writing to the Flexible Filter Mask Table the driver must first disable the flexible filters by writing 0b's to the Flexible Filter Enable bits of the Wakeup Filter Control Register (WUFC.FLXn).

31 0	31 4	3 0
Reserved	Reserved	Byte 0 Mask
Reserved	Reserved	Byte 1 Mask
Reserved	Reserved	Byte 2 Mask
Reserved	Reserved	Byte 126 Mask
Reserved	Reserved	Byte 127 Mask

Field	Dword #	Address	Bit(s)	Initial Value	Description
MASK0	0	9000h	15:0	X	Mask for Filter [3:0] for Byte 0
MASK1	2	9008h	15:0	X	Mask for Filter [3:0] for Byte 2
MASK2	4	9010h	15:0	X	Mask for Filter [3:0] for Byte 3
...					
MASK127	254	93F8h	15:0	X	Mask for Filter [3:0] for Byte 127

13.6.11 Flexible Filter Value Table

FFVT (09800h - 09BF8h; R/W)

The Flexible Filter Value and Table is used to store the one value for each byte location in a packet for each flexible filter. If the corresponding mask bit is set to 1b, the Flexible Filter compares the incoming data byte to the values stored in this table.

Before writing to the Flexible Filter Value Table the driver must first disable the flexible filters by writing 0b's to the Flexible Filter Enable bits of the Wakeup Filter Control Register (WUFC.FLXn).

31   0
Reserved
Reserved
Reserved
Reserved

31 24	23 16	15 8	7 0
Byte0: Value3	Value2	Value1	Value0
Byte1: Value3	Value2	Value1	Value0
Byte2: Value3	Value2	Value1	Value0
Byte127: Value3	Value2	Value1	Value0

Field	Dword #	Address	Bit(s)	Initial Value	Description
MASK0	0	9800h	15:0	X	Mask for Filter [3:0] for Byte 0
MASK1	2	9808h	15:0	X	Mask for Filter [3:0] for Byte 2
MASK2	4	9810h	15:0	X	Mask for Filter [3:0] for Byte 3
...					
MASK127	254	9BF8h	15:0	X	Mask for Filter [3:0] for Byte 127

13.7 Statistics Registers

All statistics registers are implemented as 32-bit registers. 64-bit accesses to these registers must have the upper byte enables de-asserted. 32-bit registers with addresses not on a quadword boundary cannot be accessed through a 64-bit access.

Registers that count octets make up 64-bit registers.

All Statistics registers reset when read. 64-bit registers reset whenever the upper 32 bits are read. In addition, they stick at FFFFh_FFFFh when the maximum value is reached.

The Statistics registers are not hardware initialized. Their default value is unknown. Software should read the contents of all registers in order to clear them prior to enabling the receive and transmit channels.

Note: For the receive statistics, it should be noted that a packet is indicated as "received" if it passes the device filters, and it is placed in the packet buffer memory. A packet does not have to be transferred to host memory in order to be counted as "received."

13.7.1 CRC Error Count

CRCERRS (04000h; R)

Counts the number of receive packets with CRC errors. In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusively) in length. If receives are not enabled, then this register does not increment.

Table 13-93. CRCERRS Register Bit Description

31 ... 0
CEC

Field	Bit(s)	Initial Value	Description
CEC	31:0	0b	CRC error count

13.7.2 Alignment Error Count

ALGNERRC (04004h; R)

Counts the number of receive packets with alignment errors (the packet is not an integer number of bytes in length). In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusively) in length. If receives are not enabled, then this register does not increment. This register is valid only in MII mode during 10/100 Mb/s operation.

Table 13-94. ALGNERRC Register Bit Description

31	0
AEC	

Field	Bit(s)	Initial Value	Description
AEC	31:0	0b	Alignment error count

13.7.3 Symbol Error Count

SYMERRS (04008h; R)

Counts the number of symbol errors between reads. The count increases for every bad symbol received, whether or not a packet is currently being received and whether or not the link is up. This register only increments in internal SerDes mode (TBI mode for the **82544GC/EI**).

Table 13-95. SYMERRS Register Bit Description

31	0
SYMERRS	

Field	Bit(s)	Initial Value	Description
SYMERRS	31:0	0b	Symbol Error Count

13.7.4 RX Error Count

RXERRC (0400Ch; R)

Counts the number of packets received in which I_RX_ER was asserted by the PHY. In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusively) in length. If receives are not enabled, then this register does not increment. In internal SerDes mode (TBI mode for the **82544GC/EI**), this register increments on the reception of /V/ codes.

Table 13-96. RXERRC Register Bit Description

31	0
RXEC	

Field	Bit(s)	Initial Value	Description
RXEC	31:0	0b	RX error count

13.7.5 Missed Packets Count

MPC (04010h; R)

Counts the number of missed packets. Packets are missed when the receive FIFO has insufficient space to store the incoming packet. This can be caused because of too few buffers allocated, or because there is insufficient bandwidth on the PCI bus. Events setting this counter cause RXO, the Receiver Overrun Interrupt, to be set. This register does not increment if receives are not enabled.

These packets are also counted in the Total Packets Received register as well as in Total Octets Received.

Table 13-97. MPC Register Bit Description

31	0
MPC	

Field	Bit(s)	Initial Value	Description
MPC	31:0	0b	Missed Packets Count

13.7.6 Single Collision Count

SCC (04014h; R)

This register counts the number of times that a successfully transmitted packet encountered a single collision. This register only increments if transmits are enabled and the Ethernet controller is in half-duplex mode.

Table 13-98. SCC Register Bit Description

31 ... 0
SCC

Field	Bit(s)	Initial Value	Description
SCC	31:0	0b	Number of times a transmit encountered a single collision.

13.7.7 Excessive Collisions Count

ECOL (04018h; R)

When 16 or more collisions have occurred on a packet, this register increments, regardless of the value of collision threshold. If collision threshold is set below 16, this counter won't increment. This register only increments if transmits are enabled and the Ethernet controller is in half-duplex mode.

Table 13-99. ECOL Register Bit Description

31 ... 0
ECC

Field	Bit(s)	Initial Value	Description
ECC	31:0	0b	Number of packets with more than 16 collisions.

13.7.8 Multiple Collision Count

MCC (0401Ch; R)

This register counts the number of times that a transmit encountered more than one collision but less than 16. This register only increments if transmits are enabled and the Ethernet controller is in half-duplex mode.

Table 13-100. MCC Register Bit Description

31	0
MCC	

Field	Bit(s)	Initial Value	Description
MCC	31:0	0b	Number of times a successful transmit encountered multiple collisions.

13.7.9 Late Collisions Count

LATECOL (04020h; R)

Late collisions are collisions that occur after 64-byte time into the transmission of the packet while working in 10-100 Mb/s data rate, and 512 byte time into the transmission of the packet while working in the 1000 Mb/s data rate. This register only increments if transmits are enabled and the device is in half-duplex mode.

Table 13-101. LATECOL Register Bit Description

31	0
LCC	

Field	Bit(s)	Initial Value	Description
LCC	31:0	0b	Number of packets with late collisions.

13.7.10 Collision Count

COLC (04028h; R)

This register counts the total number of collisions that are not late collisions seen by the transmitter. This register only increments if transmits are enabled and the Ethernet controller is in half-duplex mode.

Table 13-102. COLC Register Bit Description

31	0
CCC	

Field	Bit(s)	Initial Value	Description
CCC	31:0	0b	Total number of collisions experienced by the transmitter.

13.7.11 Defer Count

DC (04030h; R)

This register counts defer events. A defer event occurs when the transmitter cannot immediately send a packet due to the medium being busy either because another device is transmitting, the IPG timer has not expired, half-duplex deferral events, reception of XOFF frames, or the link is not up. This register only increments if transmits are enabled. This counter does not increment for streaming transmits that are deferred due to TX IPG.

Table 13-103. DC Register Bit Description

31 0
CDC

Field	Bit(s)	Initial Value	Description
CDC	31:0	0b	Number of defer events.

13.7.12 Transmit with No CRS

TNCRS (04034h; R)

This register counts the number of successful packet transmissions in which the internal carrier sense signal from the PHY was not asserted within one slot time of start of transmission.

The PHY should assert the internal carrier sense signal during every transmission. Failure to do so may indicate that the link has failed, or the PHY has an incorrect link configuration. This register only increments if transmits are enabled. This register is not valid in internal SerDes[1] mode (TBI mode for the **82544GC/EI**), and is only valid when the Ethernet controller is operating at full duplex.

Table 13-104. TNCRS Register Bit Description

31 0
TNCRS

Field	Bit(s)	Initial Value	Description
TNCRS	31:0	0b	Number of transmissions without a CRS assertion from the PHY.

1. The **82540EP/EM**, **82541xx**, and **82547GI/EI** do not support SerDes functionality.

13.7.13 Sequence Error Count

SEC (04038h; R)

This register counts sequence error events. The proper sequence of 8b/10b symbols is as follows: idle, start-of-frame (SOF), data, pad (optional), end-of-frame (EOF), fill (optional), idle. Hardware increments this counter for any illegal sequence of delimiters. If the link is not up, this register does not increment. This register is only valid in internal SerDes mode (TBI mode for the **82544GC/EI**).

Table 13-105. SEC Register Bit Description

31	0
SEC	

Field	Bit(s)	Initial Value	Description
SEC	31:0	0b	Number of sequence error events.

13.7.14 Carrier Extension Error Count

CEXTERR (0403Ch; R)

This register counts the number of packets received in which the carrier extension error was signaled across the internal PHY interface. The PHY propagates carrier extension errors to the MAC when an error is detected during the carrier extended time of a packet reception. An extension error is signaled by the PHY by the encoding of 1Fh on the receive data inputs. This register only increments if receives are enabled and the Ethernet controller is operating at 1000 Mb/s. For example, internal PHY or internal SerDes modes (internal PHY or TBI modes for the **82544GC/EI**).

Table 13-106. CEXTERR Register Bit Description

31	0
CEXTERR	

Field	Bit(s)	Initial Value	Description
CEXTERR	31:0	0b	Number of packets received with a carrier extension error.

13.7.15 Receive Length Error Count

RLEC (04040h; R)

This register counts receive length error events. A length error occurs if an incoming packet passes the filter criteria but is undersized or oversized. Packets less than 64 bytes are undersized. Packets over 1522 bytes are oversized if LongPacketEnable is 0b (RCTL.LPE). If LongPacketEnable (LPE) is 1b, then an incoming packet is considered oversized if it exceeds 16384 bytes.

If receives are not enabled, this register does not increment. These lengths are based on bytes in the received packet from <Destination Address> through <CRC>, inclusively.

Table 13-107. RLEC Register Bit Description

31 ... 0
RLEC

Field	Bit(s)	Initial Value	Description
RLEC	31:0	0b	Number of packets with receive length errors.

13.7.16 XON Received Count

XONRXC (04048h; R)

This register counts the number of valid XON packets received. XON packets can use the global address, or the station address. This register only increments if receives are enabled.

Table 13-108. XONRXC Register Bit Description

31 ... 0
XONRXC

Field	Bit(s)	Initial Value	Description
XONRXC	31:0	0b	Number of XON packets received.

13.7.17 XON Transmitted Count

XONTXC (0404Ch; R)

This register counts the number of XON packets transmitted. These can be either due to a full queue or due to software initiated action (using TCTL.SWXOFF). This register only increments if transmits are enabled.

Table 13-109. XONTXC Register Bit Description

31	0
XONTXC	

Field	Bit(s)	Initial Value	Description
XONTXC	31:0	0b	Number of XON packets transmitted.

13.7.18 XOFF Received Count

XOFFRXC (04050h; R)

This register counts the number of valid XOFF packets received. XOFF packets can use the global address or the station address. This register only increments if receives are enabled.

Table 13-110. XOFFRXC Register Bit Description

31	0
XOFFRXC	

Field	Bit(s)	Initial Value	Description
XOFFRXC	31:0	0b	Number of XOFF packets received.

13.7.19 XOFF Transmitted Count

XOFFTXC (04054h; R)

This register counts the number of XOFF packets transmitted. These can be either due to a full queue or due to software initiated action (using TCTL.SWXOFF). This register only increments if transmits are enabled.

Table 13-111. XOFFTXC Register Bit Description

31	0
XOFFTXC	

Field	Bit(s)	Initial Value	Description
XOFFTXC	31:0	0b	Number of XOFF packets transmitted.

13.7.20 FC Received Unsupported Count

FCRUC (04058h; R)

This register counts the number of unsupported flow control frames that are received.

The FCRUC counter increments when a flow control packet is received that matches either the reserved flow control multicast address (in FCAH/L) or the MAC station address, and has a matching flow control type field match (to the value in FCT), but has an incorrect opcode field. This register only increments if receives are enabled.

Table 13-112. FCRUC Register Bit Description

31	0
FCRUC	

Field	Bit(s)	Initial Value	Description
FCRUC	31:0	0b	Number of unsupported flow control frames received.

13.7.21 Packets Received (64 Bytes) Count

PRC64 (0405Ch; R)

This register counts the number of good packets received that are exactly 64 bytes (from <Destination Address> through <CRC>, inclusively) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. This register does not include received flow control packets and increments only if receives are enabled.

Table 13-113. PRC64 Register Bit Description

31	0
PRC64	

Field	Bit(s)	Initial Value	Description
PRC64	31:0	0b	Number of packets received that are 64 bytes in length.

13.7.22 Packets Received (65-127 Bytes) Count

PRC127 (04060h; R)

This register counts the number of good packets received that are 65-127 bytes (from <Destination Address> through <CRC>, inclusively) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. This register does not include received flow control packets and increments only if receives are enabled.

Table 13-114. PRC127 Register Bit Description

31	0
PRC127	

Field	Bit(s)	Initial Value	Description
PRC127	31:0	0b	Number of packets received that are 65-127 bytes in length.

13.7.23 Packets Received (128-255 Bytes) Count

PRC255 (04064h; R)

This register counts the number of good packets received that are 128-255 bytes (from <Destination Address> through <CRC>, inclusively) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. This register does not include received flow control packets and increments only if receives are enabled.

Table 13-115. PRC225 Register Bit Description

31	0
PRC255	

Field	Bit(s)	Initial Value	Description
PRC255	31:0	0b	Number of packets received that are 128-255 bytes in length.

13.7.24 Packets Received (256-511 Bytes) Count

PRC511 (04068h; R)

This register counts the number of good packets received that are 256-511 bytes (from <Destination Address> through <CRC>, inclusively) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. This register does not include received flow control packets and increments only if receives are enabled.

Table 13-116. PRC551 Register Bit Description

31	0
PRC511	

Field	Bit(s)	Initial Value	Description
PRC511	31:0	0b	Number of packets received that are 256-511 bytes in length.

13.7.25 Packets Received (512-1023 Bytes) Count

PRC1023 (0406Ch; R)

This register counts the number of good packets received that are 512-1023 bytes (from <Destination Address> through <CRC>, inclusively) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. This register does not include received flow control packets and increments only if receives are enabled.

Table 13-117. PRC1023 Register Bit Description

31	0
PRC1023	

Field	Bit(s)	Initial Value	Description
PRC1023	31:0	0b	Number of packets received that are 512-1023 bytes in length.

13.7.26 Packets Received (1024 to Max Bytes) Count

PRC1522 (04070h; R)

This register counts the number of good packets received that are from 1024 bytes to the maximum (from <Destination Address> through <CRC>, inclusively) in length. The maximum is dependent on the current receiver configuration and the type of packet being received. If a packet is counted in Receive Oversized Count, it is not counted in this register (see Section 13.7.36). This register does not include received flow control packets and only increments if the packet has passed address filtering and receives are enabled.

Due to changes in the standard for maximum frame size for VLAN tagged frames in IEEE Standard 802.3, these Ethernet controllers accept packets which have a maximum length of 1522 bytes. The RMON statistics associated with this range has been extended to count 1522-byte long packets.

Table 13-118. PRC1522 Register Bit Description

31	0
PRC1522	

Field	Bit(s)	Initial Value	Description
PRC1522	31:0	0b	Number of packets received that are 1024-Max bytes in length.

13.7.27 Good Packets Received Count

GPRC (04074h; R)

This register counts the number of good packets received of any legal length. The legal length for the received packet is defined by the value of LongPacketEnable (CTRL.LPE) (see Section 13.7.15). This register does not include received flow control packets and only counts packets that pass filtering. This register only increments if receives are enabled. This register does not count packets counted by the Missed Packet Count (MPC) register.

Table 13-119. GPRC Register Bit Description

31	0
GPRC	

Field	Bit(s)	Initial Value	Description
GPRC	31:0	0b	Number of good packets received (of any length).

13.7.28 Broadcast Packets Received Count

BPRC (04078h; R)

This register counts the number of good (no errors) broadcast packets received. This register does not count broadcast packets received when the broadcast address filter is disabled. This register only increments if receives are enabled.

Table 13-120. BPRC Register Bit Description

31	0
BPRC	

Field	Bit(s)	Initial Value	Description
BPRC	31:0	0b	Number of broadcast packets received.

13.7.29 Multicast Packets Received Count

MPRC (0407Ch; R)

This register counts the number of good (no errors) multicast packets received. This register does not count multicast packets received that fail to pass address filtering nor does it count received flow control packets. This register only increments if receives are enabled. This register does not count packets counted by the Missed Packet Count (MPC) register.

Table 13-121. MPRC Register Bit Description

31	0
MPRC	

Field	Bit(s)	Initial Value	Description
MPRC	31:0	0b	Number of multicast packets received.

13.7.30 Good Packets Transmitted Count

GPTC (04080h; R)

This register counts the number of good (no errors) packets transmitted. A good transmit packet is considered one that is 64 or more bytes in length (from <Destination Address> through <CRC>, inclusively) in length. This does not include transmitted flow control packets. This register only increments if transmits are enabled.

Table 13-122. GPTC Register Bit Description

31	0
GPTC	

Field	Bit(s)	Initial Value	Description
GPTC	31:0	0b	Number of good packets transmitted.

13.7.31 Good Octets Received Count

GORCL (04088h; R)/GORCH (0408Ch; R)

These registers make up a 64-bit register that counts the number of good (no errors) octets received. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusively. This register resets each time the upper 32 bits are read (GORCH).

In addition, it sticks at FFFFh_FFFFh_FFFFh_FFFFh when the maximum value is reached. Only octets of packets that pass address filtering are counted in this register. This register only increments if receives are enabled.

These octets do not include octets of received flow control packets.

Table 13-123. GORCL and GORCH Register Bit Description

31 0	31 0
GORCH	GORCL

Field	Bit(s)	Initial Value	Description
GORCL	31:0	0b	Number of good octets received – lower 4 bytes.
GORCH	31:0	0b	Number of good octets received – upper 4 bytes.

13.7.32 Good Octets Transmitted Count

GOTCL (04090h; R)/ GOTCH (04094; R)

These registers make up a 64-bit register that counts the number of good (no errors) octets transmitted. This register resets each time the upper 32 bits are read (GOTCH).

In addition, it sticks at FFFF_FFFF_FFFF_FFFFh when the maximum value is reached. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusively. This register counts octets in successfully transmitted packets that are 64 or more bytes in length. This register only increments if transmits are enabled.

These octets do not include octets in transmitted flow control packets.

Table 13-124. GOTCL and GOTCH Register Bit Description

31 0	31 0
GOTCH	GOTCL

Field	Bit(s)	Initial Value	Description
GOTCL	31:0	0b	Number of good octets transmitted – lower 4 bytes.
GOTCH	31:0	0b	Number of good octets transmitted – upper 4 bytes.

13.7.33 Receive No Buffers Count

RNBC (040A0h; R)

This register counts the number of times that frames were received when there were no available buffers in host memory to store those frames (receive descriptor head and tail pointers were equal). The packet is still received if there is space in the FIFO. This register only increments if receives are enabled.

This register does not increment when flow control packets are received.

Table 13-125. RNBC Register Bit Description

31	0
RNBC	

Field	Bit(s)	Initial Value	Description
RNBC	31:0	0b	Number of receive no buffer conditions.

13.7.34 Receive Undersize Count

RUC (040A4h; R)

This register counts the number of received frames that passed address filtering, and were less than minimum size (64 bytes from <Destination Address> through <CRC>, inclusively), and had a valid CRC. This register only increments if receives are enabled.

Table 13-126. RUC Register Bit Description

31	0
RUC	

Field	Bit(s)	Initial Value	Description
RUC	31:0	0b	Number of receive undersize errors.

13.7.35 Receive Fragment Count

RFC (040A8h; R)

This register counts the number of received frames that passed address filtering, and were less than minimum size (64 bytes from <Destination Address> through <CRC>, inclusively), but had a bad CRC (this is slightly different from the Receive Undersize Count register). This register only increments if receives are enabled.

Table 13-127. RFC Register Bit Description

31 ... 0
RFC

Field	Bit(s)	Initial Value	Description
RFC	31:0	0b	Number of receive fragment errors.

13.7.36 Receive Oversize Count

ROC (040ACh; R)

This register counts the number of received frames with valid CRC field that passed address filtering, and were greater than maximum size. Packets over 1522 bytes are oversized if LongPacketEnable (RCTL.LPE) is 0b. If LongPacketEnable is 1b, then an incoming packet is considered oversized if it exceeds 16384 bytes.

If receives are not enabled, this register does not increment. These lengths are based on bytes in the received packet from <Destination Address> through <CRC>, inclusively.

Table 13-128. ROC Register Bit Description

31 ... 0
ROC

Field	Bit(s)	Initial Value	Description
ROC	31:0	0b	Number of receive oversize errors.

13.7.37 Receive Jabber Count

RJC (040B0h; R)

This register counts the number of received frames that passed address filtering, and were greater than maximum size and had a bad CRC (this is slightly different from the Receive Oversize Count register).

Packets over 1522 bytes are oversized if LongPacketEnable (RCTL.LPE) is 0b.
If LongPacketEnable is 1b, then an incoming packet is considered oversized if it exceeds 16384 bytes.

If receives are not enabled, this register does not increment. These lengths are based on bytes in the received packet from <Destination Address> through <CRC>, inclusively.

Table 13-129. RJC Register Bit Description

31	0
RJC	

Field	Bit(s)	Initial Value	Description
RJC	31:0	0b	Number of receive jabber errors.

13.7.38 Management Packets Received Count[1]

MGTPRC (040B4h; R)

This register counts the total number of packets received that pass the management filters as described in the appropriate Total Cost of Ownership (TCO) System Management Bus Interface Application Notes. Management packets include RMCP and ARP packets. Any packets with errors are not counted, except that packets dropped because the management receive FIFO is full or the packet is longer than 200 bytes is counted.

31	0
MGTPRC	

Field	Bit(s)	Initial Value	Description
MGTPRC	31:0	0b	Number of management packets received.

1. Not applicable to the **82544GC/EI** or **82541ER**.

13.7.39 Management Packets Dropped Count[1]

MGTPDC (040B8h; R)

This register counts the total number of packets received that pass the management filters as described in the appropriate Total Cost of Ownership (TCO) System Management Bus Interface Application Notes and then are dropped because the management receive FIFO is full or the packet is longer than 200 bytes. Management packets include RMCP and ARP packets.

31 ... 0
MGTPDC

Field	Bit(s)	Initial Value	Description
MGTPDC	31:0	0b	Number of management packets dropped.

13.7.40 Management Pkts Transmitted Count[1]

MGTPTC (040BCh; R)

This register counts the total number of packets that are transmitted that are either received over the SMBus or are generated by the Ethernet controller's ASF function.

31 ... 0
MGTPTC

Field	Bit(s)	Initial Value	Description
MGTPTC	31:0	0b	Number of management packets transmitted.

13.7.41 Total Octets Received

TORL (040C0h; R) / TORH (040C4h; R)

These registers make up a 64-bit register that counts the total number of octets received. This register resets each time the upper 32 bits are read (TORH). In addition, it sticks at FFFF_FFFF_FFFF_FFFFh when the maximum value is reached.

1. Not applicable to the **82544GC/EI** or **82541ER**.

All packets received have their octets summed into this register, regardless of their length, whether they are erred, or whether they are flow control packets. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusively. This register only increments if receives are enabled.

Table 13-130. TORL and TORH Register Bit Descriptions

31 0	31 0
TORH	TORL

Field	Bit(s)	Initial Value	Description
TORL	31:0	0b	Number of total octets received – lower 4 bytes.
TORH	31:0	0b	Number of total octets received – upper 4 bytes.

13.7.42 Total Octets Transmitted

TOTL (040C8h; R/W / TOTH (040CCh; R)

These registers make up a 64-bit register that counts the total number of octets transmitted. This register resets each time the upper 32 bits are read (TOTH). In addition, it sticks at FFFF_FFFF_FFFF_FFFFh when the maximum value is reached.

All transmitted packets have their octets summed into this register, regardless of their length or whether they are flow control packets. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusively.

Octets transmitted as part of partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled.

Table 13-131. TOTL and TOTH Register Bit Descriptions

31 0	31 0
TOTH	TOTL

Field	Bit(s)	Initial Value	Description
TOTL	31:0	0b	Number of total octets transmitted – lower 4 bytes.
TOTH	31:0	0b	Number of total octets transmitted – upper 4 bytes.

13.7.43 Total Packets Received

TPR (040D0h; R)

This register counts the total number of all packets received. All packets received are counted in this register, regardless of their length, whether they have errors, or whether they are flow control packets. This register only increments if receives are enabled.

Table 13-132. TPR Register Bit Description

31	0
TPR	

Field	Bit(s)	Initial Value	Description
TPR	31:0	0b	Number of all packets received.

13.7.44 Total Packets Transmitted

TPT (040D4h; R)

This register counts the total number of all packets transmitted. All packets transmitted are counted in this register, regardless of their length, or whether they are flow control packets.

Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMBus[1], and packets generated by the ASF function.

Table 13-133. TPT Register Bit Description

31	0
TPT	

Field	Bit(s)	Initial Value	Description
TPT	31:0	0b	Number of all packets transmitted.

1. The **82544GC/EI** and the **82541ER** do not support SMBus or ASF functionality.

13.7.45 Packets Transmitted (64 Bytes) Count

PTC64 (040D8h; R)

This register counts the number of packets transmitted that are exactly 64 bytes (from <Destination Address> through <CRC>, inclusively) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register does not include transmitted flow control packets (which are 64 bytes in length). This register only increments if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMBus, and packets generated by the ASF function.

Table 13-134. PTC64 Register Bit Description

31 ... 0
PTC64

Field	Bit(s)	Initial Value	Description
PTC64	31:0	0b	Number of packets transmitted that are 64 bytes in length.

13.7.46 Packets Transmitted (65-127 Bytes) Count

PTC127 (040DCh; R)

This register counts the number of packets transmitted that are 65-127 bytes (from <Destination Address> through <CRC>, inclusively) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMBus[1], and packets generated by the ASF function.

Table 13-135. PTC127 Register Bit Description

31 ... 0
PTC127

Field	Bit(s)	Initial Value	Description
PTC127	31:0	0b	Number of packets transmitted that are 65-127 bytes in length.

1. The **82544GC/EI** does not support SMBus or ASF functionality.

13.7.47 Packets Transmitted (128-255 Bytes) Count

PTC255 (040E0h; R)

This register counts the number of packets transmitted that are 128-255 bytes (from <Destination Address> through <CRC>, inclusively) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMBus, and packets generated by the ASF function.

Table 13-136. PTC255 Register Bit Description

31 ... 0
PTC255

Field	Bit(s)	Initial Value	Description
PTC255	31:0	0b	Number of packets transmitted that are 128-255 bytes in length.

13.7.48 Packets Transmitted (256-511 Bytes) Count

PTC511 (040E4h; R)

This register counts the number of packets transmitted that are 256-511 bytes (from <Destination Address> through <CRC>, inclusively) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled. This register counts all packets, including standard packets, secure packets, packets received over the SMBus[1], and packets generated by the ASF function.

Table 13-137. PTC511 Register Bit Description

31 ... 0
PTC511

Field	Bit(s)	Initial Value	Description
PTC511	31:0	0b	Number of packets transmitted that are 256-511 bytes in length.

1. The **82544GC/EI** does not support SMBus or ASF functionality.

13.7.49 Packets Transmitted (512-1023 Bytes) Count

PTC1023 (040E8h; R)

This register counts the number of packets transmitted that are 512-1023 bytes (from <Destination Address> through <CRC>, inclusively) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled.This register counts all packets, including standard packets, secure packets, packets received over the SMBus, and packets generated by the ASF function.

Table 13-138. PTC1023 Register Bit Description)

31	0
PTC1023	

Field	Bit(s)	Initial Value	Description
PTC1023	31:0	0b	Number of packets transmitted that are 512-1023 bytes in length.

13.7.50 Packets Transmitted (1024 Bytes or Greater) Count

PTC1522 (040ECh; R)

This register counts the number of packets transmitted that are 1024 or more bytes (from <Destination Address> through <CRC>, inclusively) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled.

Due to the maximum frame size for VLAN tagged frames in IEEE Standard 802.3ac, these Ethernet controllers transmit packets that have a maximum length of 1522 bytes. The RMON statistics associated with this range has been extended to count 1522 byte long packets. This register counts all packets, including standard packets, secure packets, packets received over the SMBus[1], and packets generated by the ASF function.

Table 13-139. PTC1522 Register Bit Description

31	0
PTC1522	

Field	Bit(s)	Initial Value	Description
PTC1522	31:0	0b	Number of packets transmitted that are 1024 or more bytes in length.

1. The **82544GC/EI** does not support SMBus or ASF functionality.

13.7.51 Multicast Packets Transmitted Count

MPTC (040F0h; R)

This register counts the number of multicast packets transmitted. This register does not include flow control packets and increments only if transmits are enabled. Counts clear as well as secure traffic.

Table 13-140. MPTC Register Bit Description

31	0
MPTC	

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0b	Number of multicast packets transmitted.

13.7.52 Broadcast Packets Transmitted Count

BPTC (040F4h; R)

This register counts the number of broadcast packets transmitted. This register only increments if transmits are enabled.Counts clear as well as secure traffic. (Management packets are never more than 200 bytes).

Table 13-141. BPTC Register Bit Description

31	0
BPTC	

Field	Bit(s)	Initial Value	Description
BPTC	31:0	0b	Number of broadcast packets transmitted count.

13.7.53 TCP Segmentation Context Transmitted Count

TSCTC (040F8h; R)

This register counts the number of TCP segmentation offload transmissions and increments once the last portion of the TCP segmentation context payload is segmented and loaded as a packet into the Ethernet controller's on-chip transmit buffer. Note that this is not a measurement of the number of packets sent out (covered by other registers). This register only increments if transmits and TCP Segmentation offload are enabled.

31	0
TSCTC	

Field	Bit(s)	Initial Value	Description
TSCTC	31:0	0b	Number of TCP Segmentation contexts transmitted count.

13.7.54 TCP Segmentation Context Transmit Fail Count

TSCTFC (040FCh; R)

This register counts the number of TCP segmentation offload requests to the hardware that failed to transmit all data in the TCP segmentation context payload. There is no indication by hardware of how much data was successfully transmitted. Only one failure event is logged per TCP segmentation context. Failures can be caused by excessive collisions or PAYLEN errors. This register only increments if transmits are enabled.

31	0
TSCTFC	

Field	Bit(s)	Initial Value	Description
TSCTFC	31:0	0b	Number of TCP Segmentation contexts where the Ethernet controller failed to transmit the entire data payload.

13.8 Diagnostics Registers

The Ethernet controller contains several diagnostic registers. These registers enable software to directly access the contents of the Ethernet controller's internal Packet Buffer Memory (PBM), also referred to as FIFO space. These registers also give software visibility into what locations in the PBM that the hardware currently considers to be the "head" and "tail" for both transmit and receive operations.

13.8.1 Receive Data FIFO Head Register

RDFH (02410h; R/W)

This register stores the head of the Ethernet controller's on–chip receive data FIFO. Since the internal FIFO is organized in units of 64-bit words, this field contains the 64-bit offset of the current Receive FIFO Head. So a value of "8h" in this register corresponds to an offset of 8 quadwords into the Receive FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-142. RDFH Register Bit Description

31 ... 13	12 ... 0
Reserved	FIFO Head

Field	Bit(s)	Initial Value	Description
FIFO Head	12:0	0b	Receive FIFO Head pointer.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.2 Receive Data FIFO Tail Register

RDFT (02418h; R/W)

This register stores the tail of the Ethernet controller's on–chip receive data FIFO. Since the internal FIFO is organized in units of 64-bit words, this field contains the 64-bit offset of the current Receive FIFO Tail. So a value of "8h" in this register corresponds to an offset of eight quadwords or into the Receive FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-143. RDFT Register Bit Description

31 … 13	12 … 0
Reserved	FIFO Tail

Field	Bit(s)	Initial Value	Description
FIFO Tail	12:0	0b	Receive FIFO Tail pointer.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.3 Receive Data FIFO Head Saved Register

RDFHS (02420h; R/W)

This register stores a copy of the Receive Data FIFO Head register in case the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-144. RDFHS Register Bit Description

31 … 13	12 … 0
Reserved	FIFO Head

Field	Bit(s)	Initial Value	Description
FIFO Head	12:0	0b	A “saved” value of the Receive FIFO Head pointer.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.4 Receive Data FIFO Tail Saved Register

RDFTS (02428h; R/W)

This register stores a copy of the Receive Data FIFO Tail register in case the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-145. RDFTS Register Bit Description

31 ... 13	12 ... 0
Reserved	FIFO Tail

Field	Bit(s)	Initial Value	Description
FIFO Tail	12:0	0b	A “saved” value of the Receive FIFO Tail pointer.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.5 Receive Data FIFO Packet Count

RDFPC (02430h; R/W)

This register reflects the number of receive packets that are currently in the Receive FIFO. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-146. RDFPC Register Bit Description

31 ... 13	12 ... 0
Reserved	FIFO Tail

Field	Bit(s)	Initial Value	Description
RX FIFO Packet Count	12:0	0b	The number of received packets currently in the RX FIFO.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.6 Transmit Data FIFO Head Register

TDFH (03410h; R/W)

This register stores the head of the Ethernet controller’s on–chip transmit data FIFO. Since the internal FIFO is organized in units of 64-bit words, this field contains the 64-bit offset of the current Transmit FIFO Head. So a value of “8h” in this register corresponds to an offset of 8 quadwords into the Transmit FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-147. TDFH Register Bit Description)

31 … 11	10 … 0
Reserved	FIFO Head

Field	Bit(s)	Initial Value	Description
FIFO Head	10:0	0b	Transmit FIFO Head pointer.
Reserved	31:11	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.7 Transmit Data FIFO Tail Register

TDFT (03418h; R/W)

This register stores the head of the Ethernet controller's on–chip transmit data FIFO. Since the internal FIFO is organized in units of 64-bit words, this field contains the 64-bit offset of the current Transmit FIFO Tail. So a value of "8h" in this register corresponds to an offset of 8 quadwords into the Transmit FIFO space. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-148. TDFT Register Bit Description

31 … 11	10 … 0
Reserved	FIFO Tail

Field	Bit(s)	Initial Value	Description
FIFO Tail	10:0	0b	Transmit FIFO tail pointer.
Reserved	31:11	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.8 Transmit Data FIFO Head Saved Register

TDFHS (03420h; R/W)

This register stores a copy of the Transmit Data FIFO Head register in case the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-149. TDFHS Register Bit Description

31 ... 13	12 ... 0
Reserved	FIFO Head

Field	Bit(s)	Initial Value	Description
FIFO Head	12:0	0b	A “saved” value of the Transmit FIFO Head pointer.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.9 Transmit Data FIFO Tail Saved Register

TDFTS (03428h; R/W)

This register stores a copy of the Transmit Data FIFO Tail register in case the internal register needs to be restored. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-150. TDFTS Register Bit Description

31 ... 13	12 ... 0
Reserved	FIFO Tail

Field	Bit(s)	Initial Value	Description
FIFO Tail	12:0	0b	A “saved” value of the Transmit FIFO Tail pointer.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.10 Transmit Data FIFO Packet Count

TDFPC (03430h; R/W)

This register reflects the number of packets to be transmitted that are currently in the Transmit FIFO. This register is available for diagnostic purposes only, and should not be written during normal operation.

Table 13-151. TDFPC Register Bit Description

31 … 13	12 … 0
Reserved	FIFO Tail

Field	Bit(s)	Initial Value	Description
FIFO Tail	12:0	0b	The number of packets to be transmitted that are currently in the TX FIFO.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.8.11 Packet Buffer Memory

PBM (10000h - 1FFFCh; R/W)

All PBM (FIFO) data is available to diagnostics. Locations can be accessed as 32-bit or 64-bit words. The internal PBM is 64 KB (40 KB for the **82547GI/EI**) in size. Software can configure the amount of PBM space that is used as the transmit FIFO versus the receive FIFO. The default is 16 KB of transmit FIFO space and 48 KB of receive FIFO space. For the **82547GI/EI**, the default is 18 KB of transmit FIFO space and 22 KB of receive FIFO space.

Regardless of the individual FIFO sizes that software configures, the RX FIFO is located first in the memory mapped PBM space. So for the default FIFO configuration, the RX FIFO occupies offsets 10000h - 1BFFFh of the memory mapped space, while the TX FIFO occupies offsets 1C000h - 1FFFFh of the memory mapped space.

Table 13-152. PBM Bit Description

31 … 0
FIFO Data

Field	Bit(s)	Initial Value	Description
FIFO Data	31:0	0b	Packet Buffer Data

Note: This page intentionally left blank.

General Initialization and Reset Operation 14

14.1 Introduction

This section lists all necessary initializations and describes the reset commands for the PCI/PCI-X Family of Gigabit Ethernet Controllers.

Note: TBI mode is used by the **82544GC/EI**. Internal SerDes is used by the **82546GB/EB** and **82545GM/EM**.

14.2 Power Up State

At power up, the Ethernet controller is not automatically configured by the hardware for normal operation. Software initialization is required before normal operation can continue. In general, the Ethernet controller is considered non-functional until the software driver successfully loads and sets up the hardware. However, Auto-Negotiation can start at power up or upon receipt of an assertion of PCI reset if configured to do so by the EEPROM.

14.3 General Configuration

Several values in the Device Control Register (CTRL) need to be set upon power up or after an Ethernet controller reset for normal operation.

- Speed and duplex are determined via Auto-Negotiation by the PHY, Auto-Negotiation by the MAC for internal SerDes[1] mode, or forced by software if the link is forced. In internal PHY mode, the Ethernet controller can be configured automatically by hardware or forced by software to the same configuration as the PHY.
- In internal PHY mode, the Auto-Speed Detection Enable (CTRL.ASDE) bit, when set to 1b, detects the resolved speed and duplex of the link and self-configure the MAC appropriately. This bit should be set in conjunction with the Set Link Up (CTRL.SLU) bit.
- The MAC can also be forced to a specific Speed/Duplex combination. This is accomplished by setting the Set Link Up (CTRL.SLU), Force Speed (CTRL. FRCSPD) and Force Duplex (CTRL.FRCDPLX) bits. Once speed and duplex are determined (either via Auto-Negotiation or forced by software), speed is forced by setting the appropriate Speed Selection (CTRL.SPEED) bits and duplex is forced by updating the Full Duplex (CTRL.FD) bit.
- For the **82541xx** and **82547GI/EI**, configure the LED behavior through LEDCTRL.
- Link Reset (CTRL.LRST) should be set to 0b (normal). The Ethernet controller defaults to LRST = 1b which disables Auto-Negotiation. A transition to 0b initiates the Auto-Negotiation function. LRST can be defined in the EEPROM. This bit is only valid in internal SerDes mode and has no effect in internal PHY mode.

1. The **82540EP/EM**, **82541xx**, and **82547GI/EI** do not support any SerDes functionality.

- PHY Reset (CTRL.PHY_RST) should be set to 0b. Setting this bit to 1b resets the PHY without accessing the PHY registers. This bit is ignored in internal SerDes mode.
- CTRL.ILOS should be set to 0b (not applicable to the **82541xx** and **82547GI/EI**).
- If Flow Control is desired, program the FCAH, FCAL, FCT and FCTTV registers. If not, they should be written with 0b. To enable XON frame transmission, the XON Enable (FCTRL.XONE) bit must be set. Advertising Flow Control capabilities during the Auto-Negotiation process is dependent on whether the Ethernet controller is operating in internal SerDes or internal PHY mode. In internal SerDes mode, the TXCW register must be set up prior to starting the Auto-Negotiation process. In internal PHY mode, the appropriate PHY registers must be set up properly to advertise desired capabilities prior to starting or re-starting the Auto-Negotiation process. The Receive Flow Control Enable (CTRL.RFCE) and Transmit Flow Control Enable (CTRL.TFCE) bits need to be explicitly set by software in internal PHY mode (because Auto-Negotiation is managed by PHY rather than the MAC), or when a fiber connection is desired but link was forced rather than Auto-Negotiated.
- If VLANs are not used, software should clear VLAN Mode Enable (CTRL.VME) bit. In this instance, there is no need then to initialize the VLAN Filter Table Array (VFTA). If VLANs are desired, the VFTA should be both initialized and loaded with the desired information.
- For the **82541xx** and **82547GI/EI**, clear all statistical counters.

14.4 Receive Initialization

Program the Receive Address Register(s) (RAL/RAH) with the desired Ethernet addresses. RAL[0]/RAH[0] should always be used to store the Individual Ethernet MAC address of the Ethernet controller. This can come from the EEPROM or from any other means (for example, on some machines, this comes from the system PROM not the EEPROM on the adapter port).

Initialize the MTA (Multicast Table Array) to 0b. Per software, entries can be added to this table as desired.

Program the Interrupt Mask Set/Read (IMS) register to enable any interrupt the software driver wants to be notified of when the event occurs. Suggested bits include RXT, RXO, RXDMT, RXSEQ, and LSC. There is no immediate reason to enable the transmit interrupts.

If software uses the Receive Descriptor Minimum Threshold Interrupt, the Receive Delay Timer (RDTR) register should be initialized with the desired delay time.

Allocate a region of memory for the receive descriptor list. Software should insure this memory is aligned on a paragraph (16-byte) boundary. Program the Receive Descriptor Base Address (RDBAL/RDBAH) register(s) with the address of the region. RDBAL is used for 32-bit addresses and both RDBAL and RDBAH are used for 64-bit addresses.

Set the Receive Descriptor Length (RDLEN) register to the size (in bytes) of the descriptor ring. This register must be 128-byte aligned.

The Receive Descriptor Head and Tail registers are initialized (by hardware) to 0b after a power-on or a software-initiated Ethernet controller reset. Receive buffers of appropriate size should be allocated and pointers to these buffers should be stored in the receive descriptor ring. Software initializes the Receive Descriptor Head (RDH) register and Receive Descriptor Tail (RDT) with the appropriate head and tail addresses. Head should point to the first valid receive descriptor in the descriptor ring and tail should point to one descriptor beyond the last valid descriptor in the descriptor ring.

Program the Receive Control (RCTL) register with appropriate values for desired operation to include the following:

- Set the receiver Enable (RCTL.EN) bit to 1b for normal operation. However, it is best to leave the Ethernet controller receive logic disabled (RCTL.EN = 0b) until after the receive descriptor ring has been initialized and software is ready to process received packets.
- Set the Long Packet Enable (RCTL.LPE) bit to 1b when processing packets greater than the standard Ethernet packet size. For example, this bit would be set to 1b when processing Jumbo Frames.
- Loopback Mode (RCTL.LBM) should be set to 00b for normal operation.
- Configure the Receive Descriptor Minimum Threshold Size (RCTL.RDMTS) bits to the desired value.
- Configure the Multicast Offset (RCTL.MO) bits to the desired value.
- Set the Broadcast Accept Mode (RCTL.BAM) bit to 1b allowing the hardware to accept broadcast packets.
- Configure the Receive Buffer Size (RCTL.BSIZE) bits to reflect the size of the receive buffers software provides to hardware. Also configure the Buffer Extension Size (RCTL.BSEX) bits if receive buffer needs to be larger than 2048 bytes.
- Set the Strip Ethernet CRC (RCTL.SECRC) bit if the desire is for hardware to strip the CRC prior to DMA-ing the receive packet to host memory.
- For the **82541xx** and **82547GI/EI**, program the Interrupt Mask Set/Read (IMS) register to enable any interrupt the driver wants to be notified of when the even occurs. Suggested bits include RXT, RXO, RXDMT, RXSEQ, and LSC. There is no immediate reason to enable the transmit interrupts. Plan to optimize interrupts later, including programming the interrupt moderation registers TIDV, TADV, RADV and IDTR.
- For the **82541xx** and **82547GI/EI**, if software uses the Receive Descriptor Minimum Threshold Interrupt, the Receive Delay Timer (RDTR) register should be initialized with the desired delay time.

14.5 Transmit Initialization

Allocate a region of memory for the transmit descriptor list. Software should insure this memory is aligned on a paragraph (16-byte) boundary. Program the Transmit Descriptor Base Address (TDBAL/TDBAH) register(s) with the address of the region. TDBAL is used for 32-bit addresses and both TDBAL and TDBAH are used for 64-bit addresses.

Set the Transmit Descriptor Length (TDLEN) register to the size (in bytes) of the descriptor ring. This register must be 128-byte aligned.

The Transmit Descriptor Head and Tail (TDH/TDT) registers are initialized (by hardware) to 0b after a power-on or a software initiated Ethernet controller reset. Software should write 0b to both these registers to ensure this.

Initialize the Transmit Control Register (TCTL) for desired operation to include the following:

- Set the Enable (TCTL.EN) bit to 1b for normal operation.
- Set the Pad Short Packets (TCTL.PSP) bit to 1b.

- Configure the Collision Threshold (TCTL.CT) to the desired value. Ethernet standard is 10h. This setting only has meaning in half duplex mode.
- Configure the Collision Distance (TCTL.COLD) to its expected value. For full duplex operation, this value should be set to 40h. For gigabit half duplex, this value should be set to 200h. For 10/100 half duplex, this value should be set to 40h.

Program the Transmit IPG (TIPG) register with the following decimal values to get the minimum legal Inter Packet Gap:

	Fiber	Copper	Fiber (82544GC/EI	Copper (82544GC/EI
IPGT	10	10	6	8
IPGR1	10	10	8[a]	8[a]
IPGR2	10	10	6[a]	6[a]

a. Applicable to the **82541xx** and **82547GI/EI**.

Note: IPGR1 and IPGR2 are not needed in full duplex, but are easier to always program to the values shown.

Table 14-1. Signal Descriptions

Signal	Ball	Name and Function
LOS / LINK	A10	**Loss of Signal (TBI) / Link Indication.** Loss of signal (high for lost signal) from the optical transceiver when LINK_MODE equals 11b; active high link indication from PHY in GMII/MII mode.
TX_DATA[9] / TX_ER TX_DATA[8] / TX_EN TX_DATA[7] TX_DATA[6] TX_DATA[5] TX_DATA[4] TX_DATA[3] TX_DATA[2] TX_DATA[1] TX_DATA[0]	C7 D7 E6 B5 E5 C5 E4 C4 D5 D4	**Transmit Data.** TBI: TX_DATA[9:0] for transmit data bus. GMII: TX_DATA[7:0] for transmit data bus. TX_ER forces propagation of transmit errors and is used for carrier extension. TX_EN is asserted to indicate transmission of data on the interface. MII: TX_DATA[3:0] for transmit data bus. TX_ER is not used. TX_EN is used for transmit enable signal.
GTX_CLK	C6	**Transmit Clock.** TBI: 125 MHz transmit clock. GMII: Operates at 125 MHz. MII: Undefined.
EWRAP	E10	**Enable Wrap.** TBI: EWRAP is low in normal operation. When it is high, the SerDes device is forced to transceiver loopback the serialized transmit data to the receiver. This pin is tri-stated during EEPROM read. In order to avoid a floating input in an external SerDes, a weak external pull-down should be connected to this pin. GMII / MII: Not used.
COL	E7	**Collision.** TBI: Undefined. GMII / MII: This signal indicates that a collision was detected on the medium by the PHY. This signal remains asserted while the collision persists. For half-duplex transceivers, this signal indicates simultaneous transmission and reception. This signal is ignored in full-duplex mode. Normal Mode: This signal must be connected to VSS except for test mode.

Signal	Ball	Name and Function
CRS	A6	**Carrier Sense.** TBI: Undefined. GMII / MII: This signal indicates traffic activity on the cable, either incoming or outgoing. This signal is driven by the PHY. CS is not required to transition synchronously with respect to the RX or TX clocks. This signal is ignored in full-duplex mode. Normal Mode: This signal must be connected to VSS except for test mode.
RX_DATA[9] / RX_ER RX_DATA[8] / RX_DV RX_DATA[7] RX_DATA[6] RX_DATA[5] RX_DATA[4] RX_DATA[3] RX_DATA[2] RX_DATA[1] RX_DATA[0]	A9 D10 B9 C9 D9 E9 E8 C8 A7 B7	**Receive Data.** TBI: RX_DATA[9:0] for receive data bus GMII: RX_DATA[7:0] for receive data bus. RX_ER signals a receive error. RX_DV is asserted to indicate data is valid on the interface. MII: RX_DATA[3:0] for receive data bus. RX_ER signals a receive error. RX_DV indicates data is valid on the interface.
RBC0 / RX_CLK	C11	**Receive Clock 0.** TBI: RBC0 is receive clock (62.5 Mbps). GMII: RX_CLK is receive clock (125 Mbps). MII: RX_CLK is receive clock for 100 Mbps operation (25 Mbps) and for 10 Mbps operation (2.5 Mbps).

14.5.1 Signal Interface

The external GMII/MII interface is similar in function to the interface used to communicate between the MAC and internal PHY. As with use of the internal PHY, the external GMII/MII interface supports 10/100/1000 Mbps operation, with both half- and full-duplex operation at 10/100 Mbps, and full-duplex operation at 1000 Mbps. Unlike the communication path to the internal PHY, the external interface does not provide certain additional control/status interfaces for automatic hardware link setup and/or power-management

Table 14-2lists the signals, functions, and pins used to provide this interface.

Table 14-2. Signal Functions

Signal	Function	Pin
GMII (1000 Mbps) Operations		
CRS	Carrier Sense	CRS
COL	Collision Detect	COL
TX_ER	Transmit Code Error	TX_DATA[9]/TX_ER
TX_EN	Transmit Enable	TX_DATA[8]/TX_EN
GTX_CLK	Transmit Data Clock (125 MHz)	GTX_CLK
TX_DATA	Transmit Data	TX_DATA[7:0]
RX_CLK	Receive Data Clock (125 MHz)	RBC0/RX_CLK
RX_DATA	Receive Data	RX_DATA[7:0]
RX_ER	Receive Error	RX_DATA[9]/RX_ER
RX_DV	Receive Data Valid	RX_DATA[8]/RX_DV
LINK	PHY Link Indication	LOS/LINK
MII (10/100 Mbps) Differences		
MTX_CLK	Transmit Data Clock (25/2.5 MHz)	RBC1/MTX_CLK
TX_DATA	Transmit Data	TX_DATA[3:0]
RX_CLK	Receive Data Clock (25/2.5 MHz)	RBC0/RX_CLK
RX_DATA	Receive Data	RX_DATA[3:0]

14.5.2 GMII/MII Features not Supported

Table 14-3 lists the signals and functions not provided by this interface.

Table 14-3. Signal Functions Not Supported

Signal	Function	Ramifications
MII Management Interface (PHY Register Access)		
MDC	Management Data Clock	No support/access to MII register set.
MDI/O	Management Data I/O	
Direct PHY Indications to MAC		
FDX	PHY-negotiated full/half duplex indication	Can limit use to specific known duplex setting.
SPD_IND	PHY-negotiated speed (10/100/1000 Mbps)	Can limit use to specific known speed or require use of auto-speed detection.

14.5.3 Avoiding GMII Test Mode(s)

Note that the Ethernet controller contains a set of test modes that use this interface for component manufacturing and/or diagnostic test. To avoid accidental engagement of unexpected test mode(s) when using the external GMII (or TBI), the TEST_GMII[2:0] test pins must remain de-asserted (low) and the TEST_DM_N pin must remain de-asserted (high).

14.5.4 MAC Configuration

The Ethernet controller MAC operates in a GMII/MII mode when operating with the internal PHY; this mode is similar to the GMII/MII mode of the standalone 82543 MAC components and others. In GMII/MII mode, the MAC operates assuming use of a GMII/MII interface communication, variable duplex & speed configuration (unless forced or auto-detected). For the Ethernet controller, to use this external interface as a GMII/MII interface and have the MAC operate in this GMII/MII Mode, the LINK_MODE must be set to 01b.

It is likely that the MAC might be required to be configured in a forced-duplex configuration, as no means is provided (either the MDI/O access or direct PHY-to-MAC signaling) of any duplex configuration that might be negotiated between the attached Ethernet controller/transceiver and its link partner.

The MAC can further be required to be configured in a forced-speed configuration, as no direct speed indication is available via the external interface (compared to the SPD_IND signals provided by the internal PHY). The Auto-Speed detection (ASD) can be potentially useful in automatically calculating and configuring a speed setting based in the interface signals that are provided.

The MAC is unable to provide any access to MII Management registers through the MDIC register, as no explicit MDI/O signals are included in this interface. However, it is possible that software-definable pins (SDP) can be capable of providing the necessary access capability.

14.5.5 Link Setup

The following examples are provided as suggestions for configuring common settings between the MAC and an Ethernet controller attached in the GMII/MII mode.

- MAC duplex and speed settings forced by software based on resolution of PHY
  (CTRL.FRCDPLX = 1b, CTRL.FRCSPD = 1b, CTRL.ASDE = don't care)
  - CTRL.FD Set by software based on reading PHY status register after PHY has autonegotiated a successful link-up.
  - CTRL.SLU Must be set to 1b by software to enable communications between MAC and PHY
  - CTRL.RFCE Must be set by S/W after reading flow control resolution from PHY registers

  CTRL.TFCE - Must be set by S/W after reading flow control resolution from PHY registers
  - CTRL.SPEED Set by software based on reading PHY status register after PHY has autonegotiated a successful link-up.
  - STATUS.FD Reflects the MAC forced duplex setting written to CTRL.FD
  - STATUS.LU Reflects link indication (LINK) from PHY qualified with CTRL.SLU (set to 1b)
  - STATUS.SPEED Reflects MAC forced speed setting written in CTRL.SPEED

- MAC duplex setting forced by software based on resolution of PHY; speed auto-detected by MAC
  (CTRL.FRCDPLX = 1b, CTRL.FRCSPD = 0b, CTRL.ASDE = 1b)
  - CTRL.FD Set by software based on reading PHY status register after PHY has autonegotiated a successful link-up.
  - CTRL.SLU Must be set to 1b by software to enable communications between MAC and PHY
  - CTRL.RFCE Must be set by S/W after reading flow control resolution from PHY registers

  CTRL.TFCE .... Must be set by S/W after reading flow control resolution from PHY registers
  - CTRL.SPEED Don't care; speed setting is calculated by the MAC based on signals from the PHY after PHY has autonegotiated a successful link-up
  - STATUS.FD Reflects the MAC forced duplex setting written to CTRL.FD
  - STATUS.LU Reflects link indication (LINK) from PHY qualified with CTRL.SLU (set to 1b)
  - STATUS.SPEED Reflects actual speed setting calculated by MAC ASD function

- MAC/PHY duplex and speed settings both forced by software (fully-forced link setup) (CTRL.FRCDPLX = 1b, CTRL.FRCSPD = 1b, CTRL.SLU = 1b)

  CTRL.FD ....Set by software to desired full/half duplex operation (must match duplex setting of PHY)
  CTRL.SLU.................Must be set to 1b by software to enable communications between MAC and PHY. PHY must also be forced/configured to indicate positive link indication (LINK) to the MAC
  CTRL.RFCE ....Must be set by S/W to desired flow-control operation (must match flow-control settings of PHY)
  CTRL.TFCE ....Must be set by S/W to desired flow-control operation (must match flow-control settings of PHY)
  CTRL.SPEED ....Set by software to desired link speed (must match speed setting of PHY)
  STATUS.FD ....Reflects the MAC duplex setting written by software to CTRL.FD
  STATUS.LU...............Reflects 1b (positive link indication LINK from PHY qualified with CTRL.SLU). Note: since both CTRL.SLU and the PHY link indication LINK are forced, this bit set does not GUARANTEE that operation of the link has been truly established.
  STATUS.SPEED........Reflects MAC forced speed setting written in CTRL.SPEED

Note: It is important to note that for the Ethernet controller's link indication (LINK) to be noted by the MAC, the MAC control bit CTRL.SLU must be set to 1b. Normal MAC/PHY speed and duplex configuration are based on observing events on this link indication from the Ethernet controller.

14.6 PHY Initialization (10/100/1000 Mb/s Copper Media)

Software needs to determine the PHY address at which the PHY actually resides. This number can be anywhere from 0 to 31.The PHY address is programmable. Board designers can then choose at what PHY address the PHY resides. Software needs to identify the PHY address so that the PHY can be accessed successfully.

To accomplish read and write access to any of the PHY registers, software must program the MDI Control Register (MDIC) with the appropriate data. A PHY is reset at power-up and is enabled to Auto-Negotiate by default. Typically in most environments, by the time the software driver is loaded, the Auto-Negotiation process has completed. However, the PHY might or might not advertise the appropriate capabilities desired by the design. In this instance, it is up to the software to insure that the PHY registers are set up properly to advertise the appropriate Ethernet controller capabilities. For example, by default the Ethernet controller advertises no flow control capabilities in its Auto-Negotiation Advertisement Register (MII Register 4). In order to advertise TX and/or RX Pause capabilities, this register must be modified and Auto-Negotiation re-started to advertise these capabilities to the link partner.

The MII Status Register (PHY Register 1) should be used to check link status.

Software can also force the speed/duplex of a PHY via MII/GMII register access. Note that forcing gigabit speed in a copper environment is not allowed per IEEE specification. Only 10/100 speed and duplex should be forced in the PHY.

Once link is achieved by the PHY, software is notified when a Link Status Change (LSC) interrupt is generated by the Ethernet controller. This only occurs if software enabled the LSC bit in the Interrupt Mask Set/Read (MS) Register.

14.7 Reset Operation

The following reset signals affect the Ethernet controller in different ways. RST# is the only external signal. Other reset events are asserted by performing slave writes to specific bits in the control registers.

Values indicated as "?" imply the default value is either unknown or is read from the EEPROM.

Note: In situations where the TX block is reset, the TX data lines are forced to all 0b's. This causes a substantial number of symbol errors to be detected by the link partner. In TBI mode (**82544GC/EI**)/internal SerDes (**82546GB/EB** and **82545GM/EM**), if the duration is long enough, the link partner can restart the Auto-Negotiation process by sending "break-link" (/C/ codes with the configuration register value set to all 0b's).

LAN_PWR_GOOD:

Deasserting LAN_PWR_GOOD resets all resettable registers in the Ethernet controller. The signal is level-sensitive, and the Ethernet controller is held in reset until LAN_PWR_GOOD is asserted. While asserted, all PCI signals are forced to a high impedance state.

General Registers:	Reset to power-on values.
Interrupt Registers:	Reset to power-on values.
Receive Registers:	Reset to power-on values (exceptions are the RAH/RAL, MTA, VFTA and RDBAH/RDBAL registers, which are not reset to any preset value. The valid bit of the RAH register is cleared).
Transmit Registers:	Reset to power-on values (exceptions are the TDBAH/TDBAL registers, which are not reset to any preset value).
Statistics Registers:	Reset to power-on values.
Wakeup Registers:	The WUC (except for the *PME_En* and *PME_Status* bits if AUX_POWER = 1b), WUFC, IPAV, and FFLT registers are reset to their default value.
Diagnostic Registers:	Reset to power-on values (exception is the PBM memory, which is not reset to any preset value).
PCI Config Space:	Context Lost; requires initialization.
PHY:	RST# is asserted to reset the PHY while LAN_PWR_GOOD is deasserted.

In addition, the Ethernet controller automatically reads certain values from the EEPROM and configures itself to use those EEPROM settings.

RST#:

When asserted, all PCI signals are forced to a high impedance state. Upon deassertion, the Ethernet controller's internal registers, excluding the following exceptions, are reset.

General Registers:	Reset to power-on values.
Interrupt Registers:	Reset to power-on values.
Receive Registers:	Reset to power-on values (exceptions are the RAH/RAL, MTA, VFTA and RDBAH/RDBAL registers, which are not reset to any preset value. The valid bit of the RAH register is cleared).
Transmit Registers:	Reset to power-on values (exceptions are the TDBAH/TDBAL, and TIPG registers, which is not reset to any preset value).
Statistics Registers:	Reset to power-on values.
Wakeup Registers:	The WUC (except for the *PME_En* and *PME_Status* bits if AUX_POWER = 1b), WUFC, IPAV, and FFLT registers are reset to their default value.
Diagnostic Registers:	Reset to power-on values (exception is the PBM memory, which is not reset to any preset value).
PCI Config Space:	Context Lost; requires initialization. If AUX_POWER = 1b then the *PME_En* and *PME_Status* bits of the Power Management Control/Status Register are preserved.
PHY:	RST# is asserted for 400 ns after deassertion of RST#.

Asserting RST# puts the Ethernet controller into the "Dr" Power Management state. See Section 6.3.1.1 for details on the power states, and Section 6.3.2.4 for reset related timing.

Deasserting RST# also causes the EEPROM to be re-read and the registers that get values from the EEPROM to be re-loaded.

Global Reset:

Bit 26 of the Device Control Register (CTRL.RST) performs an Ethernet controller reset of all functions to their equivalent power on state similar to asserting RST#, except that the state of the PCI core and PCI configuration space is not affected.

General Registers:	Reset to power-on values.
Interrupt Registers:	Reset to power-on values.
Receive Registers:	Reset to power-on values (exceptions are the RAH/RAL, MTA, VFTA and RDBAH/RDBAL registers, which are not reset to any preset value. The valid bit of the RAH register is cleared).
Transmit Registers:	Reset to power-on values (exceptions are the TDBAH/TDBAL, and TIPG registers).
Statistics Registers:	Reset to power-on values.
Wakeup Registers:	The WUC (except for the *PME_En* and *PME_Status* bits), WUFC, IPAV, and FFLT registers are reset to their default value.
Diagnostic Registers:	Reset to power-on values (exception is the PBM memory, which is not reset to any preset value).
PCI Config Space:	No Change.
PHY:	No effect.

Default values for certain bits of the Device Control Register must be read out of the EEPROM and appropriately set by software if an EEPROM is used.

Global Reset does NOT affect the direction of the software programmable pins.

Link_Reset:

When LRST (bit 3 of the Device Control register) is written as a logic 1b, the Ethernet controller is forced into a link reset state. When LRST is set to 1b the Auto-Negotiation function is disabled. The Auto-Negotiation logic is initiated/restarted when LRST is transitions to 0b. A link reset is only relevant in TBI mode/internal SerDes (not applicable to the **82540EP/EM**, **82541xx** and **82547GI/EI**).

The transmitter sends /C/ ordered_sets when LRST is asserted.

General Registers:	No change.
Interrupt Registers:	No change.
Receive Registers:	The RXCW register is cleared.
Transmit Registers:	No change.
Statistics Registers:	No change.
Wakeup Registers:	No change.
Diagnostic Registers:	No change.
PHY:	No effect.

EE_RST (Extended Device Control Register):

EEPROM reset bit. Initiates a “reset-like” event to the EEPROM function that causes the EEPROM to be read again. Control registers bits are not affected other than those read from the EEPROM.

PHY_RST (Device Control Register):

PHY reset bit in the Device Control Register. By writing a 1b to this bit the software forces the assertion of an internal signal output to reset the PHY device without accessing the PHY registers through the MII management interface (MDI/O & MDC). Internal states of the Ethernet controller are not impacted. To release the PHY reset the software must write a 0b to the bit.

In situations where the Ethernet controller is reset using the software reset CTRL.RST, the TX data lines are forced to all 0b’s. This causes a substantial number of symbol errors to be detected by the link partner. In TBI mode/internal SerDes, if the duration is long enough, the link partner can restart the Auto-Negotiation process by sending “break-link” (/C/ codes with the config register value set to all 0b’s).

Some registers mentioned above within the Ethernet controller are treated specially. The RAH/RAL[n], MTA[n], VFTA[n], WUPM[n], FFMT[n], FFVT[n], TDBAH/TDBAL, and RDBAH/RDBAL registers have no default value and if the functions associated with the registers are enabled they must be programmed by software. Once programmed, their value is preserved through all resets as long as power is applied to the Ethernet controller. Bit 31, the valid bit, of the RAH[n] registers is the exception and is reset with the LAN_PWR_GOOD and RST# and software reset (CTRL.RST) bit.

Driver accessible Wakeup Status registers are excluded from all resets except for LAN_PWR_GOOD. This includes:

- Wakeup Status Register.
- Wakeup Packet Length.
- Wakeup Packet Memory.

Finally, the "Wakeup Context" as defined in the PCI Bus Power Management Interface Specification is reset on LAN_PWR_GOOD, and is also reset on the deassertion of RST# if AUX_POWER = 0b. This includes:

- PME_En bit of the Power Management Control/Status Register (PMCSR).
- PME_Status bit of the Power Management Control/Status Register (PMCSR).

The shadow copies of these bits in the Wakeup Control Register are treated identically.

14.8 Initialization of Statistics

Statistics registers are hardware-initialized to values as detailed in each particular register's description. The initialization of these registers begins upon transition to $D0_{active}$ power state (when internal registers become accessible, as enabled by setting the Memory Access Enable of the PCI Command register), and is guaranteed to be completed within 1 µs of this transition. Access to statistics registers prior to this interval can return indeterminate values. Given typical system boot times and the software driver's Ethernet controller initialization routines, no initialization of these registers through software should be necessary.

Diagnostics and Testability 15

15.1 Diagnostics

This section explains the registers provided for diagnostic access.

These registers enable system level integration and debugging, including the ability to access all internal memories. This information is often critical in determining failure modes and in developing software workarounds.

At a diagnostic level, all of the major internal data structures visible to and controllable by software, including all of the FIFO space. However, interlocks are not provided for any operations, so diagnostic accesses need to be performed under very controlled circumstances.

15.1.1 FIFO State

The internal data FIFO pointers are visible through the head and tail diagnostic data FIFO registers (see Section 13.8). Diagnostics software uses these FIFO pointers to confirm correct operation and to directly write packets into, or directly read out of, the FIFO.

These registers are available for diagnostic purposes only and should not be written during normal operation.

15.1.2 FIFO Data

All of the FIFO data is visible through the PBM register. Locations can be accessed as 32-bit or 64-bit words. Refer to Section 13.8.11 for details.

15.1.3 Loopback

One loopback mode is provided in the Ethernet controller to assist with system and device debug. This loopback mode is enabled via RCTL.LBM control bits. The Ethernet controller must be operating in full-duplex mode for loopback.

15.1.3.1 Internal Loopback

This loopback mode internally loops back the transmit to receive path in the PHY, exercising the internal GMII/MII bus. Programming both MAC and PHY is required. Following is the flow:

```
/* Auto-MDI/MDIX Off */

e1000_write_phy_reg(16, 0x0808);

/* reset to update Auto-MDI/MDIX */

e1000_write_phy_reg(0, 0x9140);

/* autoneg off */

e1000_write_phy_reg(0, 0x8140);

/* force 1000, set loopback */

e1000_write_phy_reg(0, 0x4140);

/* Now set up the MAC to the same speed/duplex as the PHY. */

ctrl_reg = E1000_READ_REG(CTRL);

ctrl_reg &= ~E1000_CTRL_SPD_SEL;            /* Clear the speed sel bits */

ctrl_reg |= (E1000_CTRL_FRCSPD |            /* Set the Force Speed Bit */

            E1000_CTRL_FRCDPLX |            /* Set the Force Duplex Bit */

            E1000_CTRL_SPD_1000 |           /* Force Speed to 1000 */

            E1000_CTRL_FD);                 /* Force Duplex to FULL */

/* Set the ILOS bit on the fiber Nic is half duplex link is detected. */

stat_reg = E1000_READ_REG(STATUS);

if((stat_reg & E1000_STATUS_FD) = 0)

      ctrl_reg |= (E1000_CTRL_ILOS | E1000_CTRL_SLU);

E1000_WRITE_REG(CTRL, ctrl_reg);
```

15.2 Testability

The Ethernet controller uses full Boundary Scan/IEEE 1149.1 JTAG standard test methods. The TAP controller supports EXTEST, SAMPLE/PRELOAD, IDCODE, USERCODE, and BYPASS instructions.

15.2.1 EXTEST Instruction

This instruction allows testing of off-chip circuitry and board level interconnections. Data is typically loaded onto the latched parallel outputs of the boundary-scan shift register stages using the SAMPLE/PRELOAD instruction prior to selection of the EXTEST instruction.

15.2.2 SAMPLE/PRELOAD Instruction

This mandatory instruction allows a snapshot of the normal operation of the component to be taken and examined. It also allows data values to be loaded onto the latched parallel outputs of the boundary-scan shift register prior to selection of the other boundary-scan test instructions.

15.2.3 IDCODE Instruction

The IDCODE instruction provides information on the base component. When an Ethernet controller identification register is included in a component design, the IDCODE instruction is forced into the instruction register's parallel output latches.

For example, the 82546EB controller's ID is determined and derived from the manufacturer as follows:

Component Product Code	Ver	V	Product	Gen	Model	Manf ID	1	ID Code (hex)
82546EB	0001	1	001001	0010	00100	00000001001	1	19244013

15.2.4 BYPASS Instruction

This instruction is the only instruction defined by the standard that causes operation of the bypass register. The bypass register contains a single-shift register stage and is used to provide a minimum length serial path between the TDI and TDO pins of a component when no test operation of that component is required. This allows more rapid movement of test data to and from other components on a board that are required to perform test operations.

Note: This page intentionally left blank.

Appendix (Changes From 82544EI/82544GC) A

A.1 Introduction

This section describes the new features that have been added to the PCI/PCI-X Family of Gigabit Ethernet Controllers from its predecessor, the 82544EI/82544GC and highlights its registers that have been changed.

A.2 New Features

Following is a list of the new features in the Ethernet controller, along with sections in this manual that describe these features in detail:

- **Integrated dual-port solution**. The **82546GB/EB** architecture includes two instances of both the MAC and PHY. The Ethernet controller contains a single PCI/PCI-X interface so that each of the logical LAN devices appear as a distinct PCI/PCI-X bus device (see Chapter 12).
- **IPv6 Support**. The Ethernet controller supports IP/TCP/UDP receive checksum offload for IPv6 packets, IPv6 wakeup filters, and IPv6 TCP segmentation (see Chapter 3).
- **Improved Interrupts**. The Ethernet controller has the following new interrupt generation features to enhance driver performance:
  - — Packet timers and absolute delay timers for transmit and receive (see Section 3.2.7).
  - — Short packet detection interrupts (see Section 3.2.7).
  - — Transmit descriptor low interrupts (see Section 3.4.3).
  - — Interrupt throttling control to limit maximum interrupt rate (see Section 13.4.8).
  - — Acknowledge interrupts by writing 1b's to the ICR (see Section 13.4.17).
- **EEPROM access**. Because the **82546GB/EB** has two MAC/PHY instances that could potentially access the EEPROM at the same time, a semaphore has been added to gate access. In addition, the EERD now allows easy EEPROM access (see Section 12.3.2).
- **Manageability**. The Ethernet controller introduces manageability for ASF 1.0 and AOL 2.0.
- **Configurable LED**. The Ethernet controller enables software to customize LED displays (see Chapter 10).
- **Power Management**. The internal copper PHY features 10/100/1000-BaseT signalling and is capable of performing intelligent power-management based on both the system power-state and LAN energy-detection (detection of unplugged cables). See Chapter 6.

A.3 Register Changes

Table A-1 lists the registers that have been added or changed in the Ethernet controller.

Table A-1. Register Changes

Register	Offset
CTRL	00000h
STATUS	00008h
EEC	00010h
EERD	00014h
CTRL_EXT	00018h
LEDCTL	00E00h
ICR	000C0h
ITR	000C4h
ICS	000C8h
IMS	000D0h
IMC	000D8h
RDTR	02820h
RXDCTL	02828h
RADV	0282Ch
RSRPD	02C00h
RXCSUM	05000h
TXDCTL	03828h
TADV	0382Ch
TPT	040D4h
PTC64	040D8h
PTC127	040DCh
PTC255	040E0h
PTC511	040E4h
PTC1023	040E8h
PTC1522	040ECh
MPTC	040F0h
BPTC	040F4h

Appendix (82540EP/EM and 82545GM/EM Differences) B

B.1 Introduction

This section describes the differences between the **82546GB/EB**, the **82540EP/EM** and the **82545GM/EM**. All three of these Ethernet controllers come from the same family so their register sets are essentially the same. The sections that follow describe the differences between the **82546GB/EB** and the **82540EP/EM** or **82545GM/EM**, and resulting register differences and developer impact.

B.2 82540EP/EM Differences

Below are the differences between the **82540EP/EM** and the **82546GM/EM**:

- **Serial FLASH interface.** The **82540EP/EM** does not support a parallel FLASH interface.
- **No TBI/internal SerDes interface.** The **82540EP/EM** provides internal GMII / MII interfaces only
- **Single-port functionality.** The dual-port functionality of the **82546GB/EB** is not included.
- **32-bit PCI support only.** The **82540EP/EM** does not support 64-bit PCI or PCI-X.
- **Internal SerDes.** The **82540EP/EM** does not support internal SerDes.

The impact to registers and the developer are outlined in the following sections.

B.2.1 Serial FLASH Interface

The **82540EP/EM** provides an external interface to a serial Flash or Boot EPROM device. Hardware implements a serial command set compatible with the Atmel AT25-series devices in the 512 Kb (64 KB) thru 1024 Kb (128 KB) sizes. The size of the FLASH used with the 82540EP/EM should be encoded into bits in EEPROM to configure the amount of address space required when mapped. All accesses to this device are controlled by hardware and are accessible to software as normal PCI reads or writes to the FLASH memory mapping range. The FLASH and Expansion ROM BARs are reconfigured based on these EEPROM settings.

Note: The **82540EP/EM** serial FLASH controller supports reads from programmed FLASH devices, and writes to erased FLASH devices. Chip and sector-erase commands are not supported. It is recommended that FLASH devices be socketed to enable removal for re-programming if necessary.

Note: Though the **82540EP/EM** supports devices with up to 512 KB of memory, smaller devices may also be used. Accesses to memory beyond the FLASH device size results in access wrapping as only the lower address bits are utilized by the FLASH.

The **82540EP/EM** does not provide an interface for performing an "Erase" operation to the serial Flash device. Flash write operations must be performed to an initialized or pre-erased Flash device. If in-circuit erase is required, an external source (such as hardware's software-definable pins) can be used to drive the Flash pins to perform the erase operation.

B.2.2 No TBI/Internal SerDes Interface

The **82540EP/EM** does not support a TBI/internal Serdes interface. As a result, all TBI/internal SerDes-related registers (for example, RXCW) should not be used.

B.2.3 Single-Port Functionality

The **82540EP/EM** and the **82545GM/EM** do not have the dual-port functionality of the **82546GB/EB**. As a result, the **82540EP/EM** and the **82545GM/EM** appear as single PCI bus devices.

B.2.4 32-Bit PCI Support

No developer impact.