PCIe* GbE Controllers Open Source Software Developer's Manual

631xESB/632xESB, 82563EB/82564EB, 82571EB/82572EI & 82573E/ 82573V/82573L

316080-009
Revision 1.8

Revision History

Date	Version	Comments
June 2008	1.8	Updated Sections 13.3.43 and 13.3.45 (added text stating to use the Interrupt Throttling Register (ITR) instead of registers RDTR and RADV for applications requiring an interrupt moderation mechanism). Removed the table notes from Sections 13.5.4 through 13.5.6.
May 2008	1.7	Updated Table 13.56 (bit 6 description). Added the Adaptive IFS Throttle register details and bit assignments.
April 2008	1.6	Updated Section 4.7.4 (changed the Device ID for the 82573E to 108Ch and the Device ID for the 82573V to 108Bh).
Mar 2008	1.5	Updated section 14.9; PHY reset procedure for the 82573.
Jan 2008	1.4	Updated Table 13-35 (bit 15 description). Updated Table 13.56 (bit 1 description). Updated section 5.7.8. Updated Table 13.23.
Oct 2007	1.3	Updated sections 3.2.5, 3.2.12, and 13.4.3. Updated EEPROM words 10h/20h and 21h. Added section 3.2.8 "Using Extended Descriptors".
Sept 2007	1.2	Updated section 13.3.8
April 2007	1.1	Updated: Section 5 default values and bit descriptions for words 03h, 0Ah, 17h, 1Bh, 1Eh, and 21h. TXDCTL, TXDCTL1, TARC0, and TARC1 bit descriptions. Tables 4-5, 5-2, 13-148, and 14-1.
Dec 2006	1.0	Initial public release.

Note: This page 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 the PCI Express* (PCIe*) Family of Gigabit Ethernet Controllers (**82571EB/82572EI**, **631xESB/632xESB, 82563EB/82564EB**, and **82573E/82573V/82573L**). Throughout this manual references are made to the PCIe* Family of Gigabit Ethernet Controllers or Ethernet controllers. Unless specifically noted, these references apply to all the Gigabit Ethernet controllers listed above.

1.2 Overview

The PCIe* 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 PCIe* architecture (Revision 1.0a).

The PCIe* Family of Gigabit Ethernet Controllers handle all IEEE 802.3 receive and transmit MAC functions. They contain fully integrated physical-layer circuitry[1] for 1000Base-T, 100Base-TX, and 10Base-T applications (IEEE 802.3, 802.3u, and 802.3ab) as well as on-chip Serializer/Deserializer (SerDes)[2] functionality that fully complies with IEEE 802.3z Physical Coding sub-Layer (PCS).

631xESB/632xESB, **82571EB**, and **82563EB** 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 PCIe* devices containing two identically-functioning Ethernet controllers. See Section 12 for details.

1.3 Ethernet Controller Features

This section describes the features of the PCIe* Family of Gigabit Ethernet Controllers.

1.3.1 PCI Features

- x4 PCIe* interface on an MCH device (x1 PCIe* interface on an I/O Controller Hub 7 (ICH7) or Memory Controller Hub (MCH) device for the **82573E/82573V/82573L**)
- Peak bandwidth 2 GB/s in each direction per PCIe* lane
- PCIe* Power Management
- High bandwidth density per pin

1. For **631xESB/632xESB**, this circuitry is provided by a separate PHY device (**82563EB/82564EB**).
2. SerDes functionality is not applicable to the **82573E/82573V/82573L**.

1.3.2 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 10/100/1000Base-T 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
- 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 Digital Signal Processor (DSP) architecture implementation (not applicable to the **631xESB/632xESB**)
  - Digital adaptive equalization and crosstalk
  - Echo and crosstalk cancellation
- Integrated dual-port solution comprised of two distinct MAC/PHY instances (**631xESB/632xESB**, **82571EB**, and **82563EB**)
- Provides on-chip IEEE 802.3z PCS SerDes functionality (**631xESB/632xESB**, **82571EB/82572EI**, and **82563EB/82564EB**)

1.3.3 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
  - 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:
  - IP/TCP and IP/UDP receive checksum offload
  - Wake up filters
  - TCP segmentation

1.3.4 Additional Performance Features

- Programmable 256 B to 16 KB host memory receive buffers (1 KB to 16 KB for the **631xESB/632xESB**).
- Programmable cache line size from 16 B to 128 B for efficient usage of PCIe* bandwidth
- Configurable transmit and receive FIFO buffer:
  - 48 KB for the **82571EB/82572EI** and the **631xESB/632xESB**
  - 32 KB for the **82573E/82573V/82573L**
- Descriptor ring management hardware for transmit and receive. Optimized descriptor fetching and write-back mechanisms for efficient system memory and PCIe* bandwidth usage
- Supports reception and transmission of packets with length up to 9014 bytes (9234 byes for the **82571EB/82572EI/631xESB/632xESB**)
- New intelligent interrupt generation features to enhance driver performance:
  - 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

1.3.5 Manageability Features

- Manageability support for Alert Standard Format (ASF) ASF 1.0 or ASF 2.0 and Alert on LAN 2.0 by way of a System Management Bus (SMB) SMB 2.0 interface and either:
  - Total Cost of Ownership (TCO) mode SMBus-based management packet transmit / receive support
  - Internal ASF-compliant TCO controller
- Fast Management Link (FML) support (not applicable to the **82573E/82573V/82573L**)
- Intel® Active Management Technology (Intel® AMT) support (**82573E** only)

1.3.6 Additional Ethernet Controller Features

- Dual transmit and receive queues for Receive Side Scaling (RSS), QoS and other programmable usage models
- For the **631xESB/632xESB** only, dual GLCI interface to two external 1000BASE-T PHYs (**82563EB/82564EB**)
- Implements ACPI register set and power down functionality supporting D0 and D3 states
- Supports Wake on LAN (WoL)
- Provides four-wire serial EEPROM interface for loading product configuration information (four-wire Flash/EEPROM for the **82573E/82573V/82573L**)
- Provides external parallel interface for up to 512 KB of FLASH memory for support of Pre-Boot Execution Environment (PXE)
- 2 (per function) general purpose user mode pins (**82571EB/82572EI** and the **631xESB/632xESB**)
- 2 additional (per function) general purpose user mode pins; configurable as general purpose interrupts (**631xESB/632xESB**)
- Provides Activity and Link LED indications
- Supports little-endian byte ordering for 32- and 64-bit systems
- Provides loopback capabilities under SerDes (**631xESB/632xESB**, **82571EB/82572EI**, and **82563EB/82564EB**) for 10/100/1000Base-T modes of operation
- Provides IEEE JTAG boundary scan support
- Four programmable LED outputs (three for the **82573E/82573V/82573L**)
  - For **631xESB/632xESB**, **82571EB**, and **82563EB**, four programmable LED outputs for each port
- Detection and improved power-management with LAN cable unconnected

1.3.7 Technology Features

- Packaging:
  - 256-pin FC-BGA (**82571EB/82572EI**)
  - 196-pin TF-BGA (**82573E/82573V/82573L**)
  - 100-pin TQFL with ExposedPad* (**82563EB/82564EB**)

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 product features.

1.4.1 Register and Bit References

This document refers to device register names with all capital letters. To refer to a specific bit in a register the convention REGISTER.BIT is used. For example CTRL.FD refers to the *Full Duplex Mode* 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 Express* Specification, Revision 1.0a
- Low Pin Count Interface Specification, Revision 1.1 (LPC)
- PCI Local Bus Specification, Revision 2.3, PCI Local Bus Special Interest Group.
- System Management Bus Specification, Version 2.0 (SMBus)
- PCI Express* Card Electromechanical Specification, Revision 1.0a
- Advanced Configuration and Power Interface (ACPI) Specification, Revision 2.0
- PCI Bus Power Management Interface Specification, Revision 1.1
- PCIe iBIST Architecture Specification, Revision 1.0

1.6 Memory Alignment Terminology

Some PCIe* Family of Gigabit Ethernet Controllers' 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 PCIe* 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 PCIe* 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 PIO 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 PCIe* efficiency and performance
3. Provide a simple software interface for basic operations.
4. Provide a highly configurable design that can be used effectively in different environments.

2.2 External Architecture

Figure 2-1 shows the external interfaces to the **82571EB/82572EI**.

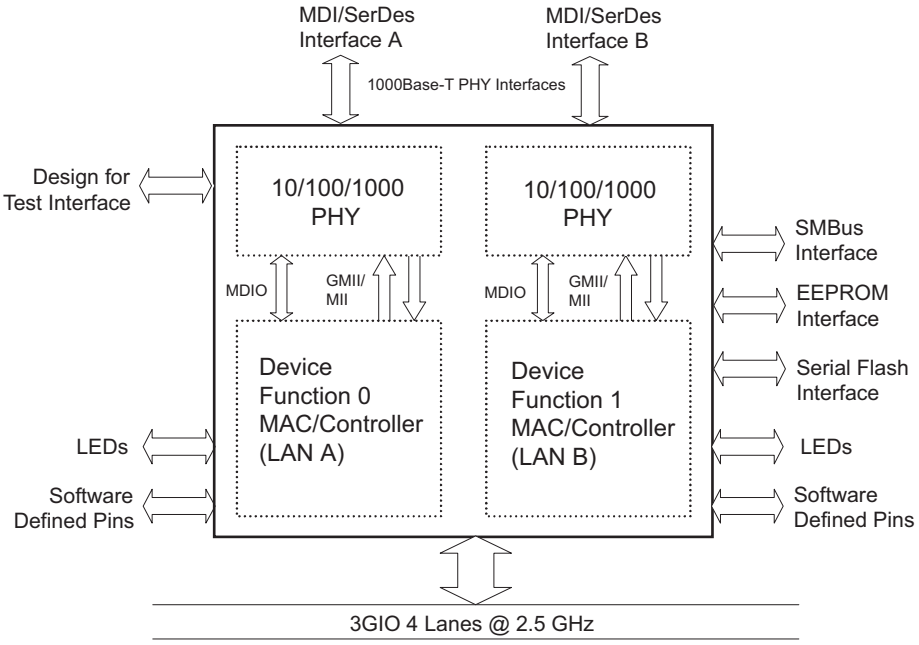


Figure 2-1. 82571EB/82572EI External Interface

Figure 2-2 shows the external interfaces to the **631xESB/632xESB**.

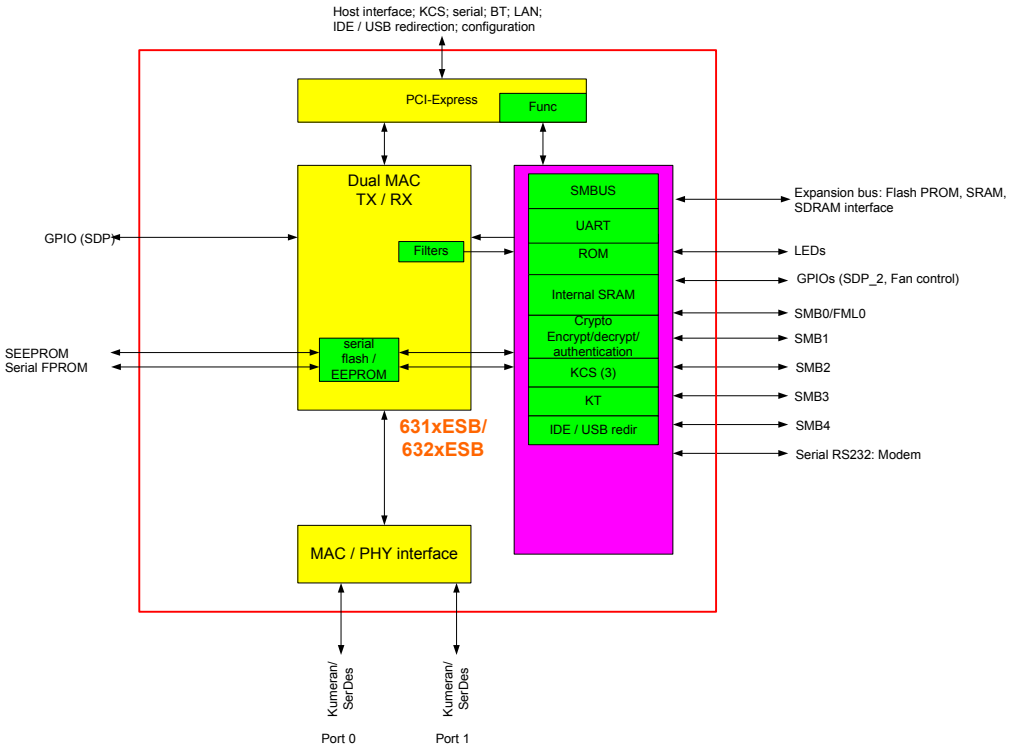


Figure 2-2. 631xESB/632xESB External Interface

Figure 2-3 shows the external interfaces to the **82573E/82573V/82573L**.

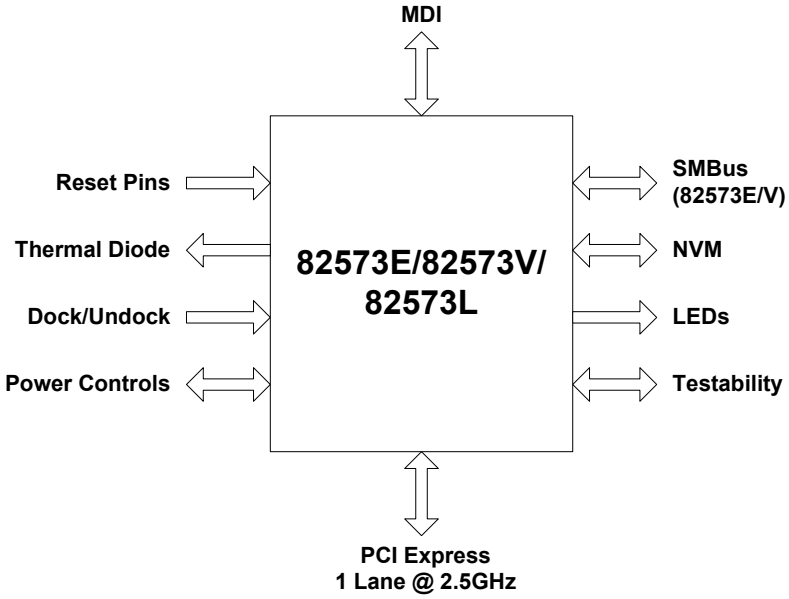


Figure 2-3. 82573E/82573V/82573L External Interface

2.3 Microarchitecture

Compared to its predecessors, the PCIe* Family of Gigabit Ethernet Controllers' 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.

The PCIe* Family of Gigabit Ethernet Controllers also introduce a new PCIe* host interface to the architecture and provide advanced performance features and improved power-management capability

For the **631xESB/632xESB**, **82571EB**, and **82563EB**, this new functionality is packaged in an integrated dual-port combination. The architecture includes two instances of both the MAC and PHY.

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

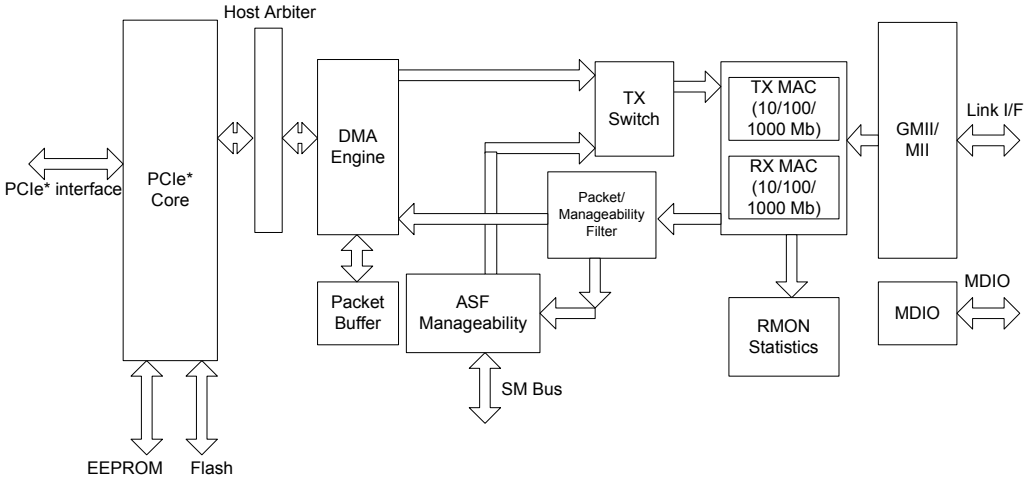


Figure 2-4. Internal Architecture Block Diagram

2.3.1 Integrated 10/100/1000 Mb/s PHY

The Ethernet controllers contain integrated 10/100/1000 Mb/s-capable Copper PHY's. Each of these PHY's communicate with its MAC controllers using a standard 10/100/1000Base-T interface internal to the component to transfer transmit and receive data. For the **631xESB/632xESB** and **82563EB/82564EB**, a GLCI interface is used to transmit and receive data between the MAC and PHY. A standard MDIO interface, accessible to software via MAC control registers, is also used to configure and monitor each PHY operation.

2.3.2 System Interface

The Ethernet controllers provide 4 lanes (1 lane for the **82573E/82573V/82573L**) of PCIe* bus interface working at 2.5 GHz each, this should provide sufficient bandwidth to support sustained dual port of 1000 Mb/s transfer rates. 48 KB (32 KB for the **82573E/82573V/82573L**) of on-chip buffering mitigates instantaneous receive bandwidth demands and eliminates transmit under-runs by buffering the entire outgoing packet prior to transmission.

2.3.3 EEPROM Interface

The PCIe* 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.

2.3.4 Flash Memory Interface

The Ethernet controller provides an external serial interface to a FLASH device. Accesses to the FLASH are controlled by the Ethernet controller and are accessible to software as normal PCIe* reads or writes to the FLASH memory mapping area. The Ethernet controller supports FLASH devices with up to 512 KB of memory (2 MB for the **82573E/82573V/82573L**).

Note: The **82573E/82573V/82573L** supports both FLASH memory and EEPROM; however, only one device can be connected at a time (not both).

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: Descriptor accesses are not byte swapped.

The following example illustrates data-byte ordering. 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 PCIe* 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.

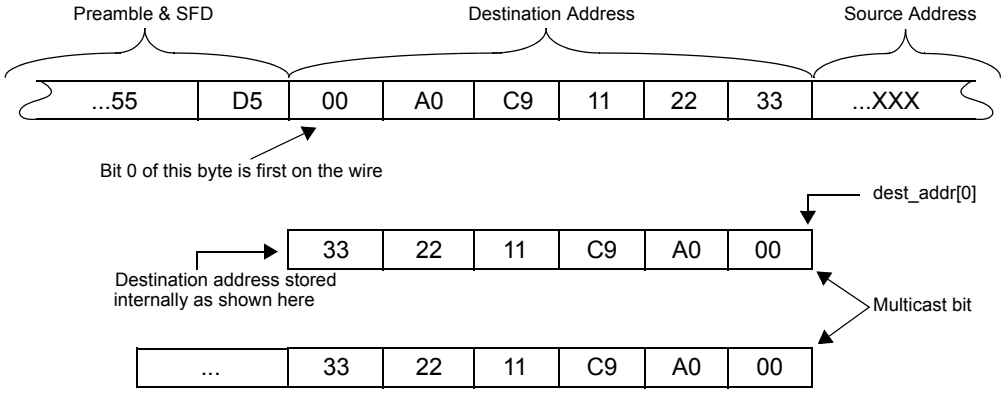


Figure 2-5. Example of Address Byte Ordering

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	A0	C9	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_A0_C9_11_22_33h, where the first byte (00_) is the first byte on the wire, with bit 0 of that byte transmitted first.

2.6 Interrupt Control and Tuning

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.
- Correlate between related bits in different registers. For example, CPU vector and Interrupt Control register (ICR)
- Minimize the number of interrupts needed relative to work accomplished.
- Minimize the processing overhead associated with each interrupt.

The interrupt logic consists of the following interrupt registers. More detail about these registers is given in sections 13.3.27 through 13.3.31.

- 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 (a few cause bits are also cleared when the CPU vector is 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 PCIe* only if the interrupt cause bit is a 1b, and the corresponding interrupt mask bit is a 1b. Software blocks submission of the Message Signaled Interrupt (MSI) 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.

- Interrupt Throttling Register

  The frequency of interrupts from the Ethernet controller can be reduced when inter-interrupt interval value is non-zero. The Ethernet controller asserts pending interrupts only at regularly scheduled intervals. When inter-interrupt interval value is zero, the Ethernet controller immediately asserts pending interrupts.

Two actions minimize the number of interrupts:

1. Reducing the frequency of all interrupts
2. Accepting multiple receive packets before signaling an interrupt.

One interrupt register consolidates all interrupt information eliminating the need for multiple accesses. However, a software driver might choose to have finer details by also reading the CPU Vector Register.

Note: Ethernet controllers support Message Signaled Interrupts per the PCI 2.2, 2.3, and PCIe* specifications. See Section 4.7.6.1 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. Features include:

- Jumbo frame support
- Receive and transmit checksum offloading
- TCP segmentation
- Receive fragmented UDP checksum offload

These features are briefly outlined in the following sections.

2.7.1 Jumbo Frame Support[1]

Ethernet controllers support jumbo frames to increase performance and decrease CPU utilization. By default, Ethernet controllers might receive packets with a maximum size of 1522 bytes. If large frame reception is enabled by the Receive Control Register (RCTL), Ethernet controllers support jumbo packet reception of up to 9014 bytes (9234 byes for the **82571EB/82572EI/631xESB/632xESB**). On the transmit size, jumbo packets are always supported by the Ethernet controllers. It is the responsibility of the software driver to initiate jumbo packets only when it is configured to do so.

Note: For the **82573L**, the packet buffer size must be at least the size of the maximum packet size. Furthermore, in order to reach wire speed the transmit packet buffer size must be at least twice the size of the maximum packet size. The total transmit and receive packet buffer size is 32 KB. Therefore, in case wire speed performance is important, the programmer should enable 9 KB packets. In the case of 9 KB, the transmit buffer size could be set to 18 KB while the receive buffer size could be set to 14 KB.

1. Not applicable to the **82573E/82573V**.

2.7.2 Receive and Transmit 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.

Note: IPv6 headers do not have a checksum.

For transmits where an Ethernet controller is doing non-TCP segmentation, every transmitted Ethernet packet can have two checksums calculated and inserted by the Ethernet controller. Typically these would be the IPv4 and either TCP or UDP checksums. The software device driver specifies which portions of the packet are included in the checksum calculations, and where the calculated values are inserted, via descriptor(s). Refer to Section 3.4.5 for details.

For receives, the hardware recognizes the packet type and performs the checksum calculations as well as error checking automatically. Checksum and error information is provided to software via the receive descriptor(s). Refer to Section 3.2.12 for details.

2.7.3 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 (IPv4 or IPv6), 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 3.6 for details.

2.7.4 Receive Fragmented UDP Checksum Offloading

Ethernet controllers provide the ability to offload inbound fragmented UDP packet reassembly. The Ethernet controllers provide the partial checksum calculation for each incoming UDP fragment so that the software device driver is required to sum the partial checksum words for each fragment to produce the complete checksum. The fragmented UDP checksum offload is provided to IPv4 packets. For more details see Section 3.

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.

2.9 Multiple Transmit Queues[1]

Ethernet controllers support two transmit descriptor rings. Each ring functionality is implemented according to the description in Section 3. The priority between the queues can be set and specified in the memory space.

2.9.1 Quality of Service (QoS)

Supporting 802.1p, while classifying packets into different priority queues. When adding a priority value between queues it might be possible to support QoS by classifying packets to queues based on their priority.

2.9.2 Resource Locking Prevention

TARC0 and TARC1 registers enable the tuning of the arbitration parameters, which control the transmission of both transmission queues (see Section 13).

2.10 iSCSI Boot (631xESB/632xESB)

This feature consists of adding an iSCSI class code to potentially replace the LAN class code of the ports. When the system boots, the BIOS detects this class code and runs the appropriate iSCSI software.

Ethernet controllers read two new control bits out of EEPROM word 29h (see Section 5). Each bit affects its respective LAN class code value. If the bit is set to 0b (this is the current value of the unused bits), the LAN class code remains as is (value = 020000 = LAN). If the bit is set to 1b, the LAN class code becomes a SCSI class code (value = 010000 = SCSI). Having this functionality allows us to change one port (or two in specific applications) to a SCSI device type and loads an iSCSI Miniport driver for that port. As a result, the port(s) function as iSCSI HBA. Default values for these fields in the EEPROM for both ports remain as a network class type.

In this case, the MAC address and the IP address of the port are used by the iSCSI function.

1. Not applicable to the **82573E/82573V/82573L**.

Note: This page intentionally left blank.

Receive and Transmit Description 3

3.1 Introduction

This section describes the packet reception, packet transmission, transmit and receive descriptor ring structures, TCP segmentation, and transmit checksum offloading for the PCIe* Family of Gigabit Ethernet Controllers.

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.

Note: The maximum supported received packet size is 9014 bytes (9234 byes for the **82571EB/82572EI/631xESB/632xESB**).

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 the packet 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.

Note: The software device driver can use only 15 entries (entries 0-14) of the exact address table. Entry 15 should be kept untouched by the software device driver. It can be used only by the manageability's firmware or external TCO controller.

- ***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 to determine 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.

Note: When a promiscuous bit is set and a multicast packet is received, the PIF bit of the packet status is not set.

- ***VLAN*** — Receive all VLAN 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. A good packet is defined as one without a:

- CRC error
- Symbol error
- Sequence error
- Length error
- Alignment error
- Receive error being signaled by the PHY

However, if the store–bad–packet bit is set in the Receive 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).

Note: All packets must have a valid SFD (RX_DV with no RX_ER in the 10/100/1000Base-T interface) in order to be recognized by the Ethernet controller). As a result, a CRC error is not counted in frames lacking an SFD.

For the **82571EB/82572EI**, a length error occurs if an incoming packet passes the filter criteria but is undersized or oversized.The **82571EB/82572EI** does not check the Ethernet length field to verify that the length of the packets matches the value in the length field. Packets with incorrect length field values are not discarded or reported as required by Section 4.3.2 of IEEE 802.3 2002.

If manageability is enabled and if RMCP is enabled then ARP request packets can be directed over the SMBus (or processed internally by the ASF controller for the **82571EB/82572EI**) rather than delivered to host memory.

3.2.1.1 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.

3.2.1.2 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.

3.2.1.3 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.

3.2.1.4 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.
  - For the **631xESB/632xESB**, the RFCTL.Ipv6_DIS bit must be cleared for this filter to pass.

3.2.1.5 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.

3.2.2 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
- FLXBUF x 1024 B while FLXBUF = 1, 2, 3, . . . 15

Buffer size is selected by bit settings in the Receive Control register (RCTL.BSIZE & RCTL.BSEX, RCTL.DTYP, RCTL, and FLXBUF). See Section 13.3.33 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 Queue Structure

Figure 3-1 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.

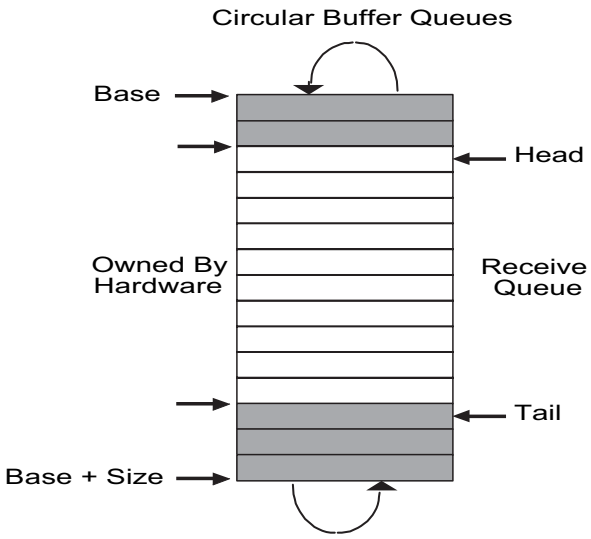


Figure 3-1. Receive Descriptor Ring Structure

When the Ethernet controller is configured to operate using the packet split feature, the descriptor tail needs to be incremented by software by two for every descriptor ready in memory (packet split descriptors are 32 bytes while regular descriptors are 16 bytes).

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 (RDBA0 and RDBA1)

  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. RDBA0 contains the lower 32-bits; RDBA1 contains the upper 32 bits. Hardware ignores the lower 4 bits in RDBA0.

- Receive Descriptor Length registers (RDLEN0 and RDLEN1)

  These registers determine 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 registers (RDH0 and RDH1)

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

- Receive Descriptor Tail registers (RDT0 and RDT1)

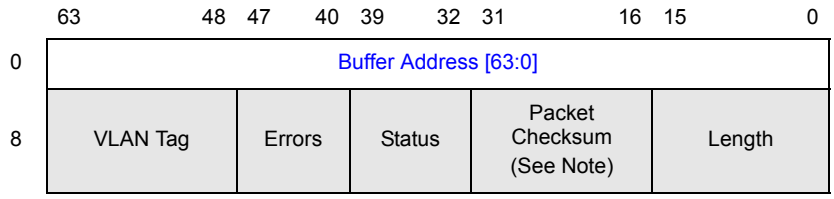
  These registers hold a value that is an offset from the base, and identifies the location beyond the last descriptor hardware can process. Note that tail should still point to an area in the descriptor ring (somewhere between RDBA and RDBA + RDLEN). This is because tail points to the location where software 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, but is necessary for performing an in–memory scan.

3.2.4 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. If the RFCTL.EXSTEN bit is cleared and the RCTL.DTYP = 00b, the Ethernet controller uses the Legacy Rx Descriptor as shown in Table 3-1. The shaded areas indicate fields that are modified by hardware upon packet reception.

Table 3-1. Receive Descriptor (RDESC) Layout



Note: The checksum indicated here is the unadjusted "16 bit ones complement" of the packet. A software assist might 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); see Section 13.3.33 for more information. 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 can modify the starting offset for the packet checksum calculation by means of the Receive Checksum Control register (RXCSUM). This register is described in Section 13.3.55. To verify the TCP/UDP 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.

For packets with VLAN header the packet checksum includes the header if VLAN stripping is not enabled by the CTRL.VME bit. If VLAN header strip is enabled, the packet checksum and the starting offset of the packet checksum exclude the VLAN header.

3.2.4.1 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.

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

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.

Receive Descriptor Status Bits	Description
IPCS (bit 6)	IPv4 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 = Did not perform IP checksum 1b = Performed 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.
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 = Did not perform TCP/UDP checksum 1b = Performed 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.
UDPCS (bit 4)	UDP Checksum Calculated on Packet
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.
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 3-10 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 IPCS or TCPCS bits set. IPv6 packets do not have the IPCS bit set, but might have the TCPCS bit set if the Ethernet controller recognized the TCP or UDP packet.

3.2.4.2 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

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 internal SerDes mode refers to the reception of a /V/ code (see Section 8.2.3.2). In 10/100/1000Base-T mode, the assertion of I_RX_ER during data reception indicates a data error.
IPE (bit 6)	IPv4 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.IPOFLD), the IPE bit is set to 0b. It has no effect on the packet filtering mechanism.
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.TUOFLD), the TCPE bit is set to 0b. It has no effect on the packet filtering mechanism.
RSV (bit 4)	Reserved
RSV (Bit 3)	Reserved
SEQ (bit 2)	Sequence Error When set, indicates a received packet with a bad delimiter sequence while in internal SerDes mode. 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 internal SerDes mode.
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.

3.2.4.3 VLAN Tag Field

Hardware stores additional information in the receive descriptor for 802.1q packets. If the packet type is 802.1q (determined when a packet matches VET and CTRL.VME = 1b), then the *VLAN Tag* field records the VLAN information and the four-byte VLAN information is stripped from the packet data storage. Otherwise, the *VLAN Tag* field contains 0000h.

Table 3-4. VLAN Tag Field Layout for 802.1q Packets

VLAN Tag 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.

3.2.5 Extended Rx Descriptor

If the RFCTL.EXSTEN bit is set and the RCTL.DTYP = 00b, the Ethernet controller uses the following Extended Rx descriptor.

Note: Refer to Section 3.2.8 for an example showing how to use extended Rx descriptors.

Descriptor Read Format:

	63	0
0	Buffer Address [63:0]	
8	Reserved	0

3.2.5.1 Buffer Address (64-Bit Offset 0.0)

This field contains the physical address of the receive data buffer. The size of the buffer is defined by the RCTL register (RCTL.BSIZE, RCTL.BSEX, RCTL.DTYP, RCTL, and FLXBUF fields).

3.2.5.2 DD (1-Bit Offset 8.0)

This is the location of the DD bit in the Status field. The software device driver must clear this bit before it hands the receive descriptor to the Ethernet controller. The software device driver can use this bit field later on as a completion indication of the hardware.

Descriptor Write Format:

	63 48	47 32	31 20	19 0
0	RSS Hash		MRQ	
	Packet Checksum[a]	IP Identification[a]		
8	VLANTag	Length	Extended Error	Extended Status

3.2.5.3 MRQC Field (32-Bit Offset 0.0)

Field	Bit(s)	Description
RSS Type	3:0	RSS Type Indicates the type of packet used for RSS computation (see Table 3-5).
Reserved	7:4	Reserved
Queue	12:8	Indicates the receive queue associated with the packet. It is generated by the indirection table as defined by the Multiple Receive Queues Enable field. This field is reserved when the Multiple Receive Queues Enable field of the Multiple Receive Queues Command register is set to 00b (Multiple Receive Queues are disabled).
Reserved	31:13	Reserved

3.2.5.4 RSS Type Decoding

The RSS Type field represents the packet type used by the RSS function as listed in Table 3-5.

Table 3-5. RSS Type Decoding

Packet Type	Description
0h	No hash computation done for this packet.
1h	IPv4 with TCP hash used.
2h	IPv4 hash used.
3h	IPv6 with TCP hash used.
4h	Reserved
5h	IPv6 hash used.
6h - Fh	Reserved

3.2.5.5 Packet Checksum (16-Bit Offset 0.48)

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 can modify the starting offset for the packet checksum calculation via the Receive Checksum Control Register (RXCSUM). To verify the TCP/UDP 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. Likewise, for fragmented UDP packets, the Packet Checksum field can be used to accelerate UDP checksum verification by the host processor.

For packets with VLAN header, the packet checksum includes the header if VLAN stripping is not enabled by the CTRL.VME. If VLAN header strip is enabled, the packet checksum and the starting offset of the packet checksum exclude the VLAN header.

3.2.5.6 IP Identification (16-Bit Offset 0.32)

This field stores the IP identification field in the IP header of the incoming packet. The software device driver should ignore this field when IPIDV is not set.

3.2.5.7 Extended Status (20-Bit Offset 8.0)

Field	Bit(s)	Description
Reserved	19:16	Reserved.
ACK	15	ACK Packet identification The ACK bit indicates that the received packet was an ACK packet with or without TCP payload depending on the *RFCTL.ACKD_DIS* bit.
Reserved	14:11	Reserved.
UDPV	10	Valid UDP XSUM The UDPV bit indicates that the incoming packet contains a valid (non-zero value) checksum field in an incoming Fragmented UDP IPv4 packet. It means that the Packet Checksum field contains the UDP checksum as described in this section. When this field is cleared in the first fragment that contains the UDP header, it means that the packet does not contain a valid UDP checksum and the checksum field in the Rx descriptor should be ignored. This field is always cleared in incoming fragments that do not contain the UDP header.
IPIDV	9	IP Identification Valid The IPIDV bit indicates that the incoming packet was identified as a fragmented IPv4 packet.
Reserved	8	Reserved.
PIF	7	Passed In-Exact Filter.
IPCS	6	IPv4 Checksum Calculated on Packet If active, hardware provides IPv4 checksum offload.
TCPCS	5	TCP Checksum Calculated on Packet If active withTCP checksum offload active, hardware provides IPv4 checksum offload. Pass/fail indication is provided in the Error field (IPE and TCPE). If IPCS is active and UDP checksum offload is active, hardware provides IPv4 checksum offload. Pass/fail indication is provided in the Error field (IPE and TCPE).
UDPCS	4	UDP Checksum Calculated on Packet If IPCS is active and fragmented UDP checksum offload is active, hardware provides IP checksum offoad for IPv4 packets. IP Pass/Fail indication is provided in the IPE field.
VP	3	Packet is 802.1q (matched VET).
IXSM	2	Ignore Checksum Indication When set to 1b, hardware does not provide checksum offload. Software device driver should ignore the IPCS, TCPCS, and UDPCS bits.
EOP	1	End of Packet.
DD	0	Descriptor Done.

3.2.5.8 Extended Errors (12-Bit Offset 8.20)

Field	Bit(s)	Description
RXE	11	Rx Data Error.
IPE	10	IPv4 Checksum Error.
TCPE	9	TCP/UDP Checksum Error.
Reserved	8	Reserved.
Reserved	7	Reserved.
SEQ	6	Sequence Error.
SE	5	Symbol Error.
CE	4	CRC Error or Alignment Error.
Reserved	3:0	Reserved.

3.2.6 Receive UDP Fragmentation Checksum

Ethernet controllers provide receive fragmented UDP checksum offload. The following hardware configuration should be used to enable this mode:

1. RXCSUM.PCSS field must be zero. The packet checksum start should be zero to enable auto start of the checksum calculation. See table below for exact description of the checksum calculation.

Incoming Packet Type	Packet Checksum	IP Identification	UDPV/IPIDV	UDPCS/TCPCS
None IPv4 packet	Unadjusted “16 bit ones complement” checksum of the entire packet (excluding VLAN header)	Reserved	0/0	0/0
Fragment IPv4 with TCP header	Same as above	Incoming IP Identification	0/1	0/0
Non-fragmented IPv4 packet	Same as above	Reserved	0b/0b	Depends on transport header and TUOFL field
Fragmented IPv4 without transport header	The unadjusted 1’s complement checksum of the IP payload	Incoming IP Identification	0b/1b	1b/0b
Fragmented IPv4 with UDP header	Same as above	Incoming IP Identification	1b if the UDP header checksum is valid/1	1b/0b

Note: When the software device driver computes the “16 bit ones complement” checksum on the incoming packets of the UDP fragments, it should expect a value of FFFFh.

3.2.7 Packet Split Receive Descriptor

The Ethernet controller uses the Packet Split feature when the RFCTL.EXSTEN bit is set and RCTL.DTYP equals 01b. The software device driver must also program the buffer sizes in the PSRCTL register.

Table 3-6. Descriptor Read Format

	63:60	59:56	55:52	51:48	47:44	43:40	39:36	35:32	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
0	Buffer Address 0															
8	Buffer Address 1															
16	Buffer Address 2															
24	Buffer Address 3															

3.2.7.1 Buffer Addresses [3:0] (4 x 64 bit)

The physical address of each buffer is written in the buffer addresses fields. The sizes of these buffers are statically defined by BSIZE0 to BSIZE3 in the PSRCTL register.

The Buffer Address 0 includes the address of the buffer that contains the header information. The Receive DMA module stores the header portion of the received packets into this buffer.

The Buffer Address 1, 2, and 3 include the address of the buffers assigned to the data payload portion of the received packet. First, buffer 1 is used. If the data payload is larger than the available memory, as defined in the PSRCTL register, then Buffer Address 2 is used for the remainder of the data. Similarly, Buffer Address 3 is used when the memory portion defined by Buffer Address 2 is full. If the buffer size in PSRCTL register is equal to 0b, then the corresponding buffer address is not used.

Note: The following items should be observed:

- All buffer addresses in a Packet Split descriptor must be word aligned.
- Packet header can't span across buffers, therefore, the size of the first buffer must be larger than any expected header size. Otherwise the packet will not be split.
- If software sets a buffer size to 0b, all buffers following that one should be set to 0b as well. Pointers in the Packet split Receive descriptors to buffers with a 0 size should be set to any address, but not to null pointers. Hardware does not write to these addresses.
- When configured to packet split and a given packet spans across two or more packet split descriptors the first buffer of any descriptor (other than the first one) will not be used.

Header splitting is shown in Figure 3-2.

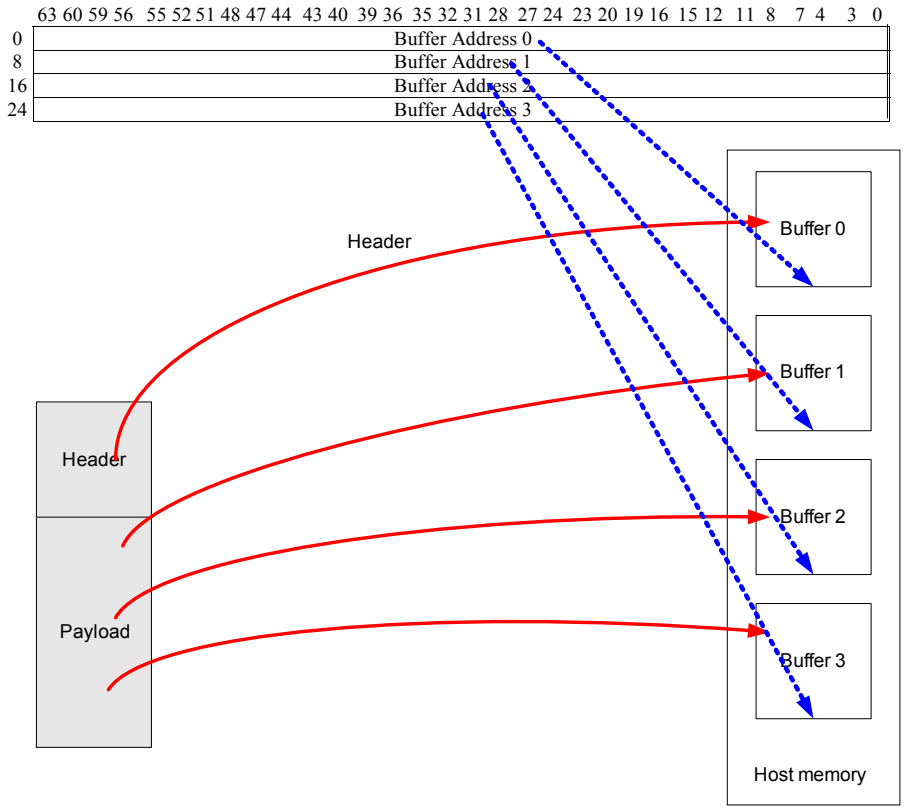


Figure 3-2. Header Splitting

3.2.7.2 DD (1 bit offset 8.0)

The software device driver can use the DD bit from the status field to determine when a descriptor has been used. Thus, the software device driver must ensure that the least significant bit of Buffer Address 1 or 3 is 0b. This should not be an issue, since the buffers should be page aligned for the packet split feature to be useful. Any software device driver that cannot align buffers should not use this descriptor format.

Table 3-7. Descriptor Write-Back Format

	63:60	59:56	55:52	51:48	47:44	43:40	39:36	35:32	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
0	RSS Hash								MRQ							
	Packet Checksum				IP Identification											
8	VLAN Tag				Length 0				Extended Error			Extended Status				
16	Length 3				Length 2				Length 1				Header Status			
24	Reserved (Least significant bit is a duplicate of the DD bit.)															

NOTES:
1. MRQC: Same as Extended Receive Descriptor.
2. Packet Checksum, IP Identification, RSS Hash: Same as Extended Receive Descriptor.
3. Extended Status, Extended Errors, VLAN Tag: Same as Extended Receive Descriptor.

3.2.7.3 Length [3:1] (3 x 16 bit offset 16.16), Length 0 (16 bit offset 8.32)

Upon a packet reception, hardware stores the packet data into one or more of the indicated buffers. The hardware writes in the length field of each buffer the number of bytes that were posted in the corresponding buffer. If no packet data is stored in a given buffer, hardware writes 0b in the corresponding length field. Length covers the data written to the receive buffer including CRC bytes. Software is responsible for checking the Length fields of all buffers for data that hardware might have written to the corresponding buffers.

Table 3-8. Header Status (16-bit Offset 16.0)

15	14:10	9:0
HDRSP	Reserved	HLEN (Header Length)

3.2.7.4 HDRSP (bit 15)

The HDRSP bit (when active) indicates that hardware split the headers from the packet data for the packet contained in this descriptor. Table 3-8 identifies the packets supported by header/data split functionality. Packets with a data portion smaller than 16 bytes are not guaranteed to be split. If the Ethernet controller is not configured to provide any offload that requires packet parsing, HDRSP bit is set to 0b even if packet split was enabled. No split packets are stored linearly in the buffers of the receive descriptor.

3.2.7.5 HLEN (bits 9:0)

The HLEN field indicates the Header Length in byte count that was analyzed by the Ethernet controller. The packet type table shows the packets supported by the Ethernet controller. The Ethernet controller posts the first HLEN bytes of the incoming packet to buffer 0 of the receive descriptor. Note that the HLEN field is active only in split packet mode.

The Ethernet controller provides header split for the packet types listed in Table 3-9. Other packet types are posted sequentially in the buffers of the packet split receive buffers.

Table 3-9. Supported Packets

Packet Type	Description	Header Split
0h	MAC, (VLAN/SNAP) Payload	No.
1h	MAC, (VLAN/SNAP) IPv4, Payload	Split header after L3 if packets are fragmented.
2h	MAC, (VLAN/SNAP) IPv4, TCP/UDP, Payload	Split header after L4 if packets are not fragmented. Otherwise, treat the packet as packet type 1.
3h	MAC (VLAN/SNAP), IPv4, IPv6, Payload	Split header after L3 if either IPv4 or IPv6 indicates a fragmented packet.
4h	MAC (VLAN/SNAP),IPv4, IPv6,TCP/UDP, Payload	Split header after L4 if IPv4 is not fragmented and if IPv6 does not include fragment extension header. Otherwise treat the packet as packet type 3.
5h	MAC (VLAN/SNAP), IPv6, Payload	Split header after L3 if packets are fragmented.
6h	MAC (VLAN/SNAP), IPv6, TCP/UDP, Payload	Split header after L4 if IPv6 does not include fragment extension header. Otherwise treat packet as packet type 5.
7h	MAC, (VLAN/SNAP) IPv4, TCP, ISCSI, Payload	Split header after L5 if packet is not fragmented. Otherwise treat packet as packet type 1.
8h	MAC, (VLAN/SNAP) IPv4, TCP/UDP, NFS, Payload	Split header after L5 if packet is not fragmented. Otherwise treat as packet type 1.
9h	MAC (VLAN/SNAP), IPv4, IPv6,TCP, ISCSI, Payload	Split header after L5 if IPv4 is not fragmented and if IPv6 does not include fragment extension header. Otherwise, treat the packet as packet type 3.
Ah	MAC (VLAN/SNAP), IPv4, IPv6,TCP/UDP, NFS, Payload	Split header after L5 if IPv4 is not fragmented and if IPv6 does not include fragment extension header. Otherwise, treat the packet as packet type 3.
Bh	MAC (VLAN/SNAP), IPv6, TCP, ISCSI, Payload	Split header after L5 if IPv6 does not include fragment extension header. Otherwise treat packet as packet type 5.
Ch	MAC (VLAN/SNAP), IPv6, TCP/UDP, NFS, Payload	Split header after L5 if IPv6 does not include fragment extension header. Otherwise, treat packet as packet type 5.

NOTES:
1. The header of a fragmented IPv6 packet is defined to the fragmented extension header.
2. If the Ethernet controller is not configured to provide any offload that requires packet parsing, the packet type field is set to 0b regardless of the actual packet type. When Packet Split is enabled, the packet type field is always valid.

3.2.8 Using Extended Rx Descriptors

The following example shows how to keep track of buffer addresses when using extended Rx descriptors.

When using extended Rx descriptors there is an array of descriptors that are “N” long. In this example two descriptors of the array are used with one descriptor being processed.

Descriptor 1 - buffer address 1
  VLAN tag, length, extended error, extended status, etc.

Descriptor 2 - buffer address 2
  VLAN tag, length, extended error, extended status, etc.

Software moves the tail by one and then hardware reads the descriptor.

Hardware processes the packet information for write back. As a result, the descriptor array now looks like:

Descriptor 1 - buffer address 1
  RSS hash type, MRQ, packet checksum, etc.
  VLAN tag, length, extended error, extended status, etc.

Descriptor 2 - buffer address 2
  VLAN tag, length, extended error, extended status, etc.

Note that the Ethernet controller returned the extended Rx descriptor in place of the buffer address. If using legacy descriptors, they would normally look like buffer address 1 but with an updated size, length, etc.

To keep track of address buffers (since buffer address 1 was placed into descriptor 1), software device drivers need a shadow list of buffer addresses that can be re-built. For example:

Descriptor 1 - buffer address 1

Descriptor 2 - buffer address 2

. . . . .

Descriptor N - buffer address N

Once descriptor 1 is processed (with new extended Rx descriptor data), software device drivers need access to the shadow list to retrieve the buffer address.

Note: Software device drivers should always use the shadow list to keep from reading corrupted data. For example, from the PCI bus.

3.2.9 Receive Descriptor Fetching

The descriptor fetching algorithm is designed to support large bursts across the PCIe* 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 PCIe* 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.

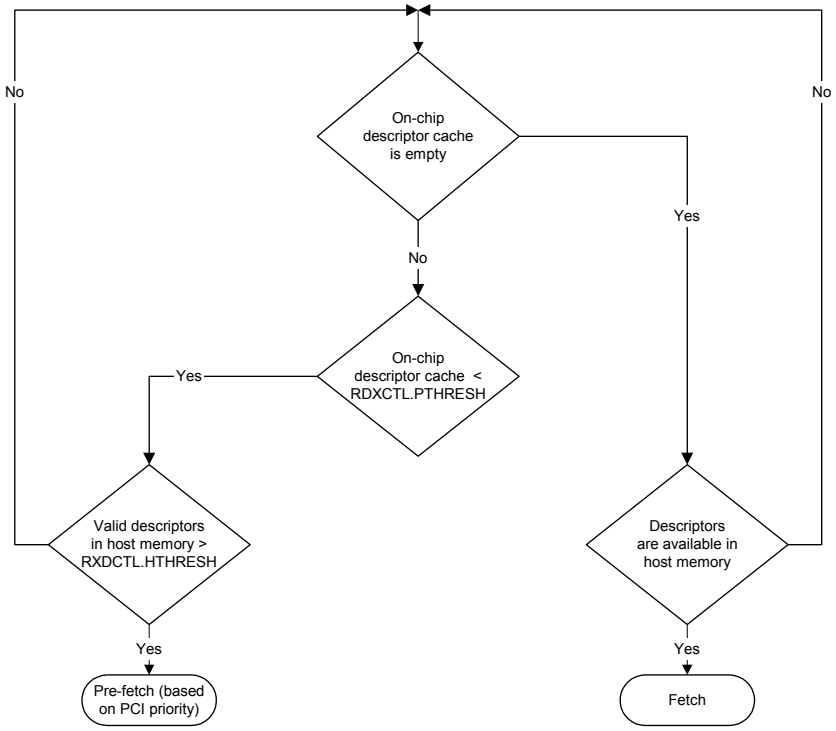


Figure 3-3. Receive Descriptor Fetching Algorithm

3.2.10 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. The receive descriptor packing mechanism minimizes the occurrence of partial line write backs.

3.2.10.1 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 these conditions:

- RXDCTL.WTHRESH descriptors have been used (the specified max threshold of unwritten used descriptors has been reached)
- The last descriptor of the allocated descriptor ring has been used (enables the hardware to re-align to the descriptor ring start)
- The receive timer expires (RADV, RDTR, or ITR)
- Explicit software flush (RDTR.FPD)

When the number of descriptors specified by RXDCTL.WTHRESH have been used, they are written back, regardless of cache line alignment. It is therefore recommended that WTHRESH be a multiple of cache line size. When a receive timer (RADV, RDTR, or ITR) expires, all used descriptors are forced to be written back prior to initiating the interrupt, for consistency. Software can explicitly flush accumulated descriptors by writing the RDTR register with the high-order bit (FPD) set.

3.2.11 Receive Interrupts

The Ethernet controller can generate these receive-related interrupts:

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

3.2.11.1 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.

3.2.11.1.1 Receive Interrupt Delay Timer / Packet Timer (RDTR)

Note: RDTR has been obsoleted by the Interrupt Throttling Rate (ITR) register. ITR provides a more straightforward interrupt moderation solution than RDTR. If possible, use ITR instead of 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 (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.

If RSS interrupt registers are enabled (Section 3.2.14.3), timer expiration sets one of more bits in the RSS Interrupt Request register, rather than ICR.RXT0. The bits set correspond to the CPUs for which packets have been posted into system memory.

See section Section 13.3.43 for more details on the Packet Timer.

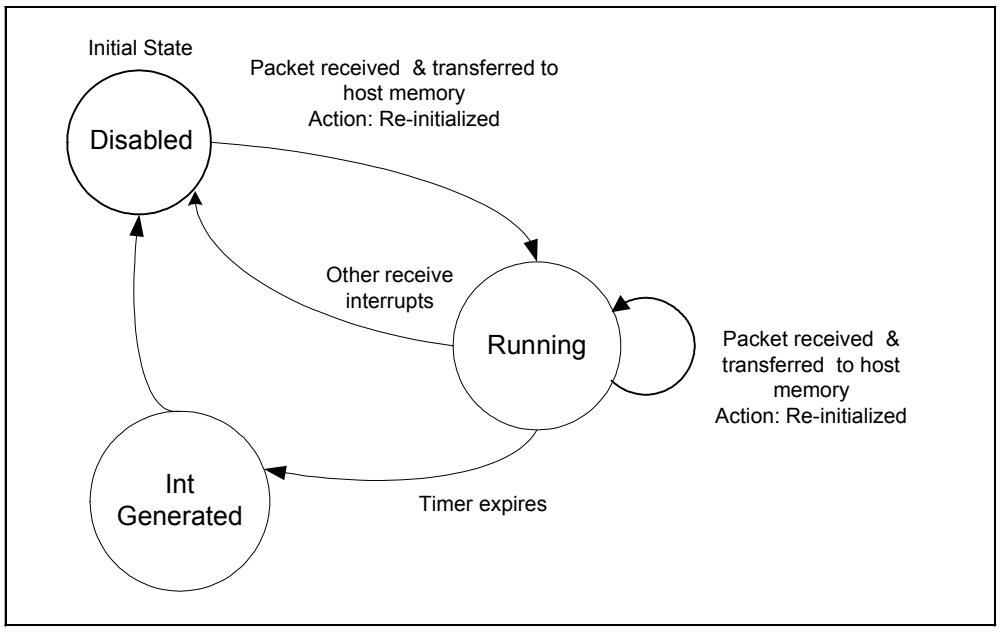


Figure 3-4. Packet Delay Timer Operation (State Diagram)

3.2.11.1.2 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.

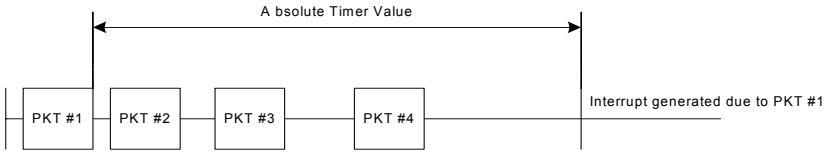
Setting RADV to 0b or RDTR to 0b disables the Absolute Timer. To disable the Packet Timer only, RDTR should be set to RADV + 1b.

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.

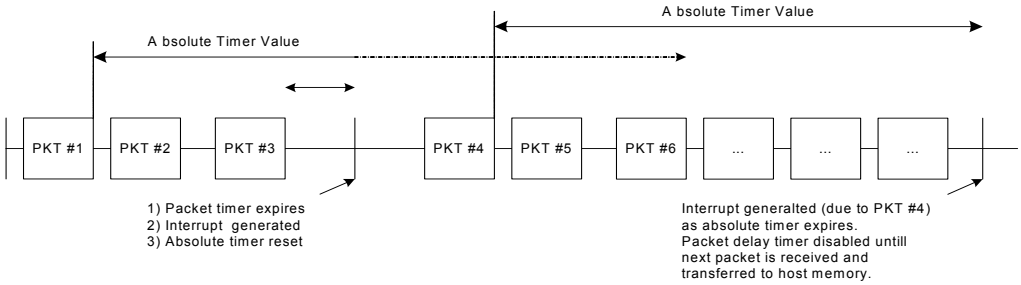
If RSS interrupt registers are enabled (Section 3.2.14), timer expiration sets one of more bits in the RSS Interrupt Request register, rather than ICR.RXT0. The bits set correspond to the CPUs for which packets have been posted into system memory.

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

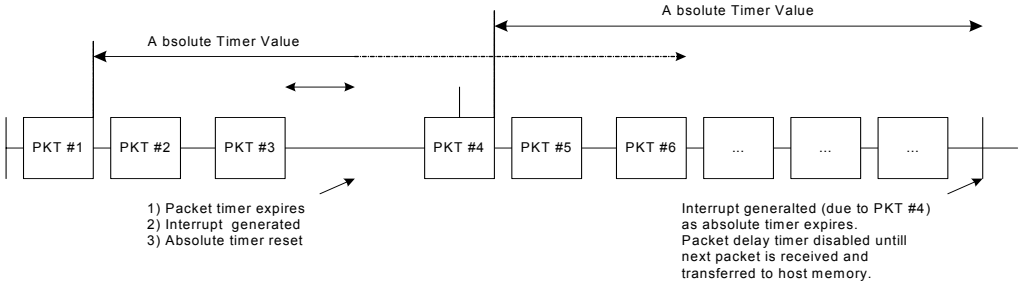
Case A: Using only an absolute timer



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.



3.2.11.2 Small Receive Packet Detect (ICR.SRPD)

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.

3.2.11.3 Receive ACK Frame Detect (ICR.ACK)

A receive ACK frame interrupt is asserted when a frame is detected to be an ACK frame. Detection of ACK frames are masked through the IMS register. When a frame is detected as an ACK frame an interrupt is asserted after the RAID.ACK_DELAY timer has expired and the ACK frame interrupts were not masked in the IMS register.

Note: The ACK Frame detect feature is only active when configured to Packet Split (RCTL.DTYP = 01b) or the Extended Status feature is enabled (RFCTL.EXSTEN is set).

3.2.11.4 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).

3.2.11.5 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 PCIe* bus is too slow draining the receive FIFO. Incoming packets that overrun the FIFO are dropped and do not affect future packet reception. Note that this interrupt stops and re-initializes the entire active delayed receive interrupt process.

3.2.12 Receive Packet Checksum Offloading

The Ethernet controller supports the offloading of three receive checksum calculations: the Packet Checksum, the IPv4 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 packets with a VLAN header, the packet checksum includes the header if VLAN stripping is not enabled by CTRL.VME. If the VLAN header strip is enabled, the packet checksum and the starting offset of the packet checksum exclude the VLAN header due to masking of VLAN header. For example, for an Ethernet II frame encapsulated as an 802.3ac VLAN packet with CTRL.VME and with RXCSUM.PCSS set to 14 decimal, the Packet Checksum includes 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 can be off-loaded to the Ethernet controller. If RXCSUM.IPOFLD is set to 1b, the Ethernet controller calculates the IPv4 checksum and indicates a pass/fail indication to software via the IPv4 Checksum Error bit (RDESC.IPE) in the ERROR field of the receive descriptor. Similarly, if RXCSUM.TUOFLD is set to 1b, the Ethernet controller calculates the TCP or UDP checksum and indicates a pass/fail condition to software via 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 respectively). In addition, if RFCTL.Ipv6_DIS and RFCTL.IP6Xsum_DIS are cleared to 0b and RXCSUM.TUOFLD is set to 1b, the Ethernet controller calculates the TCP or UDP checksum for IPv6 packets. It then indicates a pass/fail condition in the TCP/UDP Checksum Error bit (RDESC.TCPE).

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

Table 3-10. 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)	Yes	Yes
Packet has 802.3ac tag	Yes	Yes
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.

Table 3-10 lists the general details about what packets are processed. In more detail, the packets are passed through a series of filters (Section 3.2.1.1 through Section 3.2.1.5) to determine if a receive checksum is calculated.

3.2.13 Manageability Receive Filtering

Refer to the *Intel® 82571/82572/631xESB/632xESB GbE Controller TCO/System Manageability Interface* Application Note for detailed information about manageability receive filtering.

3.2.14 Multiple Receive Queues & Receive-Side Scaling (RSS)

Multiple Receive Queues are enabled when the Multiple Receive Queues Enable bits do not equal 00b. Multiple Receive Queues are mutually exclusive with UDP fragmentation and are not supported when a legacy receive descriptor format is used.

When Multiple Receive Queues are enabled, the Ethernet controller provides several pieces of information to the software. Some are requirements of RSS while others are provided for device driver assistance:

- RSS can only be enabled during initialization. Disabling/enabling the RSS feature during normal operation is not recommended.
- A Dword result of the RSS hash function. This is used by the stack for flow classification and is written into the receive packet descriptor (required by RSS).
- A 4-bit RSS Type field. This conveys the hash function used for the specific packet (required by RSS).
- A 5-bit output of a redirection table. This identifies the CPU to process the packet and is written into the receive packet descriptor (for device driver use).
- A mechanism for issuing an interrupt to one or more CPUs.

The following summarizes the process of classifying a packet into a receive queue:

1. The receive packet is parsed into the header fields used by the hash operation (for example, IP addresses, TCP port, etc.)
2. A hash calculation is performed. The Ethernet controller supports a single hash function as defined by RSS. The Ethernet controller does not indicate to the device driver which hash function is used. The 32-bit result is fed into the receive packet descriptor.
3. The seven LSBs of the hash result are used as an index into a 128-entry Redirection Table. Each entry in this table contains a 5-bit CPU number. This 5-bit value is fed into the packet receive descriptor. In addition, each entry provides a single bit queue number, which denotes the queue into which the packet is routed.

When multiple request queues are disabled, packets enter hardware queue 0. System software can enable or disable RSS at any time. While RSS is disabled, system software can update the contents of any of the RSS related registers.

When multiple request queues are enabled in RSS mode, undecoded packets enter hardware queue 0. The 32-bit tag, which is normally a result of the hash function, equals 0. The 5-bit MRQ field also equals 0.

3.2.14.1 RSS Hash Function

A single hash function is defined with five variations for the following cases:

- IPv4. The Ethernet controller parses the packet and uses the IPv4 source and destination addresses to generate the hash value.
- TCP/IPv4. The Ethernet controller parses the packet to identify an IPv4 packet containing a TCP segment. The Ethernet controller uses the IPv4 source and destination addresses and the TCP local and remote port values to generate the hash value.
- IPv6. The Ethernet controller parses the packet to identify an IPv6 packet and uses the IPv6 source and destination addresses to generate the hash value.
- TCP/IPv6. The Ethernet controller parses the packet to identify an IPv6 packet containing a TCP segment. The Ethernet controller uses the IPv6 source and destination addresses and the TCP local and remote port values to generate the hash value.
- IPv6Ex. The Ethernet controller parses the packet to identify an IPv6 packet. Extension headers should be parsed for a *Home-Address-Option* field (for source address) or the Routing-Header-Type-2 field (for destination address). Note that the packet is not required to contain any of these extension headers to be hashed by this function. If the specified extension headers are not present in the packet, the Ethernet controller uses the source/destination from the standard IPv6 header.
- TCPIPv6Ex. The Ethernet controller parses the packet to identify an IPv6 packet containing a TCP segment with extensions. Extension headers should be parsed for a Home-Address-Option field (for source address) or the *Routing-Header-Type-2* field (for destination address). Note that the packet is not required to contain any of these extension headers to be hashed by this function. If the specified extension headers are not present in the packet, the Ethernet controller uses the source/destination from the standard IPv6 header.

The following cases are in addition to the RSS standard:

- UDPIPv4 - The Ethernet controller parses the packet to identify a packet with UDP over IPv4
- UDPIPv6 - The Ethernet controller parses the packet to identify a packet with UDP over IPv6
- UDPIPv6Ex - The Ethernet controller parses the packet to identify a packet with UDP over IPv6 with extensions

A packet is identified as containing a TCP segment if all of the following conditions are met:

- The transport layer protocol is TCP (not UDP, ICMP, IGMP, etc.).
- The TCP segment can be parsed (for example, IP options can be parsed or the packet is not encrypted).
- The packet is not IP fragmented (even if the fragment contains a complete TCP header).

Bits[31:16] of the Multiple Receive Queues Command (MRQC) register enable each of the above hash function variations (several may be set at a given time). If several functions are enabled at the same time, priority is defined as follows (skip functions that are not enabled):

- IPv4 Packet.
  a. Try using the TCP/IPv4 function.
  b. Try using the IPv4_UDP function.
  c. Try using the IPv4 function.

- IPv6 Packet.
  a. If TCPIPv6Ex is enabled, try using the TCP/IPv6Ex function; else, if TCPIPv6 is enabled, try using the TCPIPv6 function.
  b. If UDPIPv6Ex is enabled, try using the UDPIPv6EX function; else, if UDPIPv6 is enabled, try using the UDPIPv6 function.
  c. If IPv6Ex is enabled, try using the IPv6Ex function; else, if IPv6 is enabled, try using the IPv6 function.

The following combinations are currently supported:

- Any combination of IPv4, TCPIPv4, and UDPIPv4, and or,
- Any combination of either IPv6, TCPIPv6, and UDPIPv6 or IPv6Ex, TCPiPv6Ex, and UDPIPv6Ex

When a packet cannot be parsed by the above rules, it enters hardware queue 0. The 32-bit tag (which is a result of the hash function) equals 0b. The 5-bit MRQ field also equals zero.

In the case of tunneling (for example, IPv4-IPv6 tunnel), the external IP address (in the base header) is used.

The 32-bit result of the hash computation is written into the packet descriptor and also provides an index into the Indirection Table.

The following notation is used to describe the following hash functions:

- Ordering is little endian in both bytes and bits. For example, the IP address 161.142.100.80 translates into A18E 6450h in the signature.
- A "^" denotes bit-wise eXclusive OR (XOR) operation of same width vectors.
- @x-y denotes bytes x through y (including both of them) of the incoming packet, where byte 0 is the first byte of the IP header. In other words, we consider all byte offsets as offsets into a packet where the framing layer header has been stripped out. Therefore, the source IPv4 address is referred to as @12-15, while the destination v4 address is referred to as @16-19.
- @x-y, @v-w denotes concatenation of bytes x-y followed by bytes v-w, preserving the order in which they occurred in the packet.

All hash function variations (IPv4 and IPv6) follow the same general structure. Specific details for each variation are described in the following section. The hash uses a random secret key of length 320 bits (40 bytes). The key is generated and supplied through the RSS Random Key Register (RSSRK).

The algorithm works by examining each bit of the hash input from left to right. Our nomenclature defines left and right for a byte array as follows:

Given an array K with k bytes, our nomenclature assumes that the array is laid out as:

```
K[0] K[1] K[2] … K[k-1]
```

K[0] is the left most byte, and the most significant bit of K[0] is the left most bit. K[k-1] is the right most byte, and the least significant bit of K[k-1] is the right most bit.

```
ComputeHash(input[], N)

For hash-input input[] of length N bytes (8N bits) and a random secret key K of 320
bits

    Result = 0;

    For each bit b in input[] {

        if (b == 1) then Result ^= (left-most 32 bits of K);

        shift K left 1 bit position;

        }

    return Result;
```

The following four pseudo-code examples are intended to help clarify exactly how the hash is to be performed in four cases: IPv4 with and without ability to parse the TCP header and IPv6 with and without a TCP header.

3.2.14.1.1 Hash for IPv4 with TCP

```
Concatenate SourceAddress, DestinationAddress, SourcePort, DestinationPort into
one single byte-array, preserving the order in which they occurred in the packet:
Input[12] = @12-15, @16-19, @20-21, @22-23.

Result = ComputeHash(Input, 12);
```

3.2.14.1.2 Hash for IPv4 without TCP

```
Concatenate SourceAddress and DestinationAddress into one single byte-array

Input[8] = @12-15, @16-19

Result = ComputeHash(Input, 8)
```

3.2.14.1.3 Hash for IPv6 with TCP

```
Similar to above:

Input[36] = @8-23, @24-39, @40-41, @42-43

Result = ComputeHash(Input, 36)
```

3.2.14.1.4 Hash for IPv6 without TCP

```
Input[32] = @8-23, @24-39

Result = ComputeHash(Input, 32)
```

3.2.14.2 Redirection Table

The redirection table is a 128-entry structure, indexed by the 7 least significant bits of the hash function output. Each entry of the table contains the following:

- Bit [7]: Queue Index.
- Bits [6:5]: Reserved.
- Bits [4:0]: CPU Index.

The CPU value indexed by the hash function is written into the packet descriptor to serve as an indication of the CPU that should process the packet. The Queue Index determines the physical queue for the packet.

System software may update the redirection table during run time. Such updates of the table are not synchronized with the arrival time of received packets. Therefore, it is not guaranteed that a table update will take effect on a specific packet boundary.

3.2.14.3 RSS Interrupt Registers

The set of registers described in this section controls the generation of a receive packet interrupt when RSS is enabled. The registers allow up to 32 processors to track and manage their respective interrupts, while not interfering with the progress of other processors.

When a receive packet and its descriptors have been posted to system memory (and the appropriate timer expired if it was enabled), the RSS Interrupt Request[i] bit is set, where i is the CPU value indicated by the RSS Redirection Table register (I = 0, … , 31). If bit i in the RSS Interrupt Mask register is set, then the respective bit is set in the CPU vector register. The first bit to be set in the CPU vector initiates an interrupt pulse request to the device interrupt logic. In other words, a pulse is generated when the CPU vector transitions from an all clear state. The interrupt pulse is routed to the ICR.RXT0 bit. If the Packet Delay Timer or Absolute Timer is enabled, the interrupt pulse is routed to that mechanism instead.

Note: If a bit is set in the CPU vector after it is read (that is, the set bit was not reflected in the data returned for the read request), then hardware must guarantee that an additional interrupt request pulse is generated. This is required to avoid losing interrupts while the CPU vector is read.

When software reads the CPU vector, the register is cleared. This also masks any bit in the RSS Interrupt Mask register that had a corresponding bit set in the CPU vector. Therefore, when the CPU vector is read and a specific bit was set in it, the masking operation blocks additional interrupts to the CPU for which the bit was set. Reading the CPU vector also clears the RXT0, ACK and SPRD bits in the ICR. Writing to a bit in the CPU vector masks the respective bit in the RSS Interrupt Mask register.

Use of the CPU vector is not required to implement RSS or multiple receive queues. The relevant RSS Interrupt Request bit is set following the posting of a packet and its descriptor to system memory. Software may clear a bit in the RSS Interrupt Request register by writing a 1b into it. This may be done in order to avoid future interrupt requests by packets that will be processed anyway. (Section 3.2.14.3.1," contains more detail.) If software requests to clear a bit in the RSS Interrupt Request register at the same clock that an interrupt request is issued by hardware to that bit, then the RSS Interrupt Request bit is not set. Writing a 0b into a RSS Interrupt Request bit has no impact on its contents.

Software may also unmask a bit in the RSS Interrupt Mask register by writing a 1b into it. Writing a 0b into a bit has no impact on its contents. Software may unmask a bit after it has completed processing and wishes to enable interrupts by future packets.

The CPU vector and its corresponding registers (RSS Interrupt Mask register and RSS Interrupt Request register) are disabled when multiple receive queues are disabled or the RSS Interrupt Enable bit is cleared in the MRQC register. In this case, receive interrupt indications bypass the CPU vector and are issued directly to the interrupt logic.

3.2.14.3.1 Usage Model

Note: This section is provided for exposition purposes only and does not dictate a specific usage of the device.

The procedure starts with by enabling one or more RSS Interrupt Mask bits. Incoming receive packets set a corresponding bit in the RSS Interrupt Request register, which initiates an interrupt to the host CPU. An ISR first reads the ICR and identifies that receive packets are pending. The ISR may then mask further interrupts through the IMC register. It then reads the CPU vector and clears the CPU vector bits in the ICR corresponding to receive packets and RSS Interrupt Mask bits set in the CPU vector. This masks further receive packet interrupt requests to that CPU. Since the CPU vector was cleared, receive packets for other CPUs generate a new interrupt request to the interrupt logic. The interrupt is processed as normal and enabled through the IMS register.

The ISR writes 1b (clear) to the RSS Interrupt Request bit that corresponds to its CPU. The write cancels any pending requests for an interrupt by packets to this CPU that have already arrived. A dummy read should be done to the device to push any pending packets into system memory. This allows all packets to be processed as normal. This sequence can be repeated as needed.

Once the ISR has completed processing all or some of the packets, it writes a 1b into the RSS Interrupt Mask bit that corresponds to this CPU, enabling interrupt requests for this CPU.

Receive packets belonging to other CPUs may still generate interrupt requests while ISR handles packets received for other CPUs. If a packet is received while the ISR is running and after the CPU vector was read, then the receive packet generates a separate interrupt request pulse to the interrupt logic. The interrupt request sets the appropriate bit in the ICR, which generates a new interrupt at some later time (after interrupts are unmasked again by the ISR).

Note: The previous sequence, while insuring that packets are not lost or ignored, allows spurious interrupts to occur. If a packet arrives after the RSS Interrupt Request register was cleared but while the ISR is handling the packets queues, the packet may set the RSS Interrupt Request register. However, it is still processed by the ISR. When the RSS Interrupt Mask bit is later set, the RSS Interrupt Request register issues an interrupt request for the packet that was processed by the ISR, generating a spurious interrupt.

Example 1. Possible RSS Interrupt Scheme

```
Read ICR

Mask interrupts in ICR

Read CPU Vector

Unmask interrupts in ICR

While (still_work and time_not_expired) {

        Write '1' to RSS Interrupt Request [i]

        Dummy Read

        Process packets

    }

    Write '1' to RSS Interrupt Mask [i]

    Return
```

3.2.15 RSS Verification Suite

This section contains the values used in the given examples. Assume that the random key byte-stream is:

```
0x6d, 0x5a, 0x56, 0xda, 0x25, 0x5b, 0x0e, 0xc2,

0x41, 0x67, 0x25, 0x3d, 0x43, 0xa3, 0x8f, 0xb0,

0xd0, 0xca, 0x2b, 0xcb, 0xae, 0x7b, 0x30, 0xb4,

0x77, 0xcb, 0x2d, 0xa3, 0x80, 0x30, 0xf2, 0x0c,

0x6a, 0x42, 0xb7, 0x3b, 0xbe, 0xac, 0x01, 0xfa
```

3.2.15.1 IPv4

Destination Address/ Port	Source Address/Port	IPv4 Only	IPv4 with TCP
161.142.100.80 :1766	66.9.149.187 : 2794	323E 8FC2h	51CC C178h
65.69.140.83 : 4739	199.92.111.2 : 14230	D718 262Ah	C626 B0EAh
12.22.207.184 : 38024	24.19.198.95 :12898	D2D0 A5DEh	5C2B 394Ah
209.142.163.6 : 2217	38.27.205.30 : 48228	8298 9176h	AFC7 327Fh
202.188.127.2 :1303	153.39.163.191 : 44251	5D18 09C5h	10E8 28A2h

3.2.15.2 IPv6

The IPv6 address tuples are only for verification purposes and may not make sense as a tuple.

Destination Address/ Port	Source Address/Port	IPv6 Only	IPv6 with TCP
3FFE:2501:200:1FFF::7 (1766)	3FFE:2501:200:3::1 (2794)	2CC1 8CD5	4020 7D3D
FF02::1 (4739)	3FFE:501:8::260:97FF:FE40:EFAB (14230)	0F0C 461C	DDE5 1BBF
FE80::200:F8FF:FE21:67CF (38024)	3FFE:1900:4545:3:200:F8FF:FE21:67CF (44251)	4B61 E985	02D1 FEEF

3.3 Transmit Descriptor Ring Structure

The transmit descriptor ring structure is shown in Figure 3-5. 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.

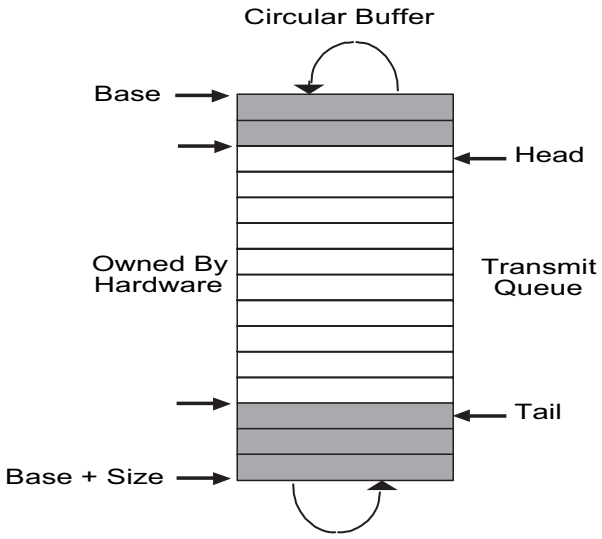


Figure 3-5. Transmit Descriptor Ring Structure

Shaded boxes in Figure 3-5 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 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.

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

- Scan memory for descriptor status write-backs.
- Read the hardware head register. All packets up to but EXCLUDING the one pointed to by head have been sent or buffered and can be reclaimed
- 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.

3.3.1 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.

3.3.2 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 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 previous Ethernet controllers, 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.

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 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 3.3.3.1.

The other two conditions are only valid if descriptor bursting is enabled (see Section 13.3.68). 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.

3.3.3 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 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 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.

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

  For the **631xESB/632xESB**, set when the link status changes. When using the **82563EB/82564EB** (PHY), link status changes are determined by the PHY and indicated by the PHY via a change in its LINK indication. When using SerDes, the **631xESB/632xESB** indicates a link status change using its LOS (loss of sync) indication. In SerDes mode, if hardware auto-negotiation is enabled, the **631xESB/632xESB** might also detect and signal a link status change if a configuration base page register that is zero is received or if either the LRST or ANE bits are set by software.
- Transmit Descriptor Ring Low Threshold Hit (TXD_LOW) - Set when the total number of transmit descriptors available hits the low threshold specified in the TXDCTL.LWTHRESH field in the Transmit Descriptor Control register. For the purposes of this interrupt, the number of transmit descriptors available is the difference between the Transmit Descriptor Tail and Transmit Descriptor Head values, minus the number of transmit descriptors that have been prefetched. Up to eight descriptors can be prefetched.

3.3.3.1 Delayed Transmit Interrupts

This mechanism enables 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 3.4.3.1 for specific details.

3.4 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 PCIe* underruns.

Another transmit feature is TCP Segmentation. The hardware has the capability to perform packet segmentation on large data buffers off-loaded from the Network Operating System (NOS). This feature is described in detail in Section 3.6.

3.4.1 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 (9014 bytes; 9234 byes for the **82571EB/82572EI/631xESB/632xESB**). A packet typically consists of two (or more) descriptors, one (or more) for the header and one or more 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.

3.4.2 Transmit Descriptors

The Ethernet controller provides three types of transmit descriptor formats.

The original descriptor is referred to as the “legacy” descriptor format. The other two 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 3-11 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 3.4.3.

Table 3-11. Transmit Descriptor (TDESC) Layout

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

3.4.3 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 3-12. 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 3-12. 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	Reserved	STA	CMD	CSO	Length

Table 3-13. 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. Length (TDESC.LENGTH) specifies the length in bytes to be fetched from the buffer address provided. The maximum length associated with any single legacy descriptor is 9014 bytes (9234 byes for the **82571EB/82572EI/631xESB/632xESB**). The maximum allowable packet size for transmits changes based on the value written to the Packet Buffer Allocation register. Descriptor length(s) can be limited by the size of the transmit FIFO. All buffers comprising a single packet must be able to be stored simultaneously in the transmit FIFO. For any individual packet, the sum of the individual descriptors' lengths must be below 9014 bytes (9234 byes for the **82571EB/82572EI/631xESB/632xESB**). Descriptors with zero length (null descriptors) transfer no data. Null descriptors might appear only between packets and must have their EOP bits set.
CSO/CSS	Checksum Offset and Start A Checksum offset (TDESC.CSO) field indicates where, relative to the start of the packet, to insert a TCP checksum if this mode is enabled. A Checksum start (TDESC.CSS) field indicates where to begin computing the checksum. Both CSO and CSS are in units of bytes.These must both be in the range of data provided to the Ethernet controller in the descriptor. This means for short packets that are padded by software, CSS and CSO must be in the range of the unpadded data length, not the eventual padded length (64 bytes). In case of 802.1Q header, the offset values depends on the VLAN insertion enable bit - CTRL.VME and the VLE bit. In case they are not set (VLAN tagging included in the packet buffers), the offset values should include the VLAN tagging. In case these bits are set (VLAN tagging is taken from the packet descriptor), the offset values should exclude the VLAN tagging. **Note:** UDP checksum calculation is not supported by the legacy descriptor. Because the CSO field is 8 bits wide, it limits the location of the checksum to 255 bytes from the beginning of the packet. EOP stands for End-Of-Packet and when set, indicates the last descriptor making up the packet. One or many descriptors can be used to form a packet. Hardware inserts a checksum at the offset indicated by the CSO field if the Insert Checksum bit (IC) is set. Checksum calculations are for the entire packet starting at the byte indicated by the CSS field. A value of 0b corresponds to the first byte in the packet. Hardware ignores IC, and CSO unless EOP is set. CSS must be set in the first descriptor for a packet. In addition, IC is ignored if CSO or CSS are out of range. This occurs if (CSS >/ = length) OR (CSO length - 1).
CMD	Command field See Section 3.4.3.1 for a detailed field description.

Transmit Descriptor Legacy	Description
STA	Status field See Section 3.4.3.2 for a detailed field description.
RSV	Reserved Should be written with 0b for future compatibility.
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 3.4.5 for more information about how the new descriptor format can be used to accomplish this task.

3.4.3.1 Transmit Descriptor Command Field Format

The CMD byte stores the applicable command and has fields shown in Table 3-14.

Note: Software must compute an offsetting entry to back out the bytes of the header that are not part of the IP pseudo header and should not be included in the TCP checksum and store it in the position where the hardware computed checksum is inserted.

Table 3-14. Transmit Command (TDESC.CMD) Layout

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.
RSV (bit 4)	Reserved Should be programmed to 0b.
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. When cleared, software should calculate the FCS for proper CRC check. There are several cases in which software must set IFCS: • Transmission of short packet while padding is enabled by the *TCTL.PSP* bit • Checksum offload is enabled by the IC bit in the TDESC.CMD • VLAN header insertion enabled by the *VLE* bit in the TDESC.CMD • Large Send or TCP/IP checksum offload using the context descriptor • CRC32 offload is used by setting the *ICRC32* bit in the TDESC.CMD
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.

3.4.3.2 Transmit Descriptor Status Field Format

The STATUS field stores the applicable transmit descriptor status and has the fields shown in Table 3-15.

The transmit descriptor status field is only present in cases where RS is set in the command field.

Table 3-15. Transmit Status Layout

TDESC.STATUS	Description
RSV (bit 3)	Reserved. Should be programmed to 0b.

TDESC.STATUS	Description
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).

Note: The DD bit reflects status of all descriptors up to and including the one with the RS bit set.

3.4.4 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 3-16. Special Field (TDESC.SPECIAL) Layout

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.

3.4.5 TCP/IP Context Transmit Descriptor Format

The TCP/IP context transmit descriptor provides access to the enhanced checksum and segmentation offload facilities that are available in the Ethernet controller. These features enable 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 contexts, they remain constant until they are modified by another context descriptor. This means that a context can (and will) 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 might 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 setup once and used for many frames. Only when some other frame type is required would a new context need to be loaded by software using a different index.

Note: For the **631xESB/632xESB** and the **82571EB/82572EI**, when operating with two descriptor queues, software needs to rewrite the context descriptor for each packet that requires it because software doesn't know if the second queue was modified. Hardware keeps track only for the last context descriptor that was written.

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 messages of the same type, TCP/IP for instance, and these messages also have the same Maximum Segment Size (MSS). In this instance, the same segmentation context could be used for multiple TCP messages that require hardware segmentation.

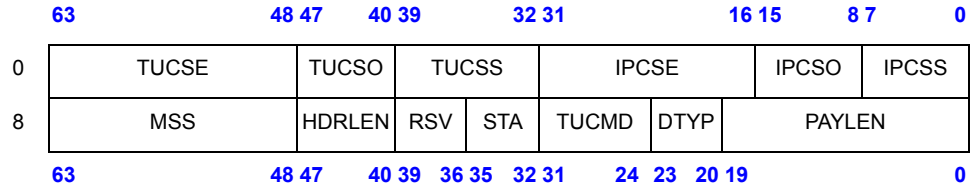
3.4.6 TCP/IP Context Descriptor Layout

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 3-17.

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 3-17. Transmit Descriptor (TDESC) Layout – (Type = 0000b)



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

Table 3-18. 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. **Note:** When the **631xESB/632xESB** is executing TCP segmentation offload the *TUCSE* field should be set to 0000h.
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 64 KB. 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. For the **631xESB/632xESB**: When executing checksum or TCP segmentation with IPv6 headers, the *IPCSE* field should be set to 0000h, IPCSS should be valid (as in IPv4 packets), and the *IXSM* bit in the data descriptor should be cleared. For proper IP checksum calculation, the *IP Header Checksum* field should be set to 0b unless some adjustment is needed by the software driver.
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.

Transmit Descriptor Offload	Description
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.
RSV	Reserved Should be programmed to 0b for future compatibility.
STA	TCP/UDP Status field Provides transmit status indication. Section 3.4.6.2 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 3.4.6.1 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 3.6 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.

3.4.6.1 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 3-19. Command Field (TDESC.TUCMD) Layout

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.
SNAP (Bit 6)	
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 3.4.6.2 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.
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.

Notes:

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.

3.4.6.2 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 bit is set in the command field.

Table 3-20. Transmit Status Layout

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.

3.4.7 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 3-21.

Table 3-21. 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	VLAN	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 3.4.7.1 for bit definitions of the DCMD field.
STA	TCP/IP Status field Provides transmit status indication. Section 3.4.7.2 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 3.4.7.3 provides the bit definition for the TDESC.POPTS field.
VLAN	VLAN field The 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).

3.4.7.1 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 3-22. Command Field (TDESC.DCMD) Layout

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.
RSV (bit 4)	Reserved Should be programmed to 0b.
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 Ethernet controller's internal packet buffer.
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.

Notes: 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).

3.4.7.2 TCP/IP Data Descriptor Status Field

Four bits are reserved to provide transmit status, although only the DD is valid. 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 3-23. Transmit Status Layout

TDESC.STA	Description
RSV (bit 3)	Reserved Should be programmed to 0b.
RSV (bit2)	Reserved Should be programmed to 0b.
RSV (bit 1)	Reserved Should be programmed to 0b.
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).

3.4.7.3 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 3-24. Packet Options Field (TDESC.POPTS) Layout

TDESC.POPTS	Description
RSV (bits 7:2)	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.

NOTE: Software should not set these fields for IPv6 packets.

3.4.7.4 TCP/IP VLAN Field

The VLAN 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 VLAN field is valid only when EOP is set.

Table 3-25. Special Field (TDESC.VLAN) Layout

TDESC.VLAN	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.

3.5 IP and TCP Transmit Checksum Offloading

The previous section on TCP Segmentation offload describes the IP and TCP 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 and TCP 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 3.4.5. 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 IP and TCP 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 IP and TCP 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.

3.5.1 Pipelined Tx Data Read Requests

Ethernet controllers support four pipelined requests from the Tx data DMA. Other DMA engines are not enhanced in a similar manner. In general, the four requests can belong to the same packet or to consecutive packets. However, the following restrictions apply:

- All requests for a packet are issued before a request is issued for a subsequent packet
- If a request (for the next packet) requires context change, the request for the next packet is not issued until the previous request completed (no pipeline across context).
- For the **631xESB/632xESB**, when executing with multiple Tx queues a context descriptor should be provided per packet.

For the **631xESB/632xESB**, if multiple Tx queues are used, read requests can be issued from any of the supported queues, as long as the above restrictions are met. Pipelined requests can belong to the same queue or to separate queues. However, as noted above, all requests for a certain packet must be issued (from same queue) before a request is issued for a different packet (potentially from a different queue).

The PCIe* specification does not insure that completions for separate requests return in-order. Read completions for concurrent requests are not required to return in the order issued. The Ethernet controller can handle completions that arrive in any order. Once all completions arrive for a given request, the Ethernet controller issues the next pending read data request.

The *MULR* bit in the Transmit Control Register (TCTL) enables multiple read requests for transmit data. Its initial value is loaded from the EEPROM/NVM *Multiple Read Request Enable* bit.

Ethernet controllers incorporate a 2 KB reorder buffer to support re-ordering of completions for four requests. Each request/completion can be up to 512 bytes long. The maximum size of a read request is defined as follows:

- When the MULR bit is set, maximum request size in bytes is the min {512, Max_Read_Request_Size}

In addition to the four pipeline requests from the Tx data DMA, The Ethernet controller can issue a single read request from each of the Tx descriptor and Rx descriptor DMA engines. The requests from the 3 DMA engines (Tx data, Tx descriptor and Rx descriptor) are independently issued. Each descriptor read request can fetch up to 16 descriptors (equal to 256 bytes of data).

3.6 TCP Segmentation

Hardware TCP Segmentation is one of the off-loading options of most modern TCP/IP stacks. 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.

Note: The **82573E/82573V/82573L** does not support multiple transmit queues.

3.6.1 Assumptions

The following assumptions apply 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.

3.6.2 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 software device 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).

3.6.2.1 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 as well as general efficiencies in the operation of the TCP Segmentation feature.

3.6.3 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.
- Interrupts are easily reduced to one per TCP message instead of one per packet.
- Fewer I/O accesses are required to command the hardware.

3.6.4 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, and TCP, including checksum, on a frame by frame basis.

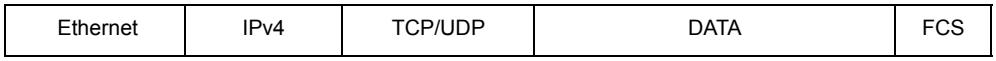


Figure 3-6. TCP/IP Packet Format

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
- TCP with options

Note: VLAN tag insertion is handled by hardware.

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.

Note: IP tunneled packets are not supported for Large Send operation.

3.6.5 TCP Segmentation Context Descriptor

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 3.4.6.

	63:48	47:40	39:32		31:16		15:8	7:0
0	TUCSE	TUCS0	TUCSS		IPCSE		IPCS0	IPCSS
8	MSS	HDRLEN	RSV	STA	TUCMD	DTYP	PAYLEN	

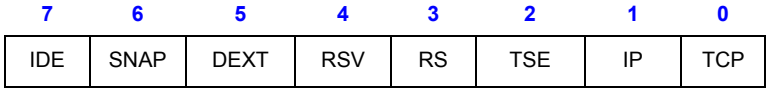


Figure 3-7. TCP/IP Context Transmit Descriptor & Command Layout

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.

Setting the *SNAP* bit enables Segmentation offload for SNAP packet type. When the *SNAP* bit is set, hardware updates the TCP/IP fields as well as the MAC header length field.

Note: When SNAP packets with a VLAN header are segmented, software must use the VLE option in the CMD field of the Data Descriptor. VLAN header insertion by software rather than use of the VLE option is not supported by the TCP segmentation offload mechanism for the SNAP packet type.

3.6.6 TCP Segmentation Data Descriptors

The TCP segmentation data descriptor is the companion to the TCP segmentation context descriptor described in the previous section. The descriptor is almost identical to the TCP/IP date descriptor. For a complete description of the descriptor please refer to Section 3.4.7.

To select this descriptor format, bit 29 (TDESC.DEXT) must be set to 1b and TDESC.DTYP must be set to 0001b.

3.6.7 TCP Segmentation Source Data

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:

- MAC Header (for SNAP)
  - MAC Header LEN field should be set to 0b

- 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 LAST packet of send
  - TCP Checksum should be set to the partial pseudo-header checksums as follows:

IP Source Address	
IP Destination Address	
Zero	
Zero	Next Header

Figure 3-8. TCP/IPv6 Partial Pseudo-Header Checksum

IP Source Address		
IP Destination Address		
Zero	Protocol ID	Zero

Figure 3-9. TCP/IPv4 Partial Pseudo-Header Checksum

3.6.8 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 and TCP 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. If all layer headers (L2-L4) are not coalesced into a single buffer, each buffer should not cross a 4 KB boundary, or be larger than the maximum read request size (see Section 13).

3.6.9 IP and TCP Headers

This section outlines the format and content for the IP and TCP 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.

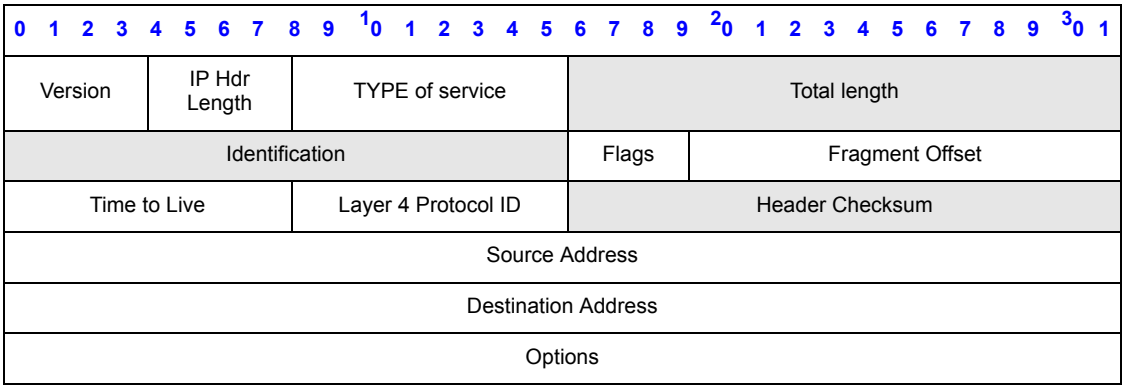


Figure 3-10. IPv4 Header (Traditional Representation)

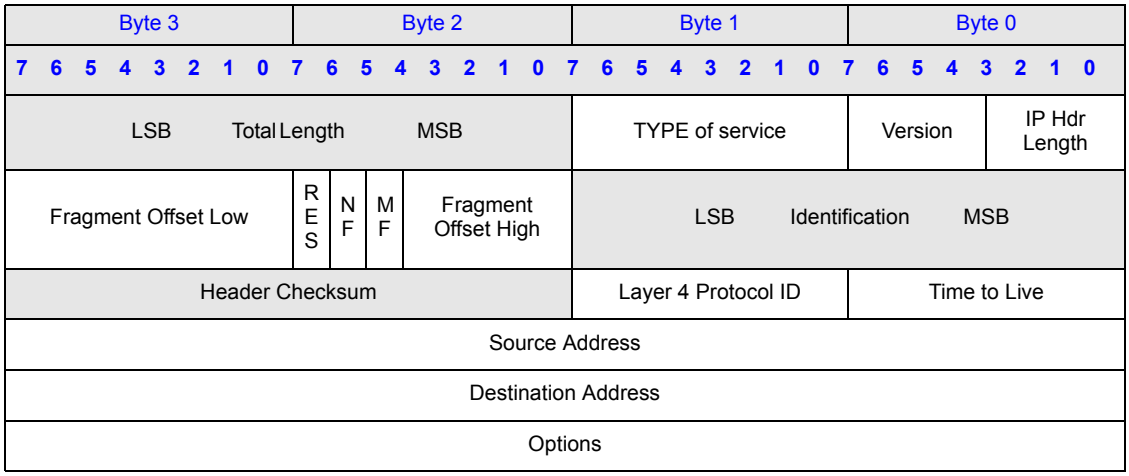


Figure 3-11. IPv4 Header (Little-Endian Order)

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.

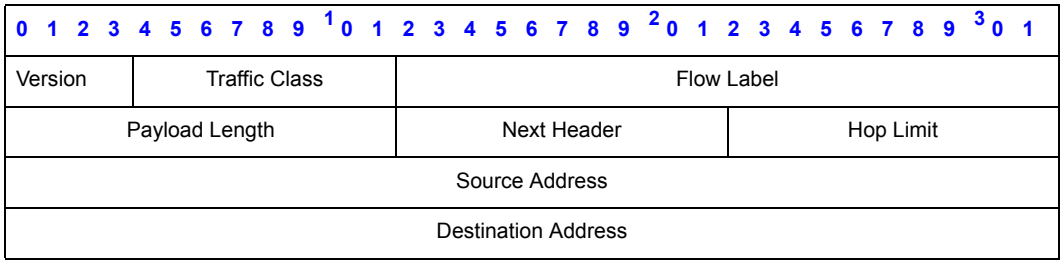


Figure 3-12. IPv6 TCP Header (Traditional Representation)

A TCP frame uses a 16 bit wide one's complement checksum. The checksum word is computed on the outgoing TCP header and payload, and on the Pseudo Header. Details on checksum computations are provided in Section 3.6. Note that TCP requires the use of checksum.

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.

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								
Source Port								Destination Port
Sequence Number								
Acknowledgement Number								
TCP Header Length	Reserved		URG	ACK	PSH	RST	SYN	FIN	Window
Checksum								Urgent Pointer
Options								

Figure 3-13. TCP Header (Traditional Representation)

Byte3								Byte2								Byte1								Byte0							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Destination Port																Source Port															
LSB Sequence Number MSB																															
Acknowledgement Number																															
Window																RES	URG	ACK	PSH	RST	SYN	FIN	TCP Header Length						Reserved		
Urgent Pointer																Checksum															
Options																															

Figure 3-14. 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 3-15 and Figure 3-16). 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.

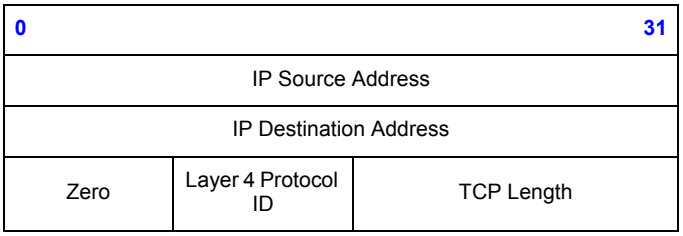


Figure 3-15. TCP Pseudo Header Content (Traditional Representation)

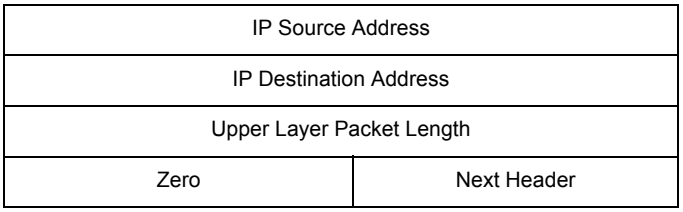


Figure 3-16. TCP Pseudo-Header Content for IPv6

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.

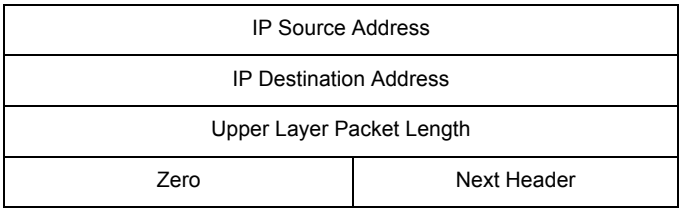


Figure 3-17. TCP Pseudo-Header Diagram for IPv6

3.6.10 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 3.6.10 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 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 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 and TCP headers of each frame. TCP checksums are a result of performing the checksum on all bytes of the payload and the pseudo header.

Two specific types of checksums are supported by the hardware in the context of the TCP Segmentation offload feature:

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

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

3.6.11 IP and TCP Header Updating

IP and TCP header is updated for each outgoing frame based on the IP and TCP header prototype that hardware transfers from the first descriptor(s) and stores on chip. The IP and TCP 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 3-18 illustrates the overall data flow.

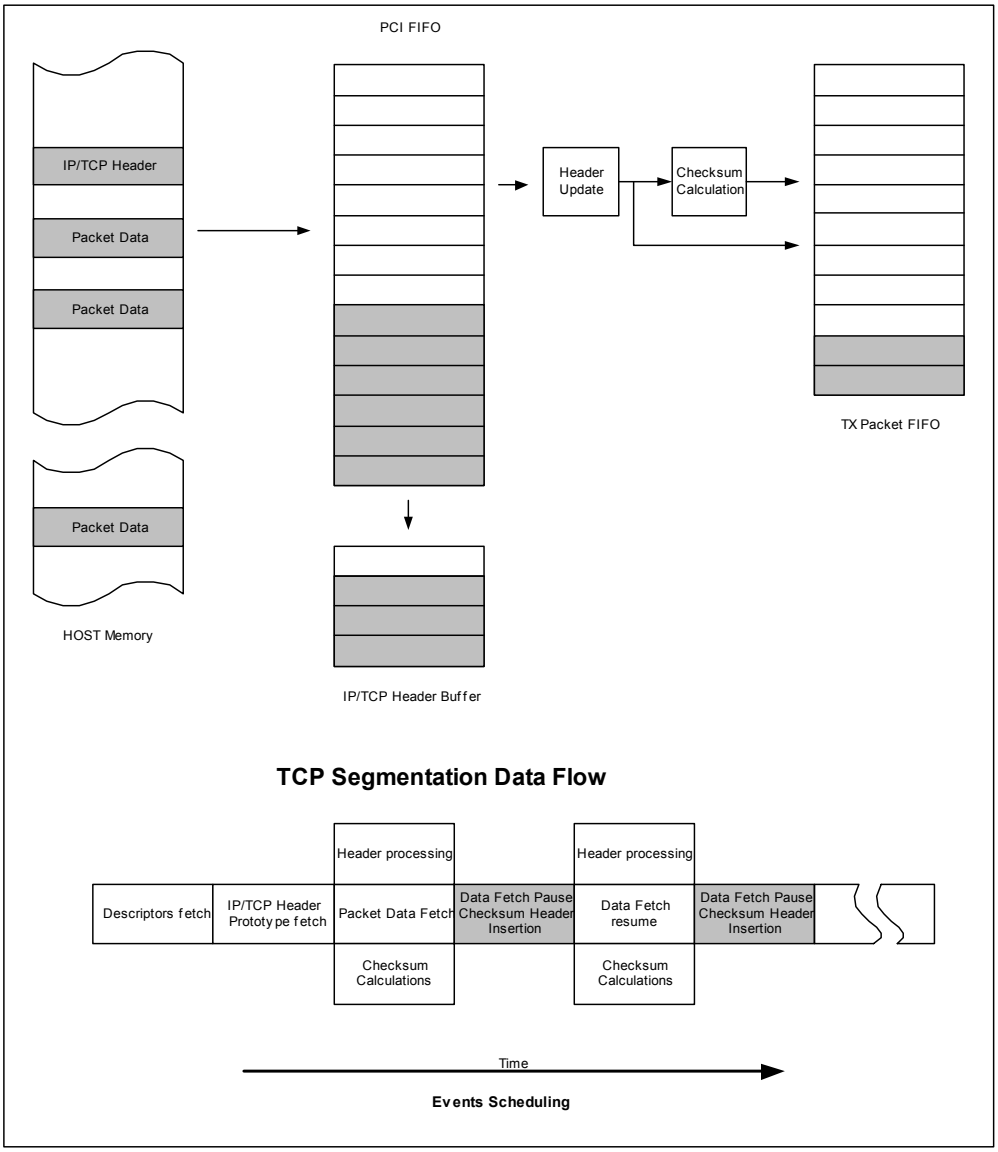


Figure 3-18. Overall Data Flow

3.6.11.1 IP and TCP 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.

- MAC Header (for SNAP)
  - — Type/Len field = MSS + HDRLEN - 14

- IPv4 Header
  - IP Total Length = MSS + HDRLEN – IPCSS
  - IP Checksum
- IPv6 Header
  - Payload Length = MSS + HDRLEN - IPCSS - IPv6Size (while IPv6Size = 40 Bytes
- 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

3.6.11.2 IP and TCP 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.

- MAC Header (for SNAP packets)
  - Type/LEN field = MSS + HDRLEN – 14
- IPv4 Header
  - IP Identification: incremented from last value (wrap around)
  - IP Total Length = MSS + HDRLEN – IPCSS
  - IP Checksum
- IPv6 Header
  - Payload Length = MSS + HDRLEN - IPCSS - IPv6Size (while IPv6Size = 40 bytes)
- 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

3.6.11.3 IP and TCP Header for the Last Frame

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

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

- MAC Header (for SNAP packets)
  - Type/LEN field = last frame payload bytes + HDRLEN – 14

- 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 = last frame payload bytes + HDRLEN - IPCSS - IPv6Size (while IPv6Size = 40 bytes
- 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

3.6.12 Limitations and Software Considerations

Packet headers should be arranged in a way such that either the whole TSO header is located in a single buffer or each one of the networking layers (L2, L3 and L4) is located in a separate buffer that doesn't cross a 4 KB boundary and is smaller than the maximum read request size (see Section 13).

Note: This page intentionally left blank.

PCIe* Local Bus Interface 4

4.1 Introduction

This section describes the software interface and some related hardware aspects of PCIe* in regard to the PCIe* Family of Gigabit Ethernet Controllers.

4.2 General Functionality

- **Native/Legacy Functionality**. The PCIe* capability register identifies the Ethernet controller and port type. In the Ethernet controller, all functions other than IDE (and UHCI for the **631xESB/632xESB** only) are defined as native devices. The IDE definition as either PCIe* native or legacy endpoint is determined by the Legacy Endpoint bit in the PCIe* initial configuration word 2 (EEPROM/NVM word 19h). For the **631xESB/632xESB** only, the UHCI definition is set by UHCI Device Type bit in the Functions Control 2 word (word 2Eh). By default, the Ethernet controller is a native device.
- **Locked Transactions**. The Ethernet controller does not support locked requests as a target or master.
- **End to End CRC**. This is not supported by the device

4.2.1 Message Handling (Receive Side)

Message packets are special packets that carry message code. The upstream device transmits special messages to the Ethernet controller by using this mechanism. The transaction layer decodes the message code and responds accordingly.

Table 4-1. Supported Messages on the Receive Side (Sheet 1 of 2)

Message Code [7:0]	Routing r2r1r0	Name	Device Response
14h	100	PM_Active_State_NAK	Internal signal set
19h	011	PME_Turn_Off	Internal signal set
41h	100	Attention_Indicator_On	Silently drop
43h	100	Attention_Indicator_Blink	Silently drop
40h	100	Attention_Indicator_Off	Silently drop
45h	100	Power_Indicator_On	Silently drop
47h	100	Power_Indicator_Blink	Silently drop
44h	100	Power_Indicator_Off	Silently drop
50h	100	Slot power limited support (one Dword)	Silently drop
7Eh	010,011,100	Vendor_Defined Type 0 (no data)	Unsupported request
7Eh	010,011,100	Vendor_Defined Type 0 (data)	Unsupported request

Table 4-1. Supported Messages on the Receive Side (Sheet 2 of 2)

Message Code [7:0]	Routing r2r1r0	Name	Device Response
7Fh	010,011,100	Vendor_Defined Type 1 (no data)	Silently drop
7Fh	010,011,100	Vendor_Defined Type 1 (data)	Silently drop
00h	011	Unlock	Silently drop

4.2.2 Message Handling (Transmit Side)

The transaction layer is also responsible for transmitting specific messages to report internal and external events such as interrupts and power management events.

Table 4-2. Supported Messages on the Transmit Side

Message Code [7:0]	Routing r2r1r0	Description
20h	100	Assert INT A
21h	100	Assert INT B
22h	100	Assert INT C
23h	100	Assert INT D
24h	100	De-assert INT A
25h	100	De-assert INT B
26h	100	De-assert INT C
27h	100	De-assert INT D
30h	000	ERR_COR
31h	000	ERR_NONFATAL
33h	000	ERR_FATAL
18h	000	PM_PME
1Bh	101	PME_to_Ack

4.2.3 Data Alignment

4.2.3.1 4 KB Boundary

Requests must not specify an address/length combination causing memory space access to cross a 4 KB boundary. It is the hardware responsibility to break requests into 4 KB aligned requests if required. This does not create any software requirement. However, if software allocates a buffer across the 4 KB boundary, hardware issues multiple requests for the buffer. Software should align buffers to the 4 KB boundary in cases where it improves performance.

The alignment to the 4 KB boundaries is done in the core. The transaction layer does not do any alignment according to these boundaries.

4.2.3.2 64 Bytes

It is also recommended that requests are multiples of 64 bytes and aligned to make better use of memory controller resources. This is also done in the core.

4.2.4 Transaction Attributes

4.2.4.1 Traffic Class and Virtual Channels

The Ethernet controller supports only Traffic Class 0 (default) and Virtual Channel 0 (default).

4.2.4.2 Relaxed Ordering

The Ethernet controller takes advantage of the relaxed ordering rules of the PCIe* Specification. By setting the relaxed ordering bit in the packet header, the Ethernet controller enables 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 write transactions, it also allows them to bypass each other in the path to system memory because the software device driver does not process this data until their associated descriptor writes are completed.
- The Ethernet controller cannot perform relax ordering for descriptor writes or an MSI write.

Relaxed ordering can be used in conjunction with the no snoop attribute to allow the memory controller to advance non-snoop writes ahead of earlier snooped writes.

Relaxed ordering is enabled in the Ethernet controller by setting the *RO_DIS* bit in the CTRL_EXT register.

4.2.4.3 Snoop Not Required

The Ethernet controller sets the *Snoop Not Required* attribute bit for master data writes. System logic can provide a separate path into system memory for non-coherent traffic. The non-coherent path to system memory provides a higher, more uniform bandwidth for write requests.

Note: The *Snoop Not Required* attribute does not alter transaction ordering. Therefore, to achieve maximum benefit from snoop not required transactions, it is advisable to also set the relaxed ordering attribute. This assumes that system logic supports both the snoop not required and relaxed ordering attributes.

Software configures no snoop support through the Device Control register and a set of *NONSNOOP* bits in the GCR register in the CSR space. The default value for all bits is disabled.

The Ethernet controller supports the no snoop bit for each relevant DMA client:

1. TXDSCR_NOSNOOP: Transmit Descriptor Read
2. TXDSCW_NOSNOOP: Transmit Descriptor Write
3. TXD_NOSNOOP: Transmit Data Read

4. RXDSCR_NOSNOOP: Receive Descriptor Read
5. RXDSCW_NOSNOOP: Receive Descriptor Write
6. RXD_NOSNOOP: Receive Data Write

All PCIe* functions in the Ethernet controller are controlled by this register.

4.2.5 Error Forwarding

If a TLP is received with an error forwarding trailer, the packet is dropped and is not delivered to its destination. The Ethernet controller does not initiate any additional master requests for that PCI function until it detects an internal reset or software reset. Software is able to access device registers after this occurs.

System logic is expected to trigger a system level interrupt to inform the operating system of the problem. The operating system has the ability to stop the process associated with the transaction, re-allocate memory instead of the faulty area, etc.

4.3 Flow Control

4.3.1 Flow Control Rules

The Ethernet controller implements only the Default Virtual Channel (VC0). A single set of credits is maintained for VC0.

Table 4-3. Flow Control Credit Allocation

Credit Type	Operations	Number of Credits
Posted Request Header (PH)	Target Write (1 unit) Message (1 unit)	2 units (allowing concurrent accesses to both LAN ports)
Posted Request Data (PD)	Target Write (length per 16 bytes = 1) Message (1 unit)	MAX_PAYLOAD_SIZE/16
Non-Posted Request Header (NPH)	Target Read (1 unit) Configuration Read (1 unit) Configuration Write (1 unit)	2 units (allowing concurrent target accesses to both LAN ports)
Non-Posted Request Data (NPD)	Configuration Write (1 unit)	2 units
Completion Header (CPLH)	Read Completion (not applicable)	Infinite (accepted immediately)
Completion Data (CPLD)	Read Completion (not applicable)	Infinite (accepted immediately)

The flow control update rules are as follows:

- The Ethernet controller maintains two credits for Non-Posted Request Data at any given time. The controller increments the credit by one after the credit is consumed and sends an UpdateFC packet as soon as possible. UpdateFC packets are scheduled immediately after a resource is available.
- The Ethernet controller provides two credits for Posted Request Header. For example, two credits are given for two concurrent target writes. The Ethernet controller also provides two credits for Non-Posted Request Header (two concurrent target reads). UpdateFC packets are scheduled immediately after a resource is available.
- The Ethernet controller follows the PCIe* recommendations for frequency of UpdateFC FCPs.

4.3.2 Upstream Flow Control Tracking

The Ethernet controller issues a master transaction only when the required flow control credits are available. Credits are tracked for posted, non-posted and completions. The later operates against a switch.

4.3.3 Flow Control Update Frequency

In any case, UpdateFC packets are scheduled immediately after a resource is available. When the Link is in the L0 or L0s link state, Update FCPs for each enabled type of non-infinite flow control credit must be scheduled for transmission at least once every 30 µs (-0% or +50%), except when the extended synchronize bit of the control link register is set. In this case, the limit is 120 µs (-0% or +50%).

4.3.4 Flow Control Timeout Mechanism

The Ethernet controller implements the optional flow control update timeout mechanism. This mechanism is activated when the link is in L0 or L0s link state. It uses a timer with a limit of 200 µs (0% or +50%), where the timer is reset by the receipt of any Initialize or Update FCP packet. Alternately, the timer can be reset by the receipt of any DLLP.

When the timer expires, the mechanism instructs the physical layer to retrain the link through the LTSSM recovery state.

Note: For the **631xESB/632xESB**, when external clock SER_CLK_IN is 62.5 MHz, the timeout is equal to 80 µs.

4.4 Host Interface

4.4.1 Tag IDs

PCIe* device numbers identify logical devices within the physical device. The Ethernet controller implements a single logical device with up to five separate PCI functions: LAN 0, LAN 1[1], IDE, Serial Port, and IPMI[1]/KCS[2]. The device number is captured from each type 0 configuration write transaction.

Note: The **631xESB/632xESB** implements two additional PCI functions: UHCI and BT.

Each of the PCIe* functions interfaces with the PCIe* unit through one or more clients. A client ID identifies the client and is included in the tag field of the PCIe* packet header. Completions always carry the tag value included in the request to allow routing of the completion to the appropriate client.

Client IDs are assigned as follows:

Table 4-4. Assignment of Client IDs

Tag	Flow : TLP Type : Usage
00h	RX : WR REQ : Data from Ethernet to main memory
01h	RX : RD REQ : To read descriptor to core
02h	RX : WR REQ : To write back descriptor from core to memory
04h	TX : RD REQ : To read descriptor to core
05h	TX : WR REQ : To write back descriptor from core to memory
06h	TX : RD REQ : To read descriptor to core second queue
07h	TX : WR REQ : To write back descriptor from core to memory second queue
08h	TX : RD REQ : Data 0 from main memory to Ethernet
09h	TX : RD REQ : Data 1 from main memory to Ethernet
0Ah	TX : RD REQ : Data 2 from main memory to Ethernet
0Bh	TX : RD REQ : Data 3 from main memory to Ethernet
0Ch	RX : RD REQ : To bring descriptor to core second queue
0Eh	RX : WR REQ : To write back descriptor from core to memory second queue
10h	MNG: RD REQ : Read data
11h	MNG: WR REQ : Write data
1Eh	MSI
1Fh	Message unit
Others	Reserved

1. Not applicable to the **82573E/82573V/82573L**.
2. Not applicable to the **82573L**.

4.4.2 Completion Timeout Mechanism

The Ethernet controllers support the completion timeout mechanism. Details relating to this feature can be found in the PCIe* specification.

4.5 Error Events and Error Reporting

The PCIe* Specification defines two error reporting paradigms:

- Baseline error reporting capability.
- Advanced error reporting capability.

The baseline error reporting capabilities are required of all PCIe* devices and define the minimum error reporting requirements. The advanced error reporting capability is defined for more robust error reporting and is implemented with a specific PCIe* capability structure. Both mechanisms are supported by the Ethernet controller.

Also, the SERR# enable and the parity error bits from the legacy command register take part in the error reporting and logging mechanism.

4.5.1 Error Events

Table 4-5lists the error events identified by the Ethernet controller and its response. The PCIe* Specification can be consulted for the effect on the PCI Status register.

Table 4-5. Response and Reporting of Error Events

Error Types	Error Events	Default Severity	Action
Physical Layer Errors			
Receiver Error	• 8b/10b decode errors. • Packet framing error.	• Correctable. • Send ERR_CORR.	• TLP → Initiate Nak; drop data. • DLLP → Drop.
Data Link Errors			
Bad TLP	• Bad CRC. • Illegal EDB. • Wrong sequence number.	• Correctable. • Send ERR_CORR.	• TLP → Initiate Nak; drop data.
Bad DLLP	Bad CRC.	• Correctable. • Send ERR_CORR.	DLLP → Drop.
Replay Timer Timeout	Replay timer expiration.	• Correctable. • Send ERR_CORR.	Follow LL rules.
Replay Number Rollover	Replay number rollover.	• Correctable. • Send ERR_CORR.	Follow LL rules.
Data Link Layer Protocol Error	Violations of flow control initialization protocol. Received ACK/NACK not corresponding to any TLP.[a]	Uncorrectable. Send ERR_FATAL.	Follow LL rules.[a]

Table 4-5. Response and Reporting of Error Events

Error Types	Error Events	Default Severity	Action
TLP Errors			
Poisoned TLP Received	TLP with error forwarding.	• Uncorrectable. • ERR_NONFATAL. • Log header.	In case of Poisoned completion, no more requests from this client.
Unsupported Request (UR)	• Wrong configuration access. • MRdLk. • Configuration request type1. • Unsupported vendor. Defined type 0 message: • Invalid MSG code. • Unsupported TLP type. • Wrong function number. • Wrong traffic class or virtual channel. • Received target access with data size larger than 64 bits. • Received TLP outside address range.	• Uncorrectable. • ERR_NONFATAL. • Log header.	Send completion with UR.
Completion Timeout	Completion Timeout timer expired.	• Uncorrectable. • ERR_NONFATAL.	Send the read request again
Completer Abort (CA)	Attempts to write to the FLASH device when writes are disabled (FWE not equal to 10b).	• Uncorrectable. • ERR_NONFATAL. • Log header.	Send completion with CA.
Unexpected Completion	Received completion w/o a request for it (tag, ID, etc.).	• Uncorrectable. • ERR_NONFATAL. • Log header.	Discard TLP.
Receiver Overflow	Received TLP beyond allocated credits.	• Uncorrectable. • ERR_FATAL.	Receiver behavior is undefined.
Flow Control Protocol Error	• Minimum initial flow control advertisements. • Flow control update for infinite credit advertisement.	• Uncorrectable. • ERR_FATAL.	Receiver behavior is undefined.

Table 4-5. Response and Reporting of Error Events

Error Types	Error Events	Default Severity	Action
Malformed TLP (MP)	• Data payload exceeds maximum payload size. • Received TLP data size does not match length field. • TD field value does not correspond with the observed size. • Byte enables violations. • PM messages that do not use TC0. • Usage of unsupported VC.	• Uncorrectable. • ERR_FATAL. • Log header.	Packet dropped. Free flow control credits.
Completion with Unsuccessful Completion Status		No action (already done by originator of completion).	Free FC credits.
Byte-count Integrity in completion process[a]	When the byte count isn't compatible with the length field and the actual expected completion length. For example, length field is 10 (in Dword), actual length is 40, but the byte-count field that indicates how many bytes are still expected is smaller than 40, which is not reasonable.	No action.	The **82571EB/82572EI** doesn't check for this error and accepts these packets. **Note:** This might cause a completion timeout condition.

a. **82571EB/82572EI** only.

4.5.2 Error Pollution

Error pollution can occur if error conditions for a given transaction are not isolated to the error's first occurrence. If the Physical Layer detects and reports a Receiver Error, to avoid having this error propagate and cause subsequent errors at upper layers, the same packet is not signaled at the Data Link or Transaction layers.

Similarly, when the Data Link Layer detects an error, subsequent errors, which occur for the same packet, are not signaled at the Transaction Layer.

4.5.3 Unsuccessful Completion Status

A completion with an unsuccessful completion status is dropped and not delivered to its destination. The request that corresponds to the unsuccessful completion is retried by sending a new request for the undeliverable data.

4.6 Link Layer

4.6.1 ACK/NAK Scheme

The Ethernet controller supports two alternative schemes for ACK/NAK rate:

- ACK/NAK is scheduled for transmission following any TLP.
- ACK/NAK is scheduled for transmission according to time-outs specified in the PCIe* Specification.

The PCIe* error recovery bit loaded from the EEPROM determines which of the two schemes is used.

4.6.2 Supported DLLPs

The following DLLPs are supported by the Ethernet controller as a receiver.

Table 4-6. DLLPs Received

Ack	
Nak	
PM_Request_Ack	
InitFC1-P	v2v1v0 = 000
InitFC1-NP	v2v1v0 = 000
InitFC1-Cpl	v2v1v0 = 000
InitFC2-P	v2v1v0 = 000
InitFC2-NP	v2v1v0 = 000
InitFC2-Cpl	v2v1v0 = 000
UpdateFC-P	v2v1v0 = 000
UpdateFC-NP	v2v1v0 = 000
UpdateFC-Cpl	v2v1v0 = 000

The following DLLPs are supported by the Ethernet controller as a transmitter.

Table 4-7. DLLPs Initiated

Ack	
Nak	
PM_Enter_L1	
PM_Enter_L23	
PM_Active_State_Request_L1	
InitFC1-P	v2v1v0 = 000
InitFC1-NP	v2v1v0 = 000
InitFC1-Cpl	v2v1v0 = 000
InitFC2-P	v2v1v0 = 000

Table 4-7. DLLPs Initiated

InitFC2-NP	v2v1v0 = 000
InitFC2-Cpl[a]	v2v1v0 = 000
UpdateFC-P	v2v1v0 = 000
UpdateFC-NP	v2v1v0 = 000

a. UpdateFC-Cpl is not transmitted due to the infinite FC-CPL allocation.

4.6.3 Transmit EDB Nullifying

In case of a retrain, there is a need to guarantee that no abrupt termination of the transmit packet occurs. For this reason, early termination of the transmitted packet is possible. This is accomplished by appending EDB to the packet.

4.7 Physical Layer

4.7.1 Link Width

Note: The **82573E/82573V/82573L** only supports a link width of x1.

The Ethernet controller link width supports 4 pairs of transmit and receive differential lanes. This is notated as "x4" and is read "by 4 link." The aggregate bandwidth is expected to meet Ethernet controller performance requirements. The **82571EB/82572EI** also supports x2 width configuration. The Ethernet controller native link width is loaded from the EEPROM to the Maximum Link Width field of the PCIe* capability register, LCAP[11:6].

During link configuration, the upstream device and Ethernet controller negotiate on a common link width. The link width must be one of the supported PCIe* link widths (x1, x2[1], or x4). The link is configured to the largest value supported by the MCH and the Ethernet controller.

Table 4-8. PCIe* Link Width Configurations

Initial Link Width	Upstream Initial Link Width	Negotiated Link Width
x4	x4 or wider	x4
x2[a]	x2 or wider[a]	x1, x2[a]
x1	any width	x1
other	other	illegal

a. **82571EB/82572EI** only.

1. **82571EB/82572EI** only.

4.7.2 Performance Monitoring

The Ethernet controller incorporates PCIe* performance monitoring counters to provide common capabilities for evaluate performance. It implements four 32-bit counters to correlate between concurrent measurements of events as well as the sample delay and interval timers. The four 32-bit counters can also operate in a two 64-bit mode to count long intervals or payloads.

The list of events supported by the Ethernet controller and the counters control bits are described in the memory register map.

4.7.3 Configuration Registers

4.7.3.1 PCI Compatibility

PCIe* is completely compatible with existing deployed PCI software. To achieve this, PCIe* hardware implementations conform to the following requirements:

- All devices must be supported by deployed PCI software and must be enumerable as part of a tree through PCI device enumeration mechanisms.
- Devices must not require any resources such as address decode ranges and interrupts beyond those claimed by PCI resources for operation of software compatible and software transparent features with respect to existing deployed PCI software.
- Devices in their default operating state must confirm to PCI ordering and cache coherency rules from a software viewpoint.
- PCIe* devices must conform to PCI power management specifications and must not require any register programming for PCI compatible power management beyond those available through PCI power management capability registers. Power management is expected to conform to a standard PCI power management by existing PCI bus drivers.

PCIe* devices implement all registers required by the PCI Specification as well as the power management registers and capability pointers specified by the PCI power management specification. In addition, PCIe* defines a PCIe* capability pointer to indicate support for PCIe* extensions and associated capabilities.

The Ethernet controller is a multi-function device with the following functions:

Function Number	Function Description
0 or 1 (0 only for the **82573E/82573V/ 82573L**)	LAN 0
1 or 0	LAN 1[a]
2	IDE Function for remote boot and installations.
3	Serial Port for remote keyboard and text redirection.
4 or 0	IPMI[a]/KCS[b] interface for enhanced manageability functionality.
631xESB/632xESB	
5	USB 1.1 host controller. Implements UHC1 compatible interface.
6	Reserved.
7	BT interface for performance enhanced manageability functionality.

a. Not applicable to the **82573E/82573V/82573L**.
b. Not applicable to the **82573L**.

The LAN0 and LAN1 are shown in PCI functions 0 and PCI functions 1, respectively. The LAN Function Select field in EEPROM word 21h is reflected in the status register and determines if LAN0 appears in PCI function 0 or PCI function 1. LAN1 appears in the complementary PCI function. If only a single LAN is enabled (controlled by the strapping signals: LAN0_DIS_N or LAN1_DIS_N) then it is shown in the PCI function 0 regardless of the LAN Function Select field in the EEPROM. If both LANs are disabled and the IPMI/KCS is enabled (controlled by the enable field in EEPROM word 21h), then the IPMI/KCS appears in PCI function 0. The tables that follow list the mapping device functions into the PCI functions according to the enabled functions.

PCI Function Number	Function 0	Function 1[a]	Function 2	Function 3	Function 4	Function 5[b]	Function 7[b]
All functions are enabled and the LAN Function Select is 0.	LAN 0	LAN 1	IDE (if enabled)	Comm (if enabled)	IPMI[a]/ KCS[c] (if enabled)	UHCI (if enabled)	BT (if enabled)
All functions are enabled and the LAN Function Select is 1[a].	LAN 1[a]	LAN 0	IDE (if enabled)	Comm (if enabled)	IPMI[a]/ KCS[c] (if enabled)	UHCI (if enabled)	BT (if enabled)

PCI Function Number	Function 0	Function 1[a]	Function 2	Function 3	Function 4	Function 5[b]	Function 7[b]
LAN 0 or LAN 1 is disabled and the LAN Function Select bit is either 0 or 1.	LAN	-	IDE (if enabled)	Comm (if enabled)	IPMI[a]/ KCS[c] (if enabled)	UHCI (if enabled)	BT (if enabled)
Both LAN functions are disabled and the IPMI/KCS is enabled[a].	IPMI/KCS	-	IDE (if enabled)	Comm (if enabled)	-	UHCI (if enabled)	BT (if enabled)
Both LAN functions are disabled and the IPMI/KCS is disabled[a].	All PCI functions are disabled. Ethernet controller is in deep power down.						

a. Not applicable to the **82573E/82573V/82573L**.
b. **631xESB/632xESB** only.
c. Not applicable to the **82573L**.

Note: For the **631xESB/632xESB**, UHCI and BT functions are always activated as functions 5 and 7, respectively. However, UHCI can be disabled through either an EEPROM bit or fuse. BT can be disabled through an EEPROM bit only.

All functions contain the following regions of the PCI configuration space:

- Mandatory PCI configuration registers
- Power management capabilities
- MSI capabilities
- PCIe* extended capabilities

4.7.4 Mandatory PCI Configuration Registers

The PCI configuration registers map follows. Registers of the LAN functions that have changed relative to earlier Gigabit Ethernet controllers are marked in ***bold italics***. Initial values of the configuration registers are marked in parenthesis.

Configuration registers are assigned one of the attributes listed in the following table.

RD/WR	Description
RO	Read only register. Register bits are read only and cannot be altered by software.
RW	Read/Write register. Register bits are read or write and may be either set or reset.
R/W1C	Read only status / Write 1b to Clear register. Writing a 0b to R/W1C bits has no effect.
ROS	Read only register with Sticky Bits. Register bits are read only and cannot be altered by software. Bits are not cleared by reset and can only be reset with the PWRGOOD signal. Devices that consume AUX power are not allowed to reset sticky bits when AUX power consumption (either through AUX power or PME enable) is enabled.
RWS	Read/Write with Sticky Bits: Register bits are read or write and might be either set or reset by software to the desired state. Bits are not cleared by a reset and can only be reset with the PWRGOOD signal. Devices that consume AUX power are not allowed to reset sticky bits when AUX power consumption (either through AUX power or PME enable) is enabled.

RD/WR	Description
R/W1CS	Read only status / Write 1b to Clear with Sticky Bits. Register bits indicate status when read. A set bit indicates a status event may be cleared by writing a 1b. Writing a 0b to R/W1C bits has no effect. Bits are not cleared by reset and can only be reset with the PWRGOOD signal. Devices that consume AUX power are not allowed to reset sticky bits when AUX power consumption (either through via AUX power or PME enable) is enabled.
HwInit	Hardware Initialized. Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM/NVM. Bits are read only after initialization and can only be reset (for write once by firmware) with the PWRGOOD signal.
RsvdP	Reserved and Preserved: This is reserved for future implementations. Software must preserve the value read for writes to bits.
RsvdZ	Reserved and 0b. This is reserved for future R/W1C implementations. Software must use 0b for writes to bits.

The functions have a separate enabling mechanism. A function that is not enabled does not function and does not expose its PCI configuration registers.

Function	Default	Initial EEPROM/NVM Address
LAN 0	1b	Strapping Option / EEPROM word 10h, bit 11. For the **82573E/82573V/82573L**, always enabled.
LAN 1[a]	1b	Strapping Option / EEPROM word 20h, bit 11.
IDE	0b	EEPROM address 21h. For the **82573E/82573V/82573L**, IDE Ena field in NVM word 49h.
Serial Port	0b	EEPROM word 21h. For the **82573E/82573V/82573L**, Serial Ena field in NVM word 49h.
IPMI[a]/KCS[b]	0b	EEPROM work 21h. For the **82573E/82573V**, KCS Ena field in NVM word 49h.
UHCI[c]	0b	Strapping Option / EEPROM word 2Eh, bit 15.
BT[c]	0b	EEPROM word 2Eh, bit 13.

a. Not applicable to the **82573E/82573V/82573L**.
b. Not applicable to the **82573L**.
c. **631xESB/632xESB** only.

Table 4-9. PCI Compatible Configuration Registers

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0h	***Device ID***		***Vendor ID (8086h)***	
4h	***Status Register (0010h)***		***Command Register (0000h)***	
8h	Class Code (020000h, 010185h, 070002h, 0C0701h)			Revision ID (00h)
Ch	BIST (00h)	Header Type (00h, 80h)	***Latency Timer (00h)***	***Cache Line Size (10h)***
10h	Base Address 0			
14h	Base Address 1			
18h	Base Address 2			

Table 4-9. PCI Compatible Configuration Registers

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
1Ch	Base Address 3			
20h	Base Address 4			
24h	Base Address 5			
28h	CardBus CIS Pointer (00000000h)			
2Ch	Subsystem ID (0000h)		Subsystem Vendor ID (8086h)	
30h	Expansion ROM Base Address			
34h	Reserved (000000h)			Cap_Ptr (C8h)
38h	Reserved (00000000h)			
3Ch	***Max_Latency (00h)***	***Min_Grant (00h)***	Interrupt Pin (01h)	Interrupt Line (00h)

NOTE: The following color notation is used for reference:

	Fields identical to all functions.
	Read only fields.
	Hard coded fields.

Interpretation of the various Ethernet controller registers is provided as follows.

Vendor ID. This is a read only register that has the same value for all PCI functions. It identifies uniquely Intel products. The field can be automatically loaded from the EEPROM/NVM at address 0Eh during initialization with a default value of 8086h.

Device ID. This is a read only register. It has the same value for the two LAN functions. This field identifies uniquely the Ethernet controller functions. The field can be automatically loaded from the EEPROM/NVM during initialization with the one of the following default values:

PCI Function	Default Value	EEPROM/ NVM Address	Description
LAN 0	105Eh/ 1081h[a] 1082h[a] 1083h[a] 1096h[a] 1097h[a] 1098h[a]	0Dh	Dual port 10/100/1000 Mb/s Ethernet controller, x4 PCIe*, copper. Dual port Mb/s Ethernet controller, fiber media from SerDes port. Dual port Mb/s Ethernet controller, 1000BASE-X backplane. Dual port 10/100/1000 Mb/s Ethernet controller, copper interface with Intel I/O Acceleration Technology. Dual port Mb/s Ethernet controller, fiber media from SerDes port with Intel I/O Acceleration Technology. Dual port Mb/s Ethernet controller, 1000BASE-X backplane with Intel I/O Acceleration Technology.
LAN 0	108Bh/ 108Ch[b] (109Ah **82573L**)	0Dh	10/100/1000 Mb/s Ethernet controller, x1 PCIe*, copper. Not applicable to the **631xESB/632xESB**.
LAN 1[c]	Same as LAN 0	11h	Same as LAN 0.
IDE	1061h 108D[d] 1084h[a]	102h/42h 43h[d]	Ethernet controller IDE manageability function.
Serial Port	1062h 108Fh[d] 1085h[a]	103h/43h 44h[d]	Ethernet controller Serial Port manageability function.
IPMI[c]/KCS[e]	1063h 108E[d] 1086[a]	104h/44h 0Dh[d]	Ethernet controller IPMI[b]/KCS[e] manageability function.
UHCI[a]	1087h	110h/50h	UHCI HUSB host controller function.
BT[a]	1089h	112h/52h	IPMI/BT manageability function.

a. **631xESB/632xESB** only.
b. The LAN function default Device ID is determined by the PT Enable fuse. Device ID is 108Ch for the **82573E** (PT Enable fuse = 1b) and is 108Bh for **82573V** (PT Enable fuse =0b)
c. Not applicable to the **82573E/82573V/82573L**.
d. **82573E/82573V/82573L** only.
e. **82573E/82573V** only.

Command. This is a read/write register. Its layout is shown below. Shaded bits are not used by this implementation and are hard wired to 0b. Each of the 5[1] functions have their own Command register. Unless explicitly specified, functionality is the same in all functions.

1. 4 functions for the **82573E/82573V/82573L**; 7 functions for the **631xESB/632xESB**.

Bit(s)	Initial Value	Description
15:11	0b	Reserved.
10	0b	Interrupt Disable.[a]
9	0b	Fast Back-to-Back Enable. Hardwired to 0b.
8	0b	SERR# Enable.
7	0b	Wait Cycle Enable. Hardwired to 0b.
6	0b	Parity Error Response.
5	0b	Palette Snoop Enable. Hardwired to 0b.
4	0b	MWI Enable. Hardwired to 0b.
3	0b	Special Cycle Monitoring. Hardwired to 0b.
2	0b	Enable Mastering: LAN 0: R/W field LAN 1: R/W field[b] IDE: R/W field Serial Port: R/W field IPMI[b]/KCS[c]: R/W field UHCI[d]: R/W field BT[d]: R/W field
1	0b	Memory Access Enable.
0	0b	I/O Access Enable.

a. The Interrupt Disable register bit is a read-write bit that controls the ability of a PCIe* device to generate a legacy interrupt message. When set, devices are prevented from generating legacy interrupt messages.
b. Not applicable to the **82573E/82573V/82573L**.
c. Not applicable to the **82573L**
d. **631xESB/632xESB** only.

Status Register. Shaded bits are not used by this implementation and are hardwired to 0b. Each of the 5 functions have their own status register. Unless explicitly specified, functionality is the same in all functions.

Bit(s)	Initial Value	RD/WR	Description
15	0b	R/W1C	Detected Parity Error.
14	0b	R/W1C	Signaled System Error.
13	0b	R/W1C	Received Master Abort.
12	0b	R/W1C	Received Target Abort.
11	0b	R/W1C	Signaled Target Abort.
10:9	00b		DEVSEL Timing. Hardwired to 0b.
8	0b	R/W1C	Data parity reported.
7	0b		Fast Back-to-Back Capable. Hardwired to 0b.
6	0b		Reserved.
5	0b		66 MHz Capable. Hardwired to 0b.

Bit(s)	Initial Value	RD/WR	Description
4	1b	RO	New Capabilities. This indicates that a device implements Extended Capabilities. The Ethernet controller sets this bit and implements a capabilities list indicating it support for PCI Power Management, message signaled interrupts, and the PCIe* extensions.
3	0b	RO	Interrupt Status.[a]
2:0	0b		Reserved.

a. The Interrupt Status field is a RO field that indicates that an interrupt message is pending internally to the device.

Revision. The default revision ID of this device is 03h (01h for the **631xESB/632xESB**, 00h for the **82573L**, and 07h for the **82573E/82573V**). The value of the revision ID is a logic eXclusive OR (XOR) between the default value and the value in EEPROM/NVM word 1Dh (1Eh for the **82573E/82573V/82573L**). For the **631xESB/632xESB**, the value of the revision ID is loaded from the EEPROM.

Note: LAN 0 and LAN 1 functions have the same revision ID. For the **82573E/82573V/82573L**, the revision ID is identical for all functions.

Class Code. The class code is a read only. Hard coded values that identify the device functionality:

LAN 0 or LAN 1	020000h	Ethernet Adapter
IDE	010185h[a]	
Serial Port	070002h	16550 COM Port Compatible
IPMI/KCS	0C0701h[b]	
IPMI/BT[c]	0C0702h[b]	
UHCI[c]	0C0300h	
iSCSI boot[c]	010000h	**Note:** The Class Code for the LAN can be configured to be iSCSI boot, depending on the EEPROM bits 8 and 9 in word 29h (LAN_0/1_iSCSI_enable).

a. The IDE Class Code is composed of the following fields:
Offset 09h = 85h, programming interface class.
Offset 0Ah = 01h, subclass code meaning IDE controller.
Offset 0Bh = 01h, base class code meaning mass storage controller.

b. The IPMI Class Code is composed of the following fields (not applicable to the **82573E/82573V/82573L**):
Base Class 0Ch
Subclass 07h
Interface Codes = 00h for IPMI SMIC; 01h for IPMI keyboard controller style; 02h for IPMI block transfer interface.

c. **631xESB/632xESB** only.

Cache Line Size. This field is implemented by PCIe* devices as a read/write field. It is loaded from EEPROM/NVM word 1Ah. All functions are initialized to the same value.

Latency Timer. The Ethernet controller does not use this and this bit is hardwired to 0b.

Header Type. This indicates if an Ethernet controller is single function or multifunction. If a single function is the only active one then this field has a value of 00h to indicate a single function Ethernet controller. If other functions are enabled then this field has a value of 80h to indicate a multi-function Ethernet controller.

BIST. The Built In Self Test (BIST) is not supported from the PCI configuration space in the Ethernet controller.

Base Address Registers.

The Base Address Registers (BARs) are used to map the Ethernet controller register space of the various functions.

BARs are defined as non-prefetchable. As a result, only 32-bit addressing is supported.

For the **82571EB/82572EI**, LAN Flash sizes can be in the range of 64 KB to 8 MB, depending on the Flash size field in EEPROM word 0Fh. The Memory Flash BAR has the following characteristics:

Flash Size	Valid Bits (R/W)	Zero Bits (0)
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
1 MB	63/31:20	19:4
2 MB	63/31:21	20:4
4 MB	63/31:22	21:4
8 MB	63/31:23	22:4

32-bit Addressing Mode. The BAR32 bit in EEPROM word 0Ah is set to 1b.

Note: For the **82573E/82573V/82573L**, the default setting of the Flash BAR enables software to execute initial programming of empty (non-valid) Flash via the (parallel) Flash BAR.

Table 4-10. LAN 0 and LAN 1 Functions

BAR	Address	Bits 31:4	Bit 3	Bit 2	Bit 1	Bit 0
0	10h	Memory BAR (R/W - 31:17; 0 - 16:4)	0b	0b	0b	0b
1	14h	Flash BAR (R/W - 31:23/16; 0 - 22/15:4)[a]	0b	0b	0b	0b
2	18h	IO BAR (R/W - 31:5; 0 - 4:1)			0b	1b
3	1Ch	Reserved (read as 0b)				
4	20h	Reserved (read as 0b)				
5	24h	Reserved (read as 0b)				

a. LAN Flash sizes can be in the range of 64 KB to 8 MB, depending on the Flash size field in EEPROM word 0Fh.

For the **82573E/82573V/82573L**, the Flash BAR is set by hardware to a default to 1MB if connected to external Flash as defined by the NVMT strapping pins. In case the Flash contains valid data, the Flash BAR is gated by the *DISLFB* field in word 4Ch in the NVM and its size is defined by the NVSize in word 0Fh in the NVM.

Table 4-11. IDE Function

BAR	Address	Bits 31:4	Bit 3	Bit 2	Bit 1	Bit 0
0	10h	Primary Command Block IO BAR (R/W - 31:3)		0b	0b	1b
1	14h	Primary Control Block IO BAR (R/W - 31:2)			0b	1b
2	18h	Secondary Command Block IO BAR (R/W - 31:3)		0b	0b	1b
3	1Ch	Secondary Control Block IO BAR (R/W - 31:2)			0b	1b
4	20h	Bus Master IO BAR (R/W - 31:4)	0b	0b	0b	1b
5	24h	Reserved (read as 0b)				

Table 4-12. Serial Port Function

BAR	Address	Bits 31:4	Bit 3	Bit 2	Bit 1	Bit 0
0	10h	IO BAR (R/W - 31:3)		0b	0b	1b
1	14h	Memory BAR (R/W - 31:12; 0 - 11:4)	0b	0b	0b	0b
2	18h	Reserved (read as 0b)				
3	1Ch	Reserved (read as 0b)				
4	20h	Reserved (read as 0b)				
5	24h	Reserved (read as 0b)				

Table 4-13. IPMI[a]/KCS Function

BAR	Address	Bits 31:4	Bit 3	Bit 2	Bit 1	Bit 0
0	10h	Memory BAR (R/W - 31:12; 0 - 11:4)	0b	0b	0b	0b
1	14h	IO Register Base Address (bits 31:2)			0b	1b
2	18h	Reserved (read as 0b)				
3	1Ch	Reserved (read as 0b)				
4	20h	Reserved (read as 0b)				
5	24h	Reserved (read as 0b)				

a. Not applicable to the **82573E/82573V/82573L**.

All base address registers have the following fields:

Table 4-14. UHCI Function[a]

BAR	Address	Bits 31:4	Bit 3	Bit 2	Bit 1	Bit 0
0	10h	Memory BAR (R/W - 31:12; 0b - 11:4)	0b	0b	0b	0b
1	14h	Reserved (read as 0b)				
2	18h	Reserved (read as 0b)				
3	1Ch	Reserved (read as 0b)				
4	20h	IO BAR (R/W 31:5; 0b - 4)	0b	0b	0b	1b
5	24h	Reserved (read as 0b)				

a. **631xESB/632xESB** only.

Table 4-15. BT Function[a]

BAR	Address	Bits 31:4	Bit 3	Bit 2	Bit 1	Bit 0
0	10h	Memory BAR (R/W - 31:12; 0b - 11:4)	0b	0b	0b	0b
1	14h	IO BAR (R/W – 32:2)			0b	1b
2	18h	Reserved (read as 0b)				
3	1Ch	Reserved (read as 0b)				
4	20h	Reserved (read as 0b)				
5	24h	Reserved (read as 0b)				

a. **631xESB/632xESB** only.

All base registers have the following fields:

Field	Bit(s)	RD/WR	Initial Value	Description
Memory Address Space[a]	63[a]/31:4	R/W	0b	These are read/write bits hardwired to 0b depending on the memory mapping window sizes. LAN Memory spaces = 128 KB. LAN Flash spaces[b] = 64 KB to 8 MB in power of 2. (The mapping window size is set by EEPROM word 0Fh.) IDE, Serial Port and IPMI/KCS, IPMI/BT[c], and UHCI[c] Memory Spaces = 4 KB.
I/O Address Space	31:2	R/W	0b	These are read/write bits hardwired to 0b depending on the I/O mapping window sizes. LAN I/O Spaces = 32 bytes. Serial Port I/O Space = 8 bytes. IPMI[a]/KCS I/O Space = 4 bytes. IDE I/O Spaces. 2 x CMD Block = 8 bytes. 2 x Control Block = 4 bytes. Bus Master Space is = 16 bytes. UHCI IO space = 32 bytes.[c] BT IO space = 4 bytes.[c]
Prefetch Memory	3	R	0b	The Ethernet controller implements non-prefetchable space due to side effects of read transactions. 0b = non-prefetchable space 1b = prefetchable space
Memory Type	2:1	R	32-bit = 00b[d] 64-bit = 10b	This field indicates the address space size. 00b = 32-bit[d] 10b = 64-bit
Memory	0	R	Memory = 0b I/O = 1b	If this bit equals 0b, it indicates memory space. If it equals 1b, it indicates input/output.

a. Not applicable to the **82573E/82573V/82573L**.
b. For the **82573E/82573V/82573L**, 64 KB to 4 MB in power of 2. (The mapping window size is set by the *NVSize* field in NVM word 0Fh.) The Flash Bar is enabled by the *DISLFB* field in the NVM at word 4Ch.
c. **631xESB/632xESB** only.
d. **82573E/82573V/82573L** only.

Table 4-16. Memory & I/O Mapping (32-bit Addressing Mode)

Function	Mapping Window	Mapping Description
LAN 0 LAN 1[a]	Memory BAR 0	The internal registers and memories are accessed as direct memory mapped offsets from the base address register. Software accesses can be byte, word, Dword or 64 bytes.
	Flash BAR 1	The external Flash can be accessed using direct memory mapped offsets from the Flash BAR. Software accesses can be byte, word, Dword or 64 bytes.
	I/O BAR 2	All internal registers, memories, and Flash can be accessed using I/O operations. There are two 4-byte registers in the I/O mapping window: Address Register and Data Register. Software accesses to internal registers must be Dwords only. Access to the Flash can be byte, word or Dword.
IDE	I/O CMD Block BAR 0	This is an 8-byte mapping window with direct access to the command block registers. Access to address 0 should be word access. Access to all other fields must be byte accesses. Access to address 1 and the upper byte of the word access to address 0 are targeted to different internal registers. Software accesses may be byte, word or Dword.
	I/O Control Block BAR 1	This is a 4-byte mapping window with only 1 active address. Read and write access is targeted to different internal registers. Software accesses may be byte, word or Dword.
	I/O Bus Master BAR 4	This is a 16-byte mapping window with direct access. Access may be byte, word, or Dword. Software access may be byte, word or Dword.
Serial Port	I/O BAR 0	The same registers may be accessed as direct IO mapped offsets from the base address register 1.
	Memory BAR 1	The internal registers are accessed as direct memory mapped offsets from the base address register 0. The memory mapping window is 4 KB but only the low 8 addresses are valid. Software accesses may be byte, word or Dword.
IPMI[a]/ KCS	Memory BAR 0	The internal registers are accessed as direct memory mapped offsets from the base address register 0. The memory mapping window is 4 KB but only the low 2 addresses are valid. Read and write access target different registers. Software accesses may be byte, word or Dword.
	I/O BAR 1	The same registers can be accessed as direct I/O mapped offsets from the base address register 1. Software accesses may be byte, word or Dword.
IPMI[b] / BT	Memory BAR 0	The internal registers are accessed as direct memory mapped offsets from the base address register 0. The memory mapping window is 4 KB but only the low 3 addresses are valid. Read and write access target different registers. Software accesses may be byte, word or Dword.
	I/O BAR 1	The same registers can be accessed as direct I/O mapped offsets from the base address register 1. Software accesses may be byte, word or Dword.
UCHI[b]	Memory BAR 0	The internal registers are accessed as direct memory mapped offsets from the base address register 0. The memory mapping window is 4 KB but only the low 19 addresses are valid. Read and write access target different registers. Software accesses may be byte, word or Dword.
	I/O BAR 1	The same registers can be accessed as direct I/O mapped offsets from the base address register 1. Software accesses may be byte, word or Dword.

a. Not applicable to the **82573E/82573V/82573L**.
b. **631xESB/632xESB** only.

Expansion ROM Base Address.

This register is used to define the address and size information for boot-time access to the optional Flash memory. Only the LAN 0/LAN 1[1], IPMI/KCS, and IDE functions can use this window. It is enabled by EEPROM[2] words 24h, 14h and 1Ch for LAN 0, LAN 1 and IDE functions, respectively. This register returns a zero value for functions without expansion ROM window.

Bits 31:11	Bits 10:1	Bit 0
Expansion ROM BAR (R/W - 31:12316; '0' - 22/15:1)		En

Field	Bit(s)	RD/WR	Initial Value	Description
Address	31:11	R/W	0b	This field contains address bits, which are read/write and hardwired to 0b, depending on the memory mapping window size. The LAN Expansion ROM space can be 64 KB to 8 MB (4 MB for the **82573E/82573V/82573L**) in powers of 2. The mapping window size is set by EEPROM[a] word 0Fh.
Reserved	10:1	R	0b	This field is reserved and should be set to 0b. (Writes are ignored.)
Enable Expansion	0	R/W	0b	1b = Enables expansion ROM access. 0b = Disables expansion ROM access.

a. NVM word 4Ch for the **82573E/82573V/82573L**.

Subsystem ID. This value can be loaded automatically from the EEPROM/NVM at power up with a default value of 0000h.

PCI Function	Default Value	EEPROM Address
LAN Functions	0000h	0Bh
IDE	0000h	105h / 45h[a]
Serial Port	0000h	106h / 46h[a]
IPMI[b]/KCS	0000h	107h / 47h[a]
UHCI[c]	0000h	113h / 55h
BT[c]	0000h	115 / 55h

a. **82573E/82573V/82573L** only.
b. Not applicable to the **82573E/82573V/82573L**.
c. **631xESB/632xESB** only.

Subsystem Vendor ID.

This value can be loaded automatically from the EEPROM/NVM address 0Ch at power up or reset. A value of 8086h is the default for this field at power up if the EEPROM/NVM does not respond or is not programmed. All functions are initialized to the same value.

1. Not applicable to the **82573E/82573V/82573L**.
2. NVM word 4Ch for the **82573E/82573V/82573L**.

Cap_Ptr. The Capabilities Pointer field (Cap_Ptr) is an 8-bit field that provides an offset in the device PCI Configuration Space for the location of the first item in the Capabilities Linked List. The Ethernet controller sets this bit and implements a capabilities list to indicate PCI Power Management support. In addition, Message Signaled Interrupts and PCIe* extended capabilities are implemented. Its value is C8h, which is the address of the first entry, PCI Power Management.

Address	Item	Next Pointer
C8h : CFh	PCI Power Management	D0h
D0h : DFh	Message Signaled Interrupt	E0h
E0h : F3h	PCIe* Capabilities	00h

Interrupt Pin. This is a read only register.

LAN 0 / LAN 1.[1] A value of 01h or 02h indicates that this function implements legacy interrupt on INTA or INTB, respectively. This value is loaded from EEPROM/NVM word 24h[2] and 14h for LAN 0 and LAN 1, respectively.

IDE Function. A value of 01h, 02h, 03h or 04h indicates that this function implements legacy interrupt on INTA, INTB, INTC or INTD, respectively. This value is loaded from EEPROM word 21h.

Serial Port Function. A value of 1, 2, 3 or 4 indicates that this function implements legacy interrupt on INTA, INTB, INTC or INTD, respectively. It is loaded from EEPROM word 21h.

IPMI/KCS Function. A value of 1, 2, 3 or 4 indicates that this function implements legacy interrupt on INTA, INTB, INTC or INTD, respectively. It is loaded from EEPROM word 21h.

UHCI Function. A value of 1, 2, 3 or 4 indicates that this function implements legacy interrupt on INTA, INTB, INTC or INTD, respectively. It is loaded from EEPROM word 6Bh/12Bh.

BT Function. A value of 1, 2, 3 or 4 indicates that this function implements legacy interrupt on INTA, INTB, INTC or INTD, respectively. It is loaded from EEPROM word 6Bh/12Bh.

For the **82573E/82573V/82573L**, IDE, Serial Port, and KCS Functions: A value of 01h, 02h, 03h or 04h indicates that this function implements legacy interrupt on INTA, INTB, INTC or INTD, respectively. This value is loaded from the *INT Sel* fields in the NVM (word 49h).

Interrupt Line. Read and write registers programmed by software indicate the type of system interrupt request lines the device interrupt pin is bound to. (Each PCIe* function has its own register.)

Max_Lat/Min_Gnt.

This field is not used and is hardwired to 0b.

1. If only a single device or function of Ethernet controller component is enabled, this value is ignored, and the Interrupt Pin field of the enabled device reports INTA# usage.
2. **82573E/82573V/82573L** only.

4.7.5 UHCI Registers (631xESB/632xESB only)

The following two registers should be implemented only in UHCI function (5) configuration space. Table 4-17 lists the organization of the UHCI specific registers.

Table 4-17. UHCI Registers

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
60h	Reserved			*SBRN*
. . .	. . .			
C0h	Reserved		USB LEGSUP	

NOTE: The following color notation is used for reference:

	Fields identical to all functions.
	Read only fields.
	Hard coded fields.

Field	Bit	Attr	Default	Description
Reserved	31:8	RO	0b	Reserved.
SBRN	7:0	RO	10h	Serial Bus Release Number Indicates compliance with USB 1.0 specification.

The Command Register indicates the command to be executed by the serial bus host controller.

Field	Bit	Attr	Default	Description
Reserved	15:14	RO	0b	Reserved.
USBPIRQEN	13	R/W	1b	PCI Interrupt Enable This bit is used to prevent the USB controller from generating an interrupt due to transactions on its ports. Note that when disabled, it probably is configured to generate an SMI using bit 4 of this register. The default to 1b for compatibility with older USB software. Mapped to UHCI_INT[5]. 0b = Disable. 1b = Enable.
SMIBYUSB	12	RO	0b	SMI Caused by USB Interrupt This bit indicates if an interrupt event occurred from this Ethernet controller. The interrupt from the Ethernet controller is taken before the enable in bit 13 has any effect to create this read-only bit. Note that even if the corresponding enable bit is not set in bit 4, this bit may still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Software should clear the interrupts via the USB controllers. Writing a 1b to this bit has no effect. Mapped to UHCI_INT[6]. 1b = Event occurred. 0b = No event.
Reserved	11:5	RO	0b	Reserved.
USBSMIEN	4	R/W	0b	SMI on USB IRQ Enable Mapped to UHCI_INT[4] 0b = Disable 1b = Enable. USB interrupt causes an SMI event.
Reserved	3:0	RO	0b	Reserved.

4.7.6 PCI Power Management Registers

All fields are reset on full power up. All of the fields except PME_En and PME_Status are reset on exit from the D3cold state. If auxiliary power is not supplied, the PME_En and PME_Status fields are also reset on exit from the D3cold state.

The tables that follow list the organization of the PCI Power Management Register block. Initial values are marked in parenthesis and the following color notation is used.

Some fields in this section depend on the power management enable bits in EEPROM/NVM word 0Ah.

Table 4-18. Power Management Register Block

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
C8h	Power Management Capabilities (PMC)		Next Pointer (0xD0)	Capability ID
CCh	Data	PMCSR_BSE Bridge Support Extensions	Power Management Control / Status Register (PMCSR)	

NOTE: The following color notation is used for reference:

	Fields identical to all functions.
	Read only fields.
	Hard coded fields and strapping options.

The following section describes the register definitions, whether they are required or optional for compliance, and how they are implemented in Ethernet controller.

Capability ID. The Capability ID is 1 byte at offset C8h and is read only. This field equals 01h, indicating the linked list item as the PCI Power Management Registers.

Next Pointer. The Next Pointer is 1 byte at offset C9h and is read only. This field provides an offset to the next capability item in the capability list. It has a value of D0h, which points to the MSI capability.

Power Management Capabilities (PMC).

The PMC is 2 bytes at offset CAh and is read only. This field describes the device functionality at the power management states as described in the table that follows. Each device function has its own register.

Table 4-19. Power Management Capabilities (PMC)

Bit(s)	Default	RD/WR	Description
15:11		RO	PME_Support. This 5-bit field indicates the power states in which the function may assert PME#. The IDE function field is hardwired to 0b while the other functions depend on EEPROM word 0Ah: Condition ⇒ Functionality ⇒ Value PM Disable in EEPROM ⇒ No PME at all states ⇒ 00000b PM Enable & No Aux Pwr ⇒ PME at D0 and D3hot ⇒ 01001b PM Enable with Aux Pwr ⇒ PME at D0, D3hot and D3cold ⇒ 11001b
10	0b	RO	D2_Support. The Ethernet controller does not support the D2 state.
9	0b	RO	D1_Support. The Ethernet controller does not support D1 state.
8:6	000b	RO	Auxiliary Current. This is the required current defined in the data register.
5	1b	RO	DSI. The Ethernet controller requires its device driver to be executed following transition to the D0 uninitialized state.
4	0b	RO	Reserved. This bit is reserved and should be set to 0b.
3	0b	RO	PME_Clock. The PME clock is disabled and is hardwired to 0b.
2:0	010b	RO	Version. The Ethernet controller complies with PCI Power Management Specification, Revision 1.1.

Power Management Control/Status Register (PMCSR).

The PMCSR is 2 bytes at offset CCh and is read/write. This register is used to control and monitor power management events in the device. Each device function has its own PMCSR.

Table 4-20. Power Management Control/Status Register

Bit(s)	Default	RD/WR	Description
15	0b (at power up)	R/W1C	PME_Status. This bit is set to 1b when the function detects a wake-up event independent of the state of the PME enable bit. Writing a 1b clears this bit.
14:13	Reflects value in Data Register	RO	Data_Scale. This field indicates the scaling factor that is used to interpret the value of the Data Register. For the LAN and Common functions this field equals 01b (indicating 0.1 watt units) if power management is enabled in the EEPROM/NVM and the Data_Select field is set to 0, 3, 4, or 7 (or 8 for Function 0). Otherwise, it equals 00b. For the manageability functions this field equals 10b (indicating 0.01 watt units) if power management is enabled in the EEPROM/NVM and the Data_Select field is set to 0, 3, 4, or 7. Otherwise, it equals 00b.
12:9	0000b	R/W	Data_Select. This four-bit field is used to select which data is to be reported through the Data Register and Data_Scale field. These bits are writable only when power management is enabled through EEPROM/NVM.
8	0b (at power up)	R/W	PME_En. If Power Management is enabled in the EEPROM/NVM, writing a 1b to this register enables wakeup. If power management is disabled in the EEPROM/NVM, writing a 1b to this bit has no affect and will not set the bit to 1b.
7:2	000000b	RO	Reserved. This bit is reserved and should return a value of 0b for this field.
1:0	00b	R/W	PowerState. This field is used to set and report the power state of a function as defined below: 00b – D0 01b – D1 (cycle ignored if written with this value) 10b – D2 (cycle ignored if written with this value) 11b – D3 (cycle ignored if power management is disabled in the EEPROM/NVM)

PMCSR_BSE Bridge Support Extensions: 1 Byte, Offset CEh, (RO).

This field is 1 byte at offset CEh and is read only. This register is not implemented in Ethernet controller and its value should be set to 0b.

Data Register: 1 Byte, Offset CFh, (RO).

The Data Register is 1 byte at offset CFh and is read only. This optional register is used to report power consumption and heat dissipation. The reported register is controlled by the Data_Select field in the PMCSR, and the power scale is reported in the Data_Scale field of the PMCSR. Data from this field is loaded from the EEPROM/NVM if power management is enabled in the EEPROM/NVM or with a default value of 00h otherwise. The values for the Ethernet controller functions are as follows:

Function	D0 (Consume/Dissipate)	D3 (Consume/Dissipate)	Common	Data Scale
Data Select	0h / 4h	3h / 7h	8h	
Function 0	EEPROM Word 22h	EEPROM Word 22h	EEPROM Word 22h	01b
Function 1[a]	EEPROM Word 22h	EEPROM Word 22h	00h (EEPROM Word 22h for the **631xESB/ 632xESB**)	01b
IDE	EEPROM Word 40h[b]/100h	EEPROM Word 41h[b]/101h	00h (EEPROM Word 22h for the **631xESB/ 632xESB**)	10b
Serial Port	EEPROM Word 40h[b]/100h	EEPROM Word 41h[b]/101h	00h (EEPROM Word 22h for the **631xESB/ 632xESB**)	10b
IPMI[a] / KCS[c]	EEPROM Word 40h[b]/100h	EEPROM Word 41h[b]/101h	00h (EEPROM Word 22h for the **631xESB/ 632xESB**)	10b
UHCI[d]	EEPROM Word 12Ch/6Ch	EEPROM Word 12Ch/6Ch	00h (EEPROM Word 22h for the **631xESB/ 632xESB**)	10b
IPMI[a] / BT[d]	EEPROM Word 12Ch/6Ch	EEPROM Word 12Ch/6Ch	00h (EEPROM Word 22h for the **631xESB/ 632xESB**)	10b

a. Not applicable to the **82573E/82573V/82573L**.
b. **82573E/82573V/82573L** only.
c. **82573E/82573V** only.
d. **631xESB/632xESB** only.

NOTE: For other Data_Select values, the Data Register output is reserved (0b).

4.7.6.1 Message Signaled Interrupt Configuration Registers

This structure is required for PCIe* devices. There are no changes to this structure from the PCI Specification, Revision 2.2. Initial values of the configuration registers are marked in parenthesis and the following color notation is used.

Table 4-21. Message Signaled Interrupt Configuration Registers

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
D0h	Message Control (0080h)		Next Pointer (E0h)	Capability ID (05h)
D4h	Message Address			
D8h	Message Upper Address			
DCh	Reserved		Message Data	

NOTE: The following color notation is used for reference:

	Fields identical to all functions.
	Read only fields.
	Hard coded fields and strapping options.

Capability ID: 1 Byte, Offset D0h, (RO).

This field equals 05h indicating the linked list item as being the Message Signaled Interrupt registers.

Next Pointer: 1 Byte, Offset D1h, (RO).

This field provides an offset to the next capability item in the capability list. Its value of E0h points to the PCIe* capability.

Message Control: 2 Byte, Offset D2h, (R/W).

The register fields are described in the table that follows. There is a dedicated register per PCI function to enable separately their MSI.

Bits 15:8	Bit 7	Bits 6:4	Bits 3:1	Bit 0
Reserved	64b	Multiple Enable	Multiple Capable	MSI Enable

Bits	Default	RD/WR	Description
15:8	0b	RO	Reserved. Reads as 0b.
7	1b	RO	64-bit Capable. A value of 1b indicates that the Ethernet controller is capable of generating 64-bit message addresses.
6:4	000b	RO	Multiple Message Enable. Ethernet controller return 000b to indicate that it supports a single message per function.
3:1	000b	RO	Multiple Message Capable. Ethernet controller indicates a single requested message per each function.
0	0b	R/W	MSI Enable. If 1b, Message Signaled Interrupts. In this case, the Ethernet controller generates MSI for interrupt assertion instead of INTx signaling.

Message Address Low 4 Byte, Offset D4h, (R/W).

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 always return 0b regardless of the write operation.

Message Address High 4 Byte, Offset D8h, (R/W).

Written by the system to indicate the upper 32-bits of the address to use for the MSI memory write transaction.

Message Data 2 Byte, Offset 0xDC, (R/W).

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.

4.7.6.2 PCIe* Configuration Registers

PCIe* provides two mechanisms to support native features:

- PCIe* defines a PCI capability pointer indicating support for PCIe*
- PCIe* extends the configuration space beyond the 256 bytes available for PCI to 4096 bytes.

The Ethernet controller implements the PCIe* Capability Structure for Endpoint Devices as follows:

Table 4-22. PCIe* Configuration Registers

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
E0h	PCIe* Capability Register		Next Pointer	Capability ID
E4h	Device Capability			
E8h	Device Status		Device Control	
ECh	Link Capability			
F0h	Link Status		Link Control	

NOTE: The following color notation is used for reference:

	Fields identical to all functions.
	Read only fields.
	Hard coded fields and strapping options.

Capability ID. The Capability ID is 1 byte at offset E0h and is read only. This field equals 10h, indicating the linked list item is a PCIe* Capabilities Register.

Next Pointer. The Next Pointer is 1 byte at offset E1h and is read only. It points to the offset of the next capability item in the capability list. A 00h value indicates that it is the last item in the capability linked list.

PCIe* CAP. The PCIe* CAP field is 2 bytes at offset E2h and is read only. The PCIe* capabilities register identifies the PCIe* device type and associated capabilities. This is a read only register identical to all functions.

Bit(s)	Default	RD/WR	Description
15:14	00b	RO	Reserved. This field is reserved and should be set to 0b.
13:9	00000b	RO	Interrupt Message Number. The Ethernet controller does not implement multiple MSI per function. Thus, this field is hardwired to 0b.
8	0b	RO	Slot Implemented. The Ethernet controller does not implement slot options. Therefore, this field is hardwired to 0b.
7:4	0000b/ 0001b	RO	Device/Port Type. This field indicates the type of PCIe* functions. All functions except the IDE are Native PCI functions with a value of 0000b. The value of the IDE function is loaded from the EEPROM/NVM word 19h bit 13 and can be either 0000b or 0001b.
3:0	0001b	RO	Capability Version. This indicates the PCIe* capability structure version number 1.

Device CAP. This field is 4 bytes at offset E4h and is read only. It identifies the PCIe* device specific capabilities. It is a read only register with the same value for LAN function 0 and LAN function 1[1].

Bit(s)	Default	RD/WR	Description
31:28	RO	0000b	Reserved. This field is reserved and should be set to 0b.
27:26	RO	00b	Slot Power Limit Scale. This field is used in upstream ports only. It is hardwired in the Ethernet controller to 0b for all functions.
25:18	RO	00h	Slot Power Limit Value. This field is used in upstream ports only. It is hardwired in the Ethernet controller to 0b for all functions.
17:15	RO	000b	Reserved. This field is reserved and should be set to 0b.
14	RO	0b	Power Indicator Present. In the Ethernet controller, this bit is hardwired 0b for all functions.
13	RO	0b	Attention Indicator Present. In the Ethernet controller, this bit is hardwired 0b for all functions.
12	RO	0b	Attention Button Present. In the Ethernet controller, this bit is hardwired 0b for all functions.
11:9	RO	110b	Endpoint L1 Acceptable Latency. This field indicates the acceptable latency that the Ethernet controller can withstand due to the transition from the L1 state to the L0 state. All functions share the same value loaded from the EEPROM/NVM PCIe* Initialization Configuration 1, word 18h.
8:6	RO	011b	Endpoint L0s Acceptable Latency. This field indicates the acceptable latency that the Ethernet controller can withstand due to the transition from the L0s state to the L0 state. All functions share the same value loaded from the EEPROM/NVM PCIe* Initialization Configuration 1, word 18h.

1. LAN function 1 not applicable to the **82573E/82573V/82573L**.

Bit(s)	Default	RD/WR	Description
5	RO	0b	Extended Tag Field Supported. This field identifies the maximum Tag field size supported. The Ethernet controller supports a 5-bit Tag field for all functions.
4:3	RO	00b	Phantom Function Supported. This is not supported by the Ethernet controller.
2:0	RO	001b	Max Payload Size Supported. This field indicates the maximum payload that the device can support for TLPs. It is loaded from the EEPROM/NVM PCIe* Initialization Configuration 3, word 1Ah bit 8, with a default value of 001b (256 bytes).

Device Control. The Device Control field is 2 bytes at offset E8h and is read/write. This register controls the PCIe* specific parameters. There is a dedicated register for each function.

Bit(s)	RD/WR	Default	Description
15	RO	0b	Reserved. This is reserved and should be set to 0b.
14:12	RW	010b / 000b	Max Read Request Size. This field sets maximum read request size for the device as a requester. 000b = 128 bytes. This is the default value for non-LAN functions. 001b = 256 bytes. 010b = 512 bytes. This is the default value for the LAN devices. 011b = 1 KB. 100b = 2 KB. 101b = Reserved. 110b = Reserved. 111b = Reserved.
11	RW	1b	Enable No Snoop. Snoop is gated by NONSNOOP bits in the GCR register in the CSR space.
10	RW	0b	Auxiliary Power PM Enable. When this is set, the device can draw auxiliary power independent of the PME AUX power signal. The **82571EB/82572EI** is a multi-function device and is allowed to draw auxiliary power if at least one of the functions has this bit set. **Note:** The **631xESB/632xESB** is always connected to the AUX plane of ESB2. This bit does not affect the **631xESB/632xESB** behavior.
9	RW	0b	Phantom Functions Enable. This field is not implemented in the Ethernet controller.
8	RW	0b	Extended Tag Field Enable. This field is not implemented in the Ethernet controller.
7:5	RW	000b (128 Bytes)	Max Payload Size. This field sets maximum TLP payload size for the device functions. As a receiver, the device must handle TLPs as large as the set value. As transmitter, the device must not generate TLPs exceeding the set value. The Max Payload Size supported in the device capabilities register indicates permissible values that can be programmed.
4	RW	1b	Enable Relaxed Ordering. If this bit is set, the device is permitted to set the Relaxed Ordering bit in the attribute field of write transactions that do not need strong ordering. (Documentation in the RO_DIS bit of the CTRL_EXT register also provides more details.)
3	RW	0b	Unsupported Request Reporting Enable. This bit enables error report.

Bit(s)	RD/WR	Default	Description
2	RW	0b	Fatal Error Reporting Enable. This bit enables error report.
1	RW	0b	Non-Fatal Error Reporting Enable. This bit enables error report.
0	RW	0b	Correctable Error Reporting Enable. This bit enables error report.

Device Status. The Device Status field is 2 bytes at offset EAh and is read only. This Register provides information about PCIe* device specific parameters. There is a dedicated register per each function.

Bit(s)	RD/WR	Default	Description
15:6	RO	00h	Reserved. This is reserved and should be set to 0b.
5	RO	0b	Transaction Pending. This indicates whether or not the Ethernet controller has any pending transactions. For the **82571EB/82572EI**, transactions include completions for any outstanding non-posted request for all used traffic classes.
4	RO	0b	Aux Power Detected. If auxiliary power is detected this field is set to 1b. It is a strapping signal from the periphery identical for all functions. This is reset on LAN Power Good and GIO Power Good only. For the **631xESB/632xESB**, This bit is hardwired to 1b.
3	RW1C	0b	Unsupported Request Detected. This indicates that the Ethernet controller received an unsupported request. This field is identical in all functions. The Ethernet controller cannot distinguish which function caused the error.
2	RW1C	0b	Fatal Error Detected. This indicates status of fatal error detection.
1	RW1C	0b	Non-Fatal Error Detected. This indicates status of non-fatal error detection.
0	RW1C	0b	Correctable Detected. This indicates status of correctable error detection.

Link CAP. The Link CAP is 4 bytes at offset ECh and is read only. This register identifies the PCIe* Link specific capabilities. This is a read only register identical to all functions.

Bit(s)	RD/WR	Default	Description
31:24	HwInit	0b	Port Number. This represents the PCIe* port number for the given PCIe* Link. The field is set in the link training phase.
23:18	RO	000000b	Reserved. This is reserved and should be set to 0b.
17:15	RO	110b (32 – 64 µs)	L1 Exit Latency. This indicates the exit latency from L1 to L0 state. This field is loaded from the EEPROM PCIe* Initialization Configuration 1, word 18h. Defined encoding: 000b = Less than 1 µs 001b = 1 µs - 2 µs 010b = 2 µs - 4 µs 011b = 4 µs - 8 µs 100b = 8 µs - 16 µs 101b = 16 µs - 32 µs 110b = 32 µs - 64 µs 111b = L1 transition not supported

Bit(s)	RD/WR	Default	Description
14:12	RO	001b (64 – 128 ns)	L0s Exit Latency. This indicates the exit latency from L0s to L0 state. This field is loaded from the EEPROM PCIe* Initialization Configuration 1, word 18h. There are two values for Common PCIe* clock or Separate PCIe* clock. Defined encoding: 000b = Less than 64 ns 001b = 64 ns – 128 ns 010b = 128 ns – 256 ns 011b = 256 ns – 512 ns 100b = 512 ns – 1 µs 101b = 1 µs – 2 µs 110b = 2 µs – 4 µs 111b = Reserved If the Ethernet controller uses common clock, PCIe* Initialization Configuration 1, equals 1B0h/70h, bits [2:0]; and if the Ethernet controller uses separate clock, 1B0h/70h, bits [5:3].
11:10	RO	11b 00b[a]	Active State Link PM Support. This indicates the level of active state power management supported in the Ethernet controller. Defined encoding: 00b = No L0s or L1 Supported (default for the **631xESB/632xESB**) 00b = Reserved for all remaining Ethernet controllers 01b = L0s Entry Supported 10b = Reserved 11b = L0s and L1 Supported This field is loaded from the EEPROM/NVM PCIe* Initialization Configuration 3, word 1Ah.
9:4	RO	4h	Max Link Width. This indicates the maximum link width. The Ethernet controller can support by 1-, by 2- and by 4-link width. The field is loaded from the EEPROM/NVM PCIe* Initialization Configuration 3, word 1Ah, with a default value of 4 lanes for the Ethernet controller. Defined encoding: 000000b = Reserved 000001b = x1 (Reserved for the **631xESB/632xESB**) 000010b = x2 (Reserved for the **82573E/82573V/82573L** and the **631xESB/632xESB**) 000100b = x4 (Reserved for the **82573E/82573V/82573L**)
3:0	RO	0001b	Max Link Speed. The Ethernet controller indicates a maximum link speed of 2.5 Gb/s.

a. **631xESB/632xESB** only.

Link Control. The Link Control field is 2 bytes at offset F0h and is read only. This register controls PCIe* link specific parameters. There is a dedicated register for each function.

Bit(s)	RD/WR	Default	Description
15:8	RO	0b	Reserved. This is reserved and should be set to 0b.
7	RW	0b	Extended Synchronization. This bit forces extended transmit of the FTS ordered set in FTS and extra TS1 at exit from L0s prior to entering L0.
6	RW	0b	Common Clock Configuration. When this is set, it indicates that the Ethernet controller and the component at the other end of the link are operating with a common reference clock. A value of 0b indicates that they are operating with an asynchronous clock. This parameter affects the L0s exit latencies.
5	RO	0b	Retrain Clock. This is not applicable for endpoint devices and is hardwired to 0b.
4	RO	0b	Link Disable. This field is not applicable for endpoint devices and is hardwired to 0b.
3	RW	0b	Read Completion Boundary.
2	RO	0b	Reserved. This is reserved and should be set to 0b.
1:0	RW	00b	Active State Link PM Control. This field controls the active state power management supported on the link. Link PM functionality is determined by the lowest common denominator of all functions. Defined encoding: 00b = PM Disabled 01b = L0s Entry Supported 10b = Reserved 11b = L0s and L1 Supported

Link Status. The Link Status field is 2 bytes at offset F2h and is read only. This register provides information about PCIe* link specific parameters. This is a read only register identical to all functions.

Bit(s)	RD/WR	Default	Description
15:13	RO	0000b	Reserved. This is reserved and should be set to 0b.
12	HwInit	1b	Slot Clock Configuration. When this is set, it indicates that the Ethernet controller uses the physical reference clock that the platform provides on the connector. This bit must be cleared if the Ethernet controller uses an independent clock. The Slot Clock Configuration bit is loaded from the Slot_Clock_Cfg EEPROM/NVM bit. For the **631xESB/632xESB**, this bit must be set to 1b.
11	RO	0b	Link Training. This indicates that link training is in progress.
10	RO	0b	Link Training Error. This indicates that a link training error has occurred. For the **631xESB/632xESB**, this bit is hardwired to 0b.
9:4	RO	000001b	Negotiated Link Width. This field indicates the negotiated width of the link. Relevant encoding: 000001b = x1 000010b = x2 (**82571EB/82572EI** only) 000100b = x4
3:0	RO	0001b	Link Speed. This field indicates the negotiated link speed. A value of 0001b is the only defined speed, which is 2.5 Gbps.

PCIe* Extended Configuration Space

PCIe* Configuration Space is located in a flat memory mapped address space. PCIe* extends the configuration space beyond the 256 bytes available for PCI to 4096 bytes. The Ethernet controller decodes additional 4-bits (bits 27:24) to provide the additional configuration space as shown in the figure below. PCIe* reserves the remaining 4 bits (bits 31:28) for future expansion of the configuration space beyond 4096 bytes.

The configuration address for a PCIe* device is computed using PCI-compatible bus, device and function numbers as follows:

Bits 31:28	Bits 27:20	Bits 19:15	Bits 14:12	Bits 11:2	Bits 1:0
0000b	Bus #	Device #	Func #	Register Address (offset)	00b

PCIe* Extended Configuration Space is allocated using a linked list of optional or required PCIe* extended capabilities following a format resembling PCI capability structures. The first PCIe* extended capability is located at offset 100h in the device configuration space. The first Dword of the capability structure identifies the capability and version and points to the next capability.

The Ethernet controller supports the Advanced Error Reporting Capability at offset 100h of the PCIe* extended capabilities.

Advanced Error Reporting Capability

The PCIe* advanced error reporting capability is an optional extended capability to support advanced error reporting. This is supported by the Ethernet controller. Details and definitions of the extended capabilities structures and advanced error reporting capabilities are documented in the PCIe* Specification.

Device Serial Number

The Ethernet controller implements the PCIe* Device Serial Number optional capability. The Device Serial Number is a read only 64-bit value that is unique for a given PCIe* device.

All multi-function devices that implement this capability must implement it for function 0. Other functions that implement this capability must return the same Device Serial Number value as that reported by function 0.

Table 4-23. PCIe* Device Serial Number Capability Structure

Offset 00h	PCIe* Enhanced Capability Header
Offset 04h	Serial Number Register (Lower Dword)
Offset 08h	Serial Number Register (Upper Dword)

Device Serial Number Enhanced Capability Header (Offset 00h).

The following tables detail allocation of register fields as well as their respective bit definitions in the Device Serial Number enhanced capability header. The Extended Capability ID for the Device Serial Number capability is 0003h.

Table 4-24. Device Serial Number Enhanced Capability Header

Bits 31:20	Next Capability Offset
Bits 19:16	Capability Version
Bits 15:0	PCIe* Extended Capability ID

Table 4-25. Device Serial Number Enhanced Capability Header

Bit(s)	Attributes	Description
31:20	RO	Next Capability Offset. This field contains the offset to the next PCIe* capability structure or 000h if no other items exist in the linked list of capabilities. For extended capabilities implemented in device configuration space, this offset is relative to the beginning of the PCI compatible configuration space and must always equal either 000h (to terminate the list of capabilities) or be greater than 0FFh.
19:16	RO	Capability Version. This field is a PCI SIG defined version number that indicates the version of the capability structure present. It must equal 1h for this version of the specification.
15:0	RO	PCIe* Extended Capability ID. This field is a PCI SIG defined identification number indicating indicates the nature and format of the extended capability. The Extended Capability ID for the Device Serial Number Capability is 0003h.

Serial Number Register (Offset 04h).

The Serial Number register is a 64-bit field that contains the IEEE defined 64-bit extended unique identifier (EUI-64*). The following tables detail the allocation of register fields as well as their respective bit definitions in the Serial Number register.

Table 4-26. Device Serial Number Enhanced Capability Header

Bits 63:32	Serial Number Register (Upper Dword)
Bits 31:0	Serial Number Register (Lower Dword)

Table 4-27. Serial Number Register

Bit(s)	Attributes	Description
63:0	RO	PCIe* Device Serial Number. This field contains the IEEE defined 64-bit extended unique identifier (EUI-64*). This identifier includes a 24-bit company identification value assigned by IEEE registration authority and a 40-bit extension identifier assigned by the manufacturer.

Serial Number Definition in Ethernet controller.

In the Ethernet controller, the serial number uses the MAC address according to the following definition.

Field	Company ID			Extension identifier				
Order	Addr+0	Addr+1	Addr+2	Addr+3	Addr+4	Addr+5	Addr+6	Addr+7
Most Significant Byte						Least Significant Byte		
Most Significant Bit						Least Significant Bit		

The serial number can be constructed from the 48-bit MAC address in the following form:

Field	Company ID			MAC Label		Extension identifier		
Order	Addr+0	Addr+1	Addr+2	Addr+3	Addr+4	Addr+5	Addr+6	Addr+7
Most Significant Bytes						Least Significant Byte		
Most Significant Bit						Least Significant Bit		

The MAC label in this case is FFFFh. For example, the vendor is Intel and the vendor ID is 00-A0-C9, and the extension identifier, 23-45-67. In this case, the 64-bit serial number is:

Field	Company ID			MAC Label		Extension identifier		
Order	Addr+0	Addr+1	Addr+2	Addr+3	Addr+4	Addr+5	Addr+6	Addr+7
	00	A0	C9	FF	FF	23	45	67
Most Significant Byte						Least Significant Byte		
Most Significant Bit						Least Significant Bit		

The MAC address is the function 0 MAC address as loaded from EEPROM into the RAL and RAH registers.

The official document defining EUI-64 can be located at: http://standards.ieee.org/regauth/oui/tutorials/EUI64.html.

EEPROM/NVM Interface 5

5.1 Introduction

The **82571EB/82572EI** and the **631xESB/632xESB** use an EEPROM device to store product configuration information. The EEPROM is divided into three general regions:

- **Hardware Accessed —** Loaded by the Ethernet controller after power-up, PCIe* reset de-assertion, in-band PCIe* reset, a D3 to D0 transition, or a software commanded EEPROM read (CTRL_EXT.EE_RST), and a software reset (CTRL.RST, word 00000h, bit 26).
- **Software Accessed —** Used by software only. These registers are listed in this document for convenience and are only for software and are ignored by the Ethernet controller.
- **Firmware Accessed —** Used by BMC, PT, SPT, or ASF firmware (For more information about manageability, contact your Intel Field Service Representative.
  - Firmware Reset (all modes) - occurs after any of the following:

    LAN power up (LAN_PWR_GOOD assertion)

    Software-initiated firmware reset through a host command

    Software-initiated firmware reset through MAC CSRs (writing 40h and then 80h to HICR)

    Certain unrecoverable errors during manageability operation (RAM parity error, hardware watchdog timer expiration, etc.)
  - Software-Initiated EEPROM Reload (ASF mode only) - occurs after any of the following:

    Software-initiated assertion of the SWSM.WMNG bit while manageability clock is off

    Software-initiated EEPROM reload through ASF register (asserting FR_RST.FRC_EELD or FRC_FLUSH)

    Software-initiated ERPROM reload through a host command

    System state transition S0 to S5 while ASF register bit CTL_PWRLS is cleared

Note: The **82573E/82573V/82573L** uses an NVM device for configuration, log events, and firmware extensions. See Section 5.5 for details.

5.2 EEPROM Device (82571EB/82572EI/631xESB/632xESB)

The EEPROM interface supports Serial Peripheral Interface (SPI) mode 0 and expects the EEPROM to be capable of 2 MHz operation.

The Ethernet controller is compatible with many sizes of 4-wire serial EEPROM devices. If flexibility mode functionality is desired (ASF, PT, SPT, or full BMC), up to a 256 Kb serial SPI can be used.

Note: Minimum EEPROM sizes for the **631xESB/632xESB**: 32 Kb (no manageability image); 128 Kb (any manageability image).

Note: Minimum EEPROM sizes for the **82571EB/82572EI**: 16 Kb (no manageability image); 128 Kb (any manageability image). Recommended EEPROM sizes are: 32 Kb (no manageability image); 256 Kb (any manageability image).

The Ethernet controller automatically determines the address size to be used with the SPI EEPROM it is connected to and sets the *EEPROM Size* field of the EEPROM/Flash Control and Data Register (EEC.EE_ADDR_SIZE; bit 10). Software uses this size to determine the EEPROM access method. The exact size of the EEPROM is stored within one of the EEPROM words.

Note: The different EEPROM sizes have two different numbers of address bits (8 bits or 16 bits). Thus they must be accessed with a slightly different serial protocol. Software must be aware of this if it accesses the EEPROM using direct access.

5.2.1 Software Accesses

The Ethernet controller provides two different methods for software access to the EEPROM. It 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 into the *Read Address* field (EERD.ADDR; bits 15:2) and simultaneously writes a 1b to the *Start Read* bit (EERD.START; bit 0). The Ethernet controller then reads the word from the EEPROM, sets the *Read Done* bit (EERD.DONE; bit 1), and puts the data in the *Read Data* field (EERD.DATA; bits 31:16). Software can poll the EEPROM Read register until it sees the *Read Done* bit set, then use the data from the *Read Data* field. Any words read this way are not written to the Ethernet controller'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; bit 6).
2. Read the *EEPROM Grant* bit (EEC.EE_GNT; bit 7) 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 datasheet.
4. Write a 0b to the *EEPROM Request* bit (EEC.EE_REQ; bit 6).

Finally, software can cause the Ethernet controller to re-read part of the hardware accessed fields of the EEPROM (setting the Ethernet controller's internal registers appropriately) by writing a 1b to the *EEPROM Reset* bit of the *Extended Device Control Register* (CTRL_EXT.EE_RST; bit 13).

5.2.2 Signature and CRC Fields

The only way the Ethernet controller can discover whether an EEPROM is present is by trying to read the EEPROM. The Ethernet controller first reads the EEPROM *Sizing & Protected* field Word at address 12h. The Ethernet controller checks the signature value for bits 15 and 14. If bit 15 is 0b and bit 14 is 1b, it considers the EEPROM to be present and valid and reads additional EEPROM words and programs its internal registers based on the values read. Otherwise, it ignores the values it read from that location and does not read any other words.

In ASF Mode, the **82571EB/82572EI'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 values. This CRC does not affect the remaining **82571EB/82572EI** configuration, including the Management Control register. For details on how the CRC is calculated see Section 5.6.7.

Note: When the signature is not correct (the EEPROM is blank), it cannot be accessed by parallel access if its size is 512 bytes or smaller. As a result, the EEPROM should be accessed using the bit-bang mechanism.

5.2.3 Protected EEPROM Space (82571EB/82572EI)

The 82571EB/82572EI provides a mechanism for a hidden area in the EEPROM to the host. The hidden area cannot be accessed via the EEPROM registers in the CSR space. It can be accessed only by the Manageability (MNG) subsystem. For more information about the management subsystem, contact your Intel Field Service Representative.

5.2.4 Protected EEPROM Space (631xESB/632xESB)

The **631xESB/632xESB** provides a mechanism to enable different levels of protection for different areas in the EEPROM. All the protection mechanisms only affect host accesses. There are no restrictions on firmware accesses to the EEPROM. The EEPROM can be divided into three areas from a host point of view:

1. Read only area: This area usually contains the basic setup of the hardware and firmware.

Note: The PXE area (30h:3Fh) can be written by software even if it's part of the read only area.

2. Read/Write area: This area usually contains the host related structures.
3. Hidden area: This area usually contains secret data such as encryption keys or authentication signatures.

The protected areas cannot be accessed via the EEPROM registers in the CSR space. It can be accessed only by the MNG subsystem. For more information on the MNG subsystem, contact your Intel Field Service Representative.

The protection mechanisms are activated using the following fields in the EEPROM:

- Word 12h[3:0] defines the size of the hidden area. This area is located at the high addresses of the EEPROM. The size is fixed according to Table 5-1.
- Word 12h[4] enables all the protection mechanisms. As long as this bit is zero, all the areas are accessible to the host to read and write. It is highly recommended to set this bit *only* after the EEPROM image is stable and tested. Once this bit is set, word 12h becomes read-only to the software.
- Word 12h[15:14] indicates the presence of a programmed EEPROM. If these bits are not equal to 01b, it is assumed the EEPROM is not present or empty. All protection mechanisms are disabled in this case.
- Word 2Ch[14:0] – indicates the end of the read-only area. The read-only area starts at address 00h. This word points to the address of the last word in the read only area. If this word is zero, then this protection mechanism is disabled.

Table 5-1. Size of EEPROM Hidden Block

Word 12h[3:0]	Hidden block size
0000	No hidden block
0001	2 bytes
0010	4 bytes
0011	8 bytes
0100	16 bytes
0101	32 bytes
0110	64 bytes
0111	128 bytes
1000	256 bytes
1001	512 bytes
1010	1 KB
1011	2 KB
1100	4 KB
1101	8 KB
1110	16 KB
1111	32 KB

Note: For the **631xESB/632xESB**, when software issues a burst to the EEPROM using the bit-bang mechanism and a burst starts at a non-protected area and ends inside the protected area, the entire burst is not performed even if protection was disabled. To remedy:

- Either no protection areas are defined in the EEPROM
- Software uses the parallel access mechanism
- The burst can be split into two bursts where the first burst ends at the boundary of the non-protected area

5.2.5 Initial EEPROM Programming

In most applications, initial EEPROM programming is done directly on the EEPROM pins. Nevertheless, it is desirable to enable existing software utilities (accessing the EEPROM via the host interface) to initially program the whole EEPROM without breaking the protection mechanism. Following a power-up sequence, the Ethernet controller reads the hardware initialization words in the EEPROM. If the signature in word 12h does not equal 01b the EEPROM is assumed as non-programmed. There are two effects for non-valid signature:

- The Ethernet controller stops reading EEPROM data and sets the relevant registers to default values
- The Ethernet controller enables access to any location in the EEPROM via the EEPROM CSR registers including the protected EEPROM space (for the **631xESB/632xESB**, bit 4 in word 12h must not be set)

5.2.6 Activating the Protection Mechanism

Following Ethernet controller initialization, it reads the EEPROM. It then turns on the protection mechanism if word 12h [15:14] contains a valid signature (equals 01b) and if required, a hidden area with a non-zero size is defined (for the **631xESB/632xESB** bit 4 in word 12h must be set). Once the protection mechanism is turned on, word 12h becomes write-protected and the area that is defined by word 12h becomes hidden (i.e., R/W protected). For the **631xESB/632xESB**, the area defined by word 2Ch also becomes read only.

5.2.7 Initial EEPROM Programming

In most applications, initial EEPROM programming is done directly on the EEPROM pins. Nevertheless, it is desirable to enable existing software utilities (accessing the EEPROM via the host interface) to initially program the whole EEPROM without breaking the protection mechanism. Following a power-up sequence, the Ethernet controller reads the hardware initialization words in the EEPROM. If the signature in word 12h does not equal 01b the EEPROM is assumed as non-programmed. There are two effects for non-valid signature:

- The Ethernet controller stops reading EEPROM data and sets the relevant registers to default values
- The Ethernet controller enables access to any location in the EEPROM via the EEPROM CSR registers including the protected EEPROM space (for the **631xESB/632xESB**, bit 4 in word 12h must not be set)

5.3 EEPROM Firmware/Software Synchronization (631xESB/632xESB)

The EEPROM can be accessed for read/write by software and by firmware. In order to preserve data integrity, EEPROM access by firmware and software should be synchronized and serialized.

Because EEPROM sections are protected by CRC code, it is important that even EEPROM read operation is serialized with write operations, because reading a section while it is being written by another entity (before a new CRC was written) might lead to a perception of data corruption.

The FWSM and SWSM registers are used to serialize EEPROM access.

The *SWESMBI* bit in SWSM register and the *FWESMB* bit in the FWSM register are connected by hardware logic as follows: When one of them is set to 1b, the other cannot be set to 1b.

This logic is used to synchronize EEPROM access in the following manner:

- When software needs to access the EEPROM (for write or read), it:
  - Writes a 1b to the *SWESMBI* bit in SWSM.
  - Reads the *SWESMBI* bit in SWSM.
  - A value of 1b indicates that EEPROM access is granted to software.
    Software reads/writes the EEPROM.
    Software clears (writes 0b) the *SWESMBI* bit in SWSM to release the semaphore.
  - A value of 0b indicates that EEPROM access is granted to firmware. Software should repeat the access attempts.
- When firmware needs to access the EEPROM (for write or read), it:
  - Writes a 1b to the *FWESMB* bit in FWSM.
  - Reads the *FWESMB* bit in FWSM.
  - A value of 1b indicates that EEPROM access is granted to firmware.
    Firmware reads/writes the EEPROM.
    Firmware clears (writes 0b) the *FWESMB* bit in FWSM to release the semaphore.
  - A value of 0b indicates that EEPROM access is granted to software. Firmware should repeat the access attempts.

5.4 EEPROM/PHY Firmware/Software Synchronization (82571EB)

For the **82571EB**, there are two semaphores located in the Software Semaphore (SWSM) register (see Section 13.7.17).

- Bit 0 (*SWSM.SMBI*) is the software/software semaphore. This bit is needed in multi-process environments to prevent software running on one port from interferring with software on another the other port.
- Bit 1 *SWSM.SWESMBI* is the software/firmware semaphore. This is always needed when accessing the PHY or EEPROM (reads, writes, or resets). This prevents the firmware and software from accessing the PHY and or EEPROM at the same time.

Note: If resetting the PHY, the software/software semaphore should not be released until 1 ms after CFG_DONE (bit 18) is set in the MNG EEPROM Control Register (EEMNGCTL) register (see Section 13.3.25). The software/firmware semaphore should be released approximately 10 ms after the PHY reset is deasserted before polling of CFG_DONE is attempted. For details on how to reset the PHY see section Section 14.9.

For EEPROM or PHY register access:

1. Software reads *SWSM.SMBI*. If *SWSM.SMBI* is 0b, then it owns the software/software semaphore and can continue. If *SWSM.SMBI* is 1b, then some other software already has the semaphore.
2. Software writes 1b to the *SWSM.SWESMBI* bit and then reads it. If the value is 1b, then software owns the software/firmware semaphore and can continue; otherwise, firmware has the semaphore.

   Software can now access the EEPROM and/or PHY.
3. Release the software/firmware semaphore by clearing *SWSM.SWESMBI*.
4. Release the software/software semaphore by clearing *SWSM.SMBI*.

5.5 NVM Device (82573E/82573V/82573L)

The **82573E/82573V/82573L** requires non-volatile content for device configuration, log events and firmware extensions. The non-volatile memory (NVM) might contain the following four main regions:

- **LAN Configuration Space for Hardware.** This area is accessed by hardware and loaded by the **82573E/82573V/82573L** after power-up, PCI Reset de-assertion, D3 to D0 transition, or software commanded EEPROM reset (CTRL_EXT.EE_RST).
- **Firmware Space.** This space is accessed by the **82573E/82573V** in Alert Standard Format (ASF) mode or by the 82573E in Intel® Active Management Technology (Intel® AMT) mode. In ASF mode, the **82573E/82573V** loads the data following power-up, ASF Soft Reset (ASF FRC_RST), or software commanded ASF EEPROM read (ASF FRC_EELD). In Intel® AMT mode, this space is protected against software access and the firmware may access it at any time.
- **LAN Configuration Space for Software.** This space is used by software only. Register descriptions are listed here as a convention for the software only and are ignored by the Ethernet controller.
- **Boot Expansion Space.** This is accessed by software and is used by the BIOS at boot time.

5.5.1 Supported NVM Devices

Predecessors to the **82573E/82573V/82573L** required both an EEPROM and Flash device for storing LAN data. However, the **82573E/82573V/82573L** reduces the bill of material (BOM) cost by consolidating the EEPROM and Flash into a single non-volatile memory device. The NVM is connected to a single SPI interface. In addition, the **82573E/82573V/82573L** reduces the BOM by enabling a solution that merges the BIOS and **82573E/82573V/82573L** storage into a single shared Flash device. Shared Flash with the BIOS is valuable for Intel® AMT, ASF and basic functionality.

The **82573E/82573V/82573L** is compatible with many sizes of 4-wire SPI EEPROM devices. The required EEPROM size is dependent upon the manageability platform. The **82573E/82573V/82573L** accesses the EEPROM at a frequency of 2 MHz and supports EEPROM devices from STM*, SST*, and Chingis*.

The **82573E/82573V/82573L** can operate with an SPI Flash as a stand alone device or shared device with the system BIOS. The Flash size is selected by the system integrator according to its usage. However, a minimum 4 Mb Flash is required for Intel® AMT support.

5.5.2 NVM Device Detection

The **82573E/82573V/82573L** detects the device connected on the SPI interface in two phases:

1. It first detects the device type by the state of the NVM Type (NVMT) strapping pin.
2. It looks at the NVM content depending on a valid signature in word 12h of the NVM device.

5.5.2.1 CRC Field

In ASF Mode, the **82573E/82573V**'s ASF function reads the ASF CRC word to determine whether 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 **82573E/82573V/82573L** configuration, including the Management Control register.

5.5.3 Device Operation with EEPROM

When the **82573E/82573V/82573L** is connected to an external EEPROM it provides similar functionality to its predecessors with the following enhancements:

- Enable a complete parallel interface for Read and Write to the EEPROM.
- Allow software to explicitly specify the address length, eliminating the need for bit clocking access even with an empty EEPROM.

5.5.4 Device Operation with Flash

The **82573E/82573V/82573L** merges the legacy EEPROM and Flash content in a single Flash device. This mechanism provides a seamless backward compatible interface for the software and firmware to the legacy EEPROM space. This also enables the **82573E/82573V/82573L** to share the BIOS content with the LAN content on the same device in separate regions.

The **82573E/82573V/82573L** supports Flash devices with block erase size of 256 bytes and 4 Kb. The **82573E/82573V/82573L** firmware relates to logical sectors of 4 Kb regardless of the block erase size.

Note: Many Flash vendors use the term sector differently. For purposes of this document, the term Flash sector refers to a logic section of 4 Kb.

In a shared Flash, the LAN data and BIOS content reside side by side. Each entity spans across non-overlapping regions and sectors. The BIOS is located at the top of the Flash and the LAN content is located at the bottom of the Flash.

5.5.5 EEPROM Mode

When an external EEPROM is present, any access to the EEPROM interface is directed to the external EEPROM device.

5.5.6 NVM Clients

Several client software components can access the NVM through their software device driver, BIOS, firmware, and hardware. These software components and their interfaces are detailed in the *82573 NVM Map and Programming Information Guide*.

5.6 EEPROM/NVM Map

The following table indicates the EEPROM/NVM map to be used for the Ethernet controller.

Note: More detailed information regarding the EEPROM/NVM map can be obtained from the appropriate EEPROM/NVM Map and Programming Information Guide.

Table 5-2. Ethernet Controller EEPROM Map

Word	Used By[a]	High Byte (15:8)	Low Byte (7:0)	Image Value	LAN 0/1[b]
00h	HW	Ethernet Address Byte 2	Ethernet Address Byte 1	IA(2,1)	LAN 0/1
01h	HW	Ethernet Address Byte 4	Ethernet Address Byte 3	IA(4,3)	LAN 0/1
02h	HW	Ethernet Address Byte 6	Ethernet Address Byte 5	IA(6,5)	LAN 0/1
03h 04h	SW	Compatibility (High Byte)	Compatibility (Low Byte)	 0000h	Both
05h	SW	EEPROM Major Version Compatibility (High Byte) for the **82573E/82573V/82573L**	EEPROM Minor Version Compatibility (Low Byte) for the **82573E/82573V/82573L**	XXXXh	
06h 07h	SW	OEM Configuration Compatibility (High/Low Byte) for the **82573E/82573V/82573L**		0000h[c] 0000h[c]	Both
08h	SW	PBA Byte 1	PBA Byte 2		
09h	SW	PBA Byte 3	PBA Byte 4		
0Ah	HW	Initialization Control 1			Both
0Bh	HW	Subsystem ID			Both
0Ch	HW	Subsystem Vendor ID			Both
0Dh	HW	Device ID			LAN 0
0Eh	HW	Vendor ID			Both
0Fh	HW	Initialization Control 2			Both
10h	HW	Software Defined Pins Control		XXXXh	LAN 1
11h	HW	Device ID		1096h (**631xESB/ 632xESB** only)	LAN 1
12h	HW	EEPROM Sizing & Protected Fields			Both
82573E/82573V/82573L 10h:12h					
10h	HW	NVM Word 0 (NVM0)			
11h	HW	NVM Word 1 (NVM1)			
12h	HW	NVM Word 2 (NVM2)			
13h	HW	Management Enable Bits	Management Capabilities		Both
14h	HW	Initialization Control 3		XXXXh	LAN 1
15h 16h	HW	Reserved			

Table 5-2. Ethernet Controller EEPROM Map

Word	Used By[a]	High Byte (15:8)	Low Byte (7:0)	Image Value	LAN 0/1[b]
17h	FW	Firmware Start Address (Including PHY Initialization Address); (Reserved **631xESB/632xESB**)		0100h	Both
82573E/82573V/82573L 14h:17h					
14h	HW	Extended Configuration Word 1			
15h	HW	Extended Configuration Word 2			
16h	HW	Extended Configuration Word 3			
17h	HW	Reserved	Memory Scrub Control		
18h	HW	PCIe* Initialization Configuration 1			Both
19h	HW	PCIe* Initialization Configuration 2			Both
1Ah	HW	PCIe* Initialization Configuration 3			Both
1Bh	HW	PCIe* Control			Both
1Ch	HW	LEDCTL 1 3 Default			Both
1Ch	HW	PHY Configuration **82573E/82573V/82573L**	LEDCTL 1 Default **82573E/82573V/82573L**		
1Dh	HW	Reserved			
1Eh	HW	Device Revision ID			Both
1Fh	FW	LEDCTL 0 2 Default			Both
20h	HW	Software Defined Pins Control (Firmware Bits **82573E/82573V/82573L**)		XXXXh	LAN 0
21h	HW	Functions Control (Reserved **82573E/82573V/82573L**)			
22h	HW	LAN Power Consumption			Both
23h	HW	Management Hardware Configuration Control (Flash Software Detection **82573E/82573V/82573L**)		XXXXh	Both
24h	HW	Initialization Control 3		XXXXh	LAN 0
25h: 2Eh		Reserved **82573E/82573V/82573L**			
25h: 2Ch		Reserved **82571EB/82572EI**			
2D	SW	**82571EB/82572EI** only ETrack_ID (high)			
2E	SW	**82571EB/82572EI** only ETrack_ID (low)			
631xESB/632xESB 25h:2Eh					
25h 26h	HW	Reserved			
27h	HW	CRID3	CRID2		Both
28h	HW	CRID1			Both
29h	HW	Hardware Setup		D080h	Both
2Ah	HW	Reserved			Both

Table 5-2. Ethernet Controller EEPROM Map

Word	Used By[a]	High Byte (15:8)	Low Byte (7:0)	Image Value	LAN 0/1[b]
2Bh	HW	Parallel Flash Info		0004h	MNG
2Ch	HW	End of RO Area			Both
2Dh	HW	LAN Boot Control			MNG
2Eh	HW	Function Control 2			MNG
2Fh		Vital Product Data (VPD) Pointer **Note:** OEM configurable, see Section 5.6.6 for detailed description. Reserved **82573E/82573V/82573L**			
30h	SW	PXE Word 0 (Software Use) Configuration			
31h	SW	PXE Word 1 (Software Use) Configuration			
32h	SW	PXE Word (Software Use) PXE Version			
33h	SW	PXE Word (Software Use) EFI Version			
34h 35h	SW	PXE Word (Copies of Words 30h and 31h) (**82571EB** and **631xESB/632xESB** only)			LAN 1
36h	SW	Reserved			
37	SW	Alternate MAC Address			
38h 3Ch	SW	Reserved			
3Dh	SW	iSCSI Boot Configuration (Start Address) Not applicable to **82573E/82573V/82573L**			
3Eh	SW	Reserved			
3Fh		Software Checksum (words 00h to 3Fh)			
	FW	**82571EB/82572EI** Reserved for SerDes/PCIe* Ana and PHY Init (~200 word for D step)			Both
1C0h . . . 1FFh	FW	**82571EB/82572EI** Saved for Firmware (TBL, Loadable Code . . .)			Both
		Secured			

a. This column specifies whether this byte is used by hardware (HW), software (SW) or firmware (FW). EEPROM words can also be used by Preboot eXecution Environment (PXE) code or Alert Standard Format (ASF).
b. This column specifies whether the byte is used by LAN 0, LAN 1 or both. LAN 1 is not applicable to the **82572EI/82573E/82573V/82573L**.
c. FFFFh for the **82571EB/82572EI**.

Table 5-3. EEPROM Manageability Map

Word	Used By[a]	High Byte (15:8)	Low Byte (7:0)	Image Value	LAN 0/1[b]
40h: FFh		ASF/Pass Through/BMC Configuration Area			
100h/ 40h	HW	MNG D0 Power Consumption			MNG
101h/ 41h	HW	MNG D3 Power Consumption			MNG
102h/ 42h	HW	IDE Device ID		(1084h **631xESB/ 632xESB**)	MNG
103h/ 43h	HW	Serial Port Device ID		(1085h **631xESB/ 632xESB**)	MNG
104h/ 44h	HW	IPMI/KCS Device ID		(1086h **631xESB/ 632xESB**)	MNG
105h/ 45h	HW	IDE Subsystem ID			MNG
106h/ 46h	HW	Serial Port Subsystem ID			MNG
107h/ 47h	HW	IPMI/KCS Subsystem ID			MNG
108h/ 48h		IDE Boot Control (Future Request Timeout **82573E/82573V/82573L**)			IDE
109h/ 49h ... 10Dh/ 4Dh		Reserved **82571EB/82572EI** and **631xESB/632xESB**			
82573E/82573V/82573L 49h:4Dh					
49h	HW	Functions Control			
4Ah	HW	Flash Parameters			
4Bh	HW	Boot Expansion Addresses			
4Ch	HW	Boot Expansion Size			
4Dh		Reserved			
10Eh/ 4Eh	HW	IPMI/KCS Device Class Code (low)			MNG
10Fh/ 4Fh	HW	IPMI/KCS Device Class Code (high)			
631xESB/632xESB 110h/50h:11Ah/5Ah					
110h/ 50h	HW	UHCI Device ID		1087h	PCIe*
111h/ 51h	HW	Reserved		1088h	PCIe*

Word	Used By[a]	High Byte (15:8)	Low Byte (7:0)	Image Value	LAN 0/1[b]
112h/ 52h	HW	BT Device ID		1089h	PCIe*
113h/ 53h	HW	UHCI Subsystem ID		0000h	PCIe*
114h/ 54h	HW	Reserved		0000h	PCIe*
115h/ 55h	HW	BT Subsystem ID		0000h	PCIe*
116h/ 56h	HW	Reserved		0000h	PCIe*
117h/ 57h	HW	BT Function Class Code Low		0702h	PCIe*
118h/ 58h	HW	BT Function Class Code High		0Ch	PCIe*
119h/ 59h	HW	Mng D0 Power Consumption 2			PCIe*
11Ah 5Ah	HW	Mng D3 Power Consumption 2			PCIe*
11Bh 5Bh ...	FW	Firmware and Software Area (**631xESB/632xESB**) Structures Located According to Internal Firmware Pointers			Both
	FW	**82571EB/82572EI** Reserved for SerDes/PCIe* Ana and PHY Init (~200 word for D step)			Both
Flex Size up to 7Eh	FW Flex Filter	**82571EB/82572EI** Flexible Filter Data			Both
1 Word	FW Flex Filter	**82571EB/82572EI** Flexible Filter Length			Both
Flex Size		**82571EB/82572EI** Reserved			
1C0h . . . 1FFh	FW	**82571EB/82572EI** Saved for Firmware (TBL, Loadable Code . . .)			Both
		Secured			

a. This column specifies whether this byte is used by hardware (HW), software (SW) or firmware (FW). EEPROM words can also be used by Preboot eXecution Environment (PXE) code or Alert Standard Format (ASF).
b. This column specifies whether the byte is used by LAN 0, LAN 1 or both. LAN 1 is not applicable to the **82572EI/82573E/ 82573V/82573L**.

Note: For the **631xESB/632xESB**, if any of the MNG function (IDE, KT, KCS, UHCI or BT) is enabled in the PCIe* configuration space, then the EEPROM size should be at least 256 bytes if the *PCICL* bit (word 13h, bit 15) is set, or 512 bytes if the *PCICL* bit is cleared.

5.6.1 Hardware Accessed Words

This section describes the EEPROM words that are loaded by the Ethernet controller hardware. Most of these bits are located in configuration registers. The words are only read and used if the signature field in the EEPROM Sizing & Protected Fields (word 12h) is valid.

Note: When changing the default value of a reserved bit, Ethernet controller behavior is undefined.

5.6.1.1 Ethernet Address (Words 02h:00h)

The Ethernet Individual Address (IA) is a 6-byte field that must be unique for each Ethernet port and each copy of the EEPROM/NVM image. The first three bytes are vendor specific. The value from this field is loaded into the Receive Address Register 0 (RAL0/RAH0) after resets and after being in D3.

The Ethernet address is loaded for LAN0 and bit 41, the least significant bit of the last Ethernet address byte, and is inverted (bit 8 of Word 2) for LAN1[1].

Note: A default value of FFFFh means the word is not used for any purpose.

5.6.1.2 Initialization Control 1 (Word 0Ah)

This word read by the Ethernet controller contains initialization values that:

- Set defaults for some internal registers.
- Enable or disable specific features.
- Determine which PCI configuration space values are loaded from the EEPROM/NVM.

Table 5-4. Initialization Control 1 (Word 0Ah)

Bit(s)	Name	Default	Description
15	Link Status Change Wake Enable	0b	This bit enables wake on link status change as part of APM wake capabilities. For the **82573E/82573V/82573L**, this bit is not automatically read by hardware. Software reads the NVM and writes it to hardware. Reserved for the **631xESB/632xESB**, always set to 0b.
14	Link Status Change Wake Override	0b	If this bit is 1b, wake on link status change does not depend on the LNKC bit in the Wake Up Filter Control (WUFC) register. Instead, it is determined by the APM settings in the Wake Up Control (WUC) register. For the **82573E/82573V/82573L**, this bit is not automatically read by hardware. Software reads the NVM and writes it to hardware. Reserved for the **631xESB/632xESB**, always set to 0b.
13	Base Area 40h	1b	For the **82573E/82573V/82573L**, when this bit is set, it indicates that the Base Area starting at address 40h must be loaded from the NVM. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**, always set to 1b.
12	Reserved	0b	Reserved. Should be set to 0b. Reserved for the **631xESB/632xESB**, always set to 1b.

1. LAN1 is not applicable to the **82572EI** or **82573E/82573V/82573L**.

Table 5-4. Initialization Control 1 (Word 0Ah)

Bit(s)	Name	Default	Description
11	FRCSPD	0b	Default setting for the *Force Speed* bit in the Device Control register (CTRL[11]).
10	FD	1b	Default for duplex setting. Mapped to Device Control register bit 0 and Transmit Configuration Word register bit 5 (TXCW[5]). The hardware default value is 1b. **Note:** The default value is 0b for the **82573E/82573V/82573L**.
9	LRST	0b	Default setting for link reset (CTRL[3]). It should set to 0b for hardware to initiate Auto-Negotiation (for SerDes mode) upon power up or assertion of a PCIe* reset without driver intervention. The image default value is 0b. Reserved for **82573E/82573V/82573L**. Should be set to 1b.
8:7	Reserved	00b	Reserved. Should be set to 00b.
6[a]	Mask g_fnc_ tar_g MSB	1b	When set to 1b, the MSB bit of g_fnc_tar_g is masked by the EEPROM or Flash.
6:5	Reserved	00b	Reserved. Should be set to 00b. For the **82571EB/82572EI**, reserved. Should be set to 01b. Bit 5 is reserved for the **82573E/82573V/82573L**. Should be set to 1b.
4	ILOS	0b	This bit represents the default setting for the Loss-of-Signal polarity setting of CTRL[7]. The hardware default value is 0b. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should be set to 0b.
3	Power Management	1b	When set to 1b (default), full support for power management is enabled. When set to 0b, the power management register settings are read only and the Ethernet controller does not execute a hardware transition to D3. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should be set to 1b.
2	Reserved	1b	Reserved, should be set to 1b. Reserved for the **631xESB/632xESB** and **82573E/82573V/82573L**. Should be set to 0b.
1	Load Subsys-tem ID	1b	When this bit equals 1b, the Ethernet controller loads its PCIe* Subsystem ID and Subsystem Vendor ID from the EEPROM words 0Bh and 0Ch. For the **82573E/82573V/82573L**: When this bit equals 0b, the Ethernet controller loads the default PCI Subsystem ID and Subsystem Vendor ID.
0	Load Vendor/ Device ID	1b	When this bit is set to 1b, the Ethernet controller loads its PCIe* Vendor ID and Device ID from the EEPROM/NVM words 0Dh, 0Eh, and 11h (0Dh and 0Eh only for the **82573E/82573V/82573L**).

a. **82573E/82573V/82573L** only.

5.6.1.3 Subsystem ID (Word 0Bh)

If the Load Subsystem IDs bit in the Initialization Control Word 1 (0Ah) is set, this word is used to initialize the Subsystem ID. Its default value is 0h.

5.6.1.4 Subsystem Vendor ID (Word 0Ch)

If the Load Subsystem IDs bit in the Initialization Control Word 1 (0Ah) is set, this word is used to initialize the Subsystem Vendor ID. Its default value is 8086h.

5.6.1.5 Device ID (Word 0Dh, 11h)

If the *Load Vendor/Device IDs* bit in the Initialization Control Word 1 (0Ah) is set, this word is used to initialize the Device ID of LAN0 and LAN1 functions, respectively. Its default value is 105Eh for the **82571EB/82572EI** and 1096h for the **631xESB/632xESB**.

For the **82573E/82573V/82572L**, if the Load Vendor/Device IDs bit in the Initialization Control Word 1 (0Ah) is set, this word is used to initialize the Device ID of the LAN function. For a basic **82573** design, the device ID is 108Bh, for an **82573E** design with Intel® AMT, it is 108Ch, and for an **82573L** design, the device ID is 109Ah.

5.6.1.6 Vendor ID (Word 0Eh)

If the Load Vendor/Device IDs in the Initialization Control Word 1 (0Ah) is set, this word is used to initialize the Vendor ID. Its default value is 8086h.

5.6.1.7 Initialization Control 2 (Word 0Fh)

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

- Set defaults for some internal registers.
- Enable and disable specific features.

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

Bit(s)	Name	Default	Description
15	APM PME# Enable	0b	The *APM PME# Enable* bit represents the initial value of the Assert PME On APM Wakeup bit in the Wake Up Control Register (WUC.APMPME).
14	Reserved	0b	Reserved, should be set to 0b.
13:12	Pause Capability	11b	This bit enables the desired PAUSE capability for the advertised configuration base page. It is mapped to TXCW[8:7].
13:12[a]	NVMTYPE	00b	These bits indicate the type of NVM present. 00b = EEPROM. 01b = Stand-alone Flash. 10b = SPI Flash. 11b = Reserved.
11:8[a]	NVSIZE	0000b	When the NVM is a Flash device, the NVSIZE should be greater than or equal to 9. The minimum supported Flash size is 64 KB.
11	ANE	0b	This bit enables Auto-Negotiation (for SerDes mode) and is mapped to TXCW[31]. Not applicable to the **82573E/82573V/82573L**.
10:8	Serial Flash Size Indication	001b	These bits indicate the Flash size according to the following equation: Flash size = 64 KB(X)^2 (where X is a 3-digit binary number from 1 to 7 in decimal). Valid Flash size indications range from 64 KB(1)^2 = 64 KB(7)^2 = 8 MB. The Flash size impacts the requested memory space for the Flash and Expansion ROM BARs in PCIe* configuration space. Not applicable to the **82573E/82573V/82573L**.
7	CLK_CNT_1_4	1b	This bit enables the automatic reduction of the DMA frequency and is mapped to the Status register bit 30. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Always set to 0b.
6	PHY Power Down Enable	1b	This bit enables the PHY to power down. When it is set, the PHY can enter into a low power state. 0b = Disable. 1b = Enable.
5	Reserved	1b	Reserved. Should be set to 1b.
5[a]	MAC_CSR_MNG	0b	When set, the mng_mac_csr FSM is reset on IN_BAND PCIe* Reset or PERST. When cleared, the FSM is reset on a soft reset.
4	CCM PLL Shutdown Enable	1b	When this bit is set, the device can shut down the CCM PLL in low power states when the PHY is in power-down mode (for example, link disconnect). When it is cleared, the CCM PLL is not shut down in a low-power state. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should be set to 0b.
3	DMA Dynamic Gating Enable	0b	When this bit is set, dynamic clock gating of the DMA and MAC units is enabled. Reserved for the **631xESB/632xESB** and the **82571EB/82572EI**, always set to 0b. For the **82573E/82573V/82573L**, should be set to 1b.
2	Enable SerDes Power Down	0b	When this bit is set, the SerDes can enter a low power state when the function is in Dr state. 0b = Disabled. 1b = Enabled. Reserved for the **631xESB/632xESB**, should be set to 0b. Reserved for the **82573E/82573V**, should be set to 1b.

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

Bit(s)	Name	Default	Description
2[b]	Clock Power Manage-ment	0b	For the **82573L**, a value of 1b indicates that the Ethernet controller supports the removal of any reference clocks when the link is in the L1 and L2/3 ready link states. A value of 0b indicates that the Ethernet controller does not have this capability; therefore, reference clocks must not be removed in these link states. This feature is only applicable in form factors that support the clock request signal (CLKREQ#). For a multi-function Ethernet controller, each function indicates its capability independently. Power management configuration software must only permit reference clock removal if all functions of the multi-function Ethernet controller indicate a 1b for this bit. **Note:** For the **82573E** and **82573V**, this bit is reserved and should be set to 1b.
1	Enable Speed	0b	Smart Power Down. When set, enables PHY Smart Power Down Mode. Reserved for the **631xESB/632xESB**. Should be set to 0b.
1[a]	Wake DMA Dynamic Clock Gating Disable	1b	When this bit is set, dynamic clock gating of the wake DMA clock is disabled in the D0a state with wakeup enabled or MNG enabled or in the auto-read process.
0	Reserved	1b	Reserved. Should be set to 1b. For the **631xESB/632xESB**, must be set to 0b.
0[a]	D0a DMA Dynamic Clock Gating Disable	0	When this bit is set, dynamic clock gating of the DMA clock in the D0a state is disabled.

a. **82573E/82573V/82573L** only.
b. **82573L** only.

5.6.1.8 NVM0 Word 10h (82573E Only)

Note: NVM0 is used by Intel® AMT.

Table 5-6. NVM0 Word 10h (82573E Only)

Bit	Name	Default	Description
15:8	Reserved	00h	Reserved
7	ICH7 (Check for Intel® AMT Disable)	0b	This firmware configuration bit is only for Ethernet controller stepping A1 and is not accessed by hardware. For Ethernet controller A3 stepping, this bit must be set to 0b. 1b = ICH7 Check for Intel® AMT mode disable. 0b = ICH7 Check for Intel® AMT mode enable.
6	Ignore Intel® AMT SKU	0b	This firmware configuration bit is only for Ethernet controller stepping A1 and is not accessed by hardware. For Ethernet controller A3 stepping, this bit must be set to 0b. 1b = Ignore Intel® AMT SKU. 0b = Don't ignore Intel® AMT SKU.

Table 5-6. NVM0 Word 10h (82573E Only)

5:4	Reserved	00b	Reserved
3:2	MNGM	00b	This field selects one of the manageability operation modes. 00b = MNG disable (clock gated). 01b = ASF mode. 10b = Pass through mode. 11b = Intel® AMT mode.[a]
1:0	Reserved		Reserved

a. When Intel® AMT is enabled, all MNG functions (KCS, Serial, IDE) should be enabled in word 49h of the NVM.

5.6.1.9 NVM1 Word 11h (82573E Only)

Note: NVM1 is used by Intel® AMT.

Table 5-7. NVM1 Word 11h (82573E Only)

Bit	Name	Default	Description
15:8	FSECER	1111b 1111b	This field defines the instruction code for the block erase used by the **82573E**. The erase block size is defined by the SECSIZE field in address 12h.
7:0	BADDR	0000b 0000b	The BADDR defines the starting address of the BIOS space in a shared Flash as follows: BIOS Word Address = 4K * BADDR.

5.6.1.10 NVM2 Word 12h (82573E/82573V/82573L)

Table 5-8. NVM2 Word 12h (82573E/82573V/82573L)

Bit	Name	Default	Description
15:8	Reserved	0111b 1110b	These bits are reserved.
7:4	Reserved for the **82573E/V**	0001b	These bits are reserved and should be set to 0001b.
7:4	Reserved for the **82573L**	0101b	These bits are reserved and should be set to 0101b.
3:2	SECSIZE	01b	The Flash sector size is defined as: 00b = 256 bytes. 01b = 4 KB. 10b = Reserved. 11b = Reserved.
1:0	Reserved	00b	These bits are reserved and should be set to 00b.

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

This configures initial settings for the Software Definable Pins.

Note: Word 10h is for LAN1 (Port 1/Port B) and Word 20h is for LAN0 (Port 0/Port A).

Table 5-9. Software Defined Pins Control (Word 10h and 20h)

Bit(s)	Name	Default	Description
15	SDPDIR [3]	0b	SDP3 Pin - Initial Direction. This bit configures the initial hardware value of the *SDP3_IODIR* bit in the Extended Device Control (CTRL_EXT) register following power up. This relates to the SDP0/ SDP1 ports, respectively, for LAN 0/LAN 1. 0b = Input. 1b = Output. Reserved for the **631xESB/632xESB**. Should be set to 0b.
14	SDPDIR [2]	0b	SDP2 Pin - Initial Direction. This bit configures the initial hardware value of the *SDP2_IODIR* bit in the Extended Device Control (CTRL_EXT) register following power up. This relates to the SDP0/ SDP1 ports, respectively, for LAN 0/LAN 1. 0b = Input. 1b = Output.
13:12	Reserved	00b	Reserved. Should be set to 00b.
11	LAN_DIS	0b	LAN Disable. When this bit is set to 1b, the appropriate LAN is disabled. 0b = Enable. 1b = Disable.
10	LAN_PCI_ DIS	0b	LAN PCI Disable. When this bit is set to 1b, the appropriate LAN PCI function is disabled. For example, the LAN is functional for MNG operation but is not connected to the host through PCIe*. 0b = Enable. 1b = Disable.
9	SDPDIR [1]	0b	SDP1 Pin - Initial Direction. This bit configures the initial hardware value of the *SDP1_IODIR* bit in the Device Control (CTRL) register following power up. This relates to the SDP0/SDP1 ports, respectively, for LAN 0/LAN 1. 0b = Input. 1b = Output.
8	SDPDIR [0]	0b	SDP0 Pin - Initial Direction. This bit configures the initial hardware value of the *SDP0_IODIR* bit in the Device Control (CTRL) register following power up. This relates to the SDP0/SDP1 ports, respectively, for LAN 0/LAN 1. 0b = Input. 1b = Output.
7	SDPVAL [3]	0b	This bit holds the value of the SDP3 pin (Initial Output Value). It configures the initial power-on value output of SDP3 when it is configured as an output. This is accomplished by configuring the initial hardware value of the SDP3_DATA bit in the Extended Device Control (CTRL_EXT) register after power up. This relates to the SDP0/SDP1 ports, respectively, for LAN 0/LAN 1. Reserved for the **631xESB/632xESB**. Should be set to 0b.

1. Not applicable to the **82573E/82573V/82573L**.

Table 5-9. Software Defined Pins Control (Word 10h and 20h)

Bit(s)	Name	Default	Description
6	SDPVAL [2]	0b	SDP2 Pin - Initial Output Value. This bit configures the initial power on value output of SDP2 (when it is configured as an output) by configuring the initial hardware value of the *SDP2_DATA* bit in the Extended Device Control (CTRL_EXT) register after power up. This relates to the SDP0/SDP1 ports, respectively, for LAN 0/LAN 1.
5	Reserved	0b	This bit is reserved and should be set to 0b.
4	Gigabit Disable	0b	When this bit is set, the Gigabit Ethernet operation is disabled. An example of when this might be used is if Gigabit Ethernet operation exceeds system power limits. Reserved for the **631xESB/632xESB and 82571EB/82572EI**. Must be set to 0b.
3	Disable 1000 in non-D0a	1b	When set to 1b, disables 1000 Mb/s operation in non-D0a states. Reserved for the **631xESB/632xESB**. Must be set to 0b.
2	D3COLD_ WAKEUP_ ADVEN	1b	Configures the initial hardware default value of the *ADVD3WUC* bit in the Device Control register (CTRL) after power up. 1b = Advertised 0b = Not advertised.
1	SDPVAL [1]	0b	SDP1 Pin - Initial Output Value. This bit configures the initial power on value output of SDP1 (when it is configured as an output) by configuring the initial hardware value of the *SDP1_DATA* bit in the Device Control (CTRL) register after power up. This relates to the SDP0/SDP1 ports, respectively, for LAN 0/LAN 1.
0	SDPVAL [0]	0b	SDP0 Pin - Initial Output Value. This bit configures the initial power on value output of SDP0 (when it is configured as an output) by configuring the initial hardware value of the *SDP0_DATA* bit in the Device Control (CTRL) register after power up. This relates to the SDP0/SDP1 ports, respectively, for LAN 0/LAN 1.

5.6.1.12 EEPROM Sizing & Protected Fields (Word 12h)[1]

Note: The software device driver can only read this word. It has no write access to this word through the EEC and EERD registers. Write access is possible only through an authenticated firmware interface.

Table 5-10. EEPROM Sizing & Protected Fields (Word 12h)

Bit(s)	Name	Default	Description
15:14	Signature	01b	The Signature field indicates to the device 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.
13:10	EEPROM Size	1000b	These bits indicate the actual EEPROM size: 0000b = 128 bytes 0001b = 256 bytes 0010b = 512 bytes 0011b = 1 KB 0100b = 2 KB 0101b = 4 KB 0110b = 8 KB 0111b = 16 KB 1000b = 32 KB 1001b = 64 KB 1010b - 1011b = Reserved
9:5	Reserved	00000b	Reserved. Should be set to 00000b. For the **82573L**, set bit 6 to 1b.
4	Enable EEPROM Protection	0b	If set, all EEPROM protection schemes are enabled. Reserved for the **82571EB/82572EI**.
3:0	Reserved	0000b	Reserved. Set to 0000b.

1. Not applicable to the **82573E/82573V/82573L**.

5.6.1.13 Management Enable (Word 13h, High Byte)

This byte contains information for firmware regarding enabled manageability functions. After this byte is updated, the driver should notify firmware of the change. If the manageability subsystem is in a mode where its host interface is active, then it should be done by the manageability host command. If the host interface is inactive, the device driver should wake the manageability subsystem by asserting the *Wake Management Clock (WMNG)* bit in the Software Semaphore (SWSM) register.

The Management Enable Byte word 13h must be set to 000b (no manageability) when using the ASF EEPROM image. It is the ASF agent's responsibility to set this field to 001b for ASF mode.

Notes: For the **82571EB/82572EI**, If the additional PCI functions (SMS, IDE, COM) are enabled in the EEPROM, it indicates to the firmware that their relevant functionality is enabled.

When no manageability mode is present or enabled and no other functionality is enabled (SMS, IDE, COM), the manageability subsystem is not functional and the host interface is not functional. In other words, it does not respond to accesses made to CSR addresses 8800h through 8FFFh.

The EEPROM contains information for the enabled mode only. Additional information on mode operation (fail over enable, active port, etc.) is available in the manageability CSR area if the block is functional.

For the **82573E/82573V/82573L**, if the additional PCI functions (KCS, IDE, COM) are enabled in the NVM, it indicates to the firmware that their relevant functionality is enabled.

5.6.1.14 Management Capabilities (Word 13h, Low Byte)

This word contains the Ethernet controller's manageability capabilities and is only used by device software. It should not enable any capability (in the upper byte) that is not enabled in this byte. The OEM is responsible for initializing this byte. For the **631xESB/632xESB**, this byte is reserved for use by firmware.

5.6.1.15 Extended Configuration Word 1 Word 14h (82573E/82573V/82573L)

Table 5-11. Extended Configuration Word 1 Word 14h (82573E/82573V/82573L)

Bit	Name	Default	Description
15:13	Reserved	000b	These bits are reserved and should be set to 000b.
12	Wake Behavior	0b	This field defines behavior of the WAKE# signal. 0b = WAKE# signal behaves as specified in the PCIe* Specification. 1b = Always assert WAKE# on PME. (This is asserted in all system modes.)
11:0	Reserved	0h	These bits are reserved and should be set to 0h.

5.6.1.16 Extended Configuration Word 2 Word 15h (82573E/82573V/82573L)

Table 5-12. Extended Configuration Word 2 Word 15h (82573E/82573V/82573L)

Bit	Name	Default	Description
15:0	Reserved	00D8h	These bits are reserved and should be set to 00D8h.

5.6.1.17 Extended Configuration Word 3 Word 16h (82573E/82573V/82573L)

Table 5-13. Extended Configuration Word 3 Word 16h (82573E/82573V/82573L)

Bit	Name	Default	Description
15:0	Reserved	00h	These bits are reserved and should be set to 00h.

5.6.1.18 Memory Scrub Control / PCIe* Delay Word 17h (82573E/82573V/82573L)

Table 5-14. Memory Scrub Control / PCIe* Delay Word 17h (82573E/82573V/82573L)

Bit	Name	Default	Description
15:13	Reserved	000b	These bits are reserved and should be set to 000b.
12:8	Electrical Idle Delay	00111b	This field identifies the delay cycles before entering the electrical idle allowing data path flush. The default value for this field is 7h.
7:0	Reserved	00h	These bits are reserved and should be set to 00h.

5.6.1.19 Initialization Control 3 (Word 14h, 24h Low Byte)[1]

Table 5-15. Initialization Control 3 (Word 14h and 24h Low Byte)

Bit(s)	Name	Default	Description
7:1	Reserved	00h	Reserved. Should be set to 00h.
0	No PHY Reset for IDE	0b	No PHY reset when IDE or SOL is enabled. When asserted, this bit can prevent the PHY reset signal according to the CTRL.BLK_PHY_RST value (MANC.BLK_PHY_RST for the **82571EB/82572EI**). This bit should be set to the same value at both words (14h, 24h) to reflect the same option to both LANs. 0b = PHY reset always asserted by a PCIe* reset. 1b = PHY reset blocked by firmware. For the **82573E/82573V/82573L**, when this bit is asserted, the PHY resets and the power changes reflected in the PHY, according to the MANC. BLK_PHY_RST value, are prevented.

1. Word 24h only for the **82573E/82573V/82573L** and **82572EI**.

5.6.1.20 Initialization Control 3 (Word 14h, 24h High Byte)[1]

This word controls general initialization values.

Note: If applicable, word 14h is used for LAN1 (Port 1/Port B). Word 24 is used for LAN0 (Port 0/Port A).

Table 5-16. Initialization Control 3 (Word 14h and 24h High Byte)

Bit(s)	Name	Default	Description
15	APM Flexible Filter Allocation	0b	This bit is loaded to the APMFFA bit in the Wake Up Control (WUC) register and allocates the flexible TCO1 filter for APM wake. It is read by firmware into hardware. For the **631xESB/632xESB** and **82571EB/82572EI**, this bit is reserved and should be set to 0b.
14	Multiple Read Request Enable	1b	When this bit is set to 0b, the Ethernet controller initiates one transmit DMA request at a time. When it equals 1b, the Ethernet controller can initiate up to 4 outstanding multiple transmit DMA requests. This bit sets the default value of the Multiple Read Request (MULR) bit in the Transmit Control (TCTL) register (bit 28). Reserved for the **631xESB/632xESB**. Should be set to 1b.
13	LAN Flash Disable	1b	For the **82571EB/82572EI**, a bit value of 1b disables the Flash logic; the Flash access BAR in the PCI Configuration space is also disabled. For NICs with Flash, always set this bit to 0b. For the **631xESB/632xESB**, The flash accessed by the LAN is defined in word 2Bh. For the **82573E/82573V/82573L**, this bit is reserved and should be set to 1b.
12	Interrupt Pin	0b for LAN 0 1b for LAN 1	This bit controls the value advertised in the *Interrupt Pin* field of the PCI Configuration header for this device and function. A value of 0b reflected in the *Interrupt Pin* field indicates that this device uses INTA#; a value of 1b indicates that this device uses INTB#. If only a single port of the Ethernet controller is enabled, this value is ignored and the *Interrupt Pin* field of the enabled port reports INTA# usage.

1. Word 24h only for the **82573E/82573V/82573L** and **82572EI**.

Table 5-16. Initialization Control 3 (Word 14h and 24h High Byte)

Bit(s)	Name	Default	Description
11	LAN Boot Disable	1b	A value of 1b disables the Expansion ROM BAR in the PCI Configuration space. For **82571EB/82572EI** LOMs, always set this bit to 1b; for NICs, this bit is used to control whether the option ROM is enabled or not. For the **82573E/82573V/82573L**, this bit is reserved and should be set to 1b.
10	APM Enable	1b	This field reflects the initial value of Advanced Power Management Wake Up Enable in the Wake Up Control Register (WUC.APME) and is mapped to CTRL[6] and to WUC[0]. 0b = Disable APM. 1b = Enable APM.
9:8	Link Mode	00b	This field reflects the 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. 00b = MAC operates in 10/100/1000Base-T mode with the internal copper PHY (1000BASE-T). For the **631xESB/632xESB**, MAC operates in GLCI or SerDes mode. • 01b = Reserved. • 10b = Reserved. 11b = MAC operates in TBI mode using internal SerDes module. For the **631xESB/632xESB**, MAC operates in SerDes mode. For the **82573E/82573V/82573L**, these bits are reserved and should be set to 00b.

Below is the description of bits 13 and 11 in various combinations:

Flash Disable (Bit 13)	Boot Disable (Bit 11)	Functionality (Active Windows)
0b	0b	Flash and Expansion ROM Bars are active.
0b	1b	Flash BAR is enabled and Expansion ROM BAR is disabled.
1b	0b	Flash BAR is disabled and Expansion ROM BAR is enabled.
1b	1b	Flash and Expansion ROM BARs are disabled.

5.6.1.21 Firmware Start Address Word 17h (82571EB/82572EI)

This field contains the firmware start address including the PHY initialization area.

Table 5-17. Firmware Start Address (Word 17h)

Bit(s)	Name	Default	Description
15:0	Address	0100h	This field defines the word address in the EEPROM of the PHY and SerDes initialization space. The first words in the initialization space define the area ID (or PHY Initialization), its size and a pointer to the next block address. It is firmware responsibility to use the content of the initialization space for PHY and SerDes initialization and read the next fields.

5.6.1.22 PCIe* Initialization Configuration 1 (Word 18h)

This field:

- Sets default values for some internal registers.
- Enables or disables specific features.

Table 5-18. PCIe* Initialization Configuration 1 (Word 18h)

Bit(s)	Name	Default	Description
15	Reserved	0b	Reserved. Should be set to 0b.
14:12	L1 Act Exit Latency	110b	This field represents the L1 active exit latency for the configuration space. When it is set to 110b, the latency range is 32 µs to 64 µs. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should be set to 110b.
11:9	L1 Act Accept Latency	110b	This field represents the L1 active acceptable latency for the configuration space. When it is set to 110b, the acceptable latency range is 32 µs to 64 µs. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should be set to 110b.
8:6	L0s Accept Latency	011b	This field represents the L0s acceptable latency for the configuration space. When it is set to 011b, the acceptable latency is 512 ns. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should be set to 011b. **Note:** Default value is 110b for the **82573E/82573V/82573L**.
5:3	L0s Separated Exit Latency	001b	This field represents the L0s exit latency for active state power management with a separated reference clock. When it is set to 001b, the latency range is between 64 ns and 128 ns. Reserved for the **631xESB/632xESB**. Should be set to 001b. Reserved for the **82571EB/82572EI**. Should be set to 110b.
2:0	L0s Common Exit Latency	001b	This field represents the L0s exit latency for active state power management with a common reference clock. When it is set to 001b, the latency range is between 64 ns and 128 ns. Reserved for the **631xESB/632xESB**. Should be set to 001b. Reserved for the **82571EB/82572EI**. Should be set to 110b.

5.6.1.23 PCIe* Initialization Configuration 2 (Word 19h)

This word sets default values for some internal registers.

Table 5-19. PCIe* Initialization Configuration 2 (Word 19h)

Bit(s)	Name	Default	Description
15	DLLP Timer Enable	0b	When it is set to 1b, the DLLP timer counter is enabled. For the **631xESB/632xESB** and **82571EB/82572EI**, this bit is reserved and should be set to 0b.
14	DISMN2CSR	0b	Asserting this bit disable a read or write transaction to the CSR from MNG. The default value of 0b enables the MNG to access the CSR. For the **631xESB/632xESB** and **82571EB/82572EI**, this bit is reserved and should be set to 0b.
13	DEV TYPE	1b	This bit defines the device type as reported in the PCIe* capability register of the IDE function. 0b = Native End Point. 1b = Legacy End Point. **Note:** DEVTYPE must be set to 1b since all other functions are defined as native devices. For the **82571EB/82572EI**, this bit is reserved and should be set to 1b.
12	Reserved	1b	Reserved. Should be set to 1b.
11:8	Extra NFTS	0111b 0001b[a] 0011b[b]	This field identifies the extra Number of Fast Training Signal (NFTS), which is added to the original requested number of NFTS (as requested by the upstream component). For the **631xESB/632xESB**, these bits are reserved and should be set to 0011b. For the **82571EB/82572EI**, these bits are reserved and should be set to 0111b.
7:0	NFTS	B0h	This field identifies the number of special sequence for L0s transition to L0. For the **631xESB/632xESB**, these bits are reserved and should be set to FFh. For the **82571EB/82572EI**, these bits are reserved and should be set to B0h.

a. **82573E/82573V/82573L** only.
b. **631xESB/632xESB** only.

5.6.1.24 PCIe* Initialization Configuration 3 (Word 1Ah)

This word sets default values for some internal registers.

Table 5-20. PCIe* Initialization Configuration 3 (Word 1Ah)

Bit(s)	Name	Default	Description
15	Master Enable	0b	When this bit is set to 1b, the PHY can act as a master (upstream component with cross link functionality). For the **631xESB/632xESB** and **82571EB/82572EI**, this bit is reserved and should be set to 0b.
14	Scramble Disable	0b	When this bit is set to 1b, the PCIe* LFSR scrambling feature is disabled. For the **631xESB/632xESB**, this bit is reserved and should be set to 0b.
13	Ack/Nak Scheme	0b	This field identifies the acknowledgement/no acknowledgement scheme for the Ethernet controller. 0b = Scheduled for transmission following any TLP. 1b = Scheduled for transmission according to time-outs specified in the PCIe* specification. For the **82571EB/82572EI**, this bit is reserved and should be set to 0b.
12	Cache Line Size	0b	This bit represents the cache line size. 0b = 64 bytes. 1b = 128 bytes. **Note:** The value loaded must be equal to the actual cache line size used by the platform as configured by system software. For the **82571EB/82572EI**, this bit is reserved and should be set to 0b.
11:10	Reserved	01b	These bits are reserved and must be set to 01b.
9	IO Support	1b	This bit represents the status of I/O support (I/O BAR request). When it is set to 1b, I/O is supported. For the **82571EB/82572EI**, this bit is reserved and should be set to 1b.
8	Max Packet Size	1b	This bit identifies the status of the default packet size. 0b = 128 bytes. 1b = 256 bytes. For the **82571EB/82572EI**, this bit is reserved and should be set to 1b.
7:6	Lane Width	10b	This field identifies the maximum link width. 00b = 1 lane. 01b = 2 lanes. 10b = 4 lanes. 11b = Reserved. These bits are reserved for the **631xESB/632xESB**. Must be set to 10b. These bits are reserved for the **82573E/82573V/82573L**. Should be set to 00b.
5	Elastic Buffer Control	1b	When this bit equals 1b, the elastic buffers operate under phase-only mode during electrical idle states. For the **631xESB/632xESB**, this bit is reserved and should be set to 0b. For the **82571EB/82572EI**, this bit is reserved and should be set to 1b.
4	Elastic Buffer Diff1	0b	When this bit is set to 1b, the elastic buffers are activated in a more limited mode (read and write pointers). For the **631xESB/632xESB** and **82571EB/82572EI**, this bit is reserved and should be set to 0b.

Table 5-20. PCIe* Initialization Configuration 3 (Word 1Ah)

Bit(s)	Name	Default	Description
3:2	Active State PM Support	01b 00b[a]	This field determines support for Active State Link Power Management. It is loaded into the PCIe* Active State Link PM Support register. For the **631xESB/632xESB**, these bits are reserved and should be set to 00b. For the **82571EB/82572EI**, these bits are reserved and should be set to 01b.
1	Slot Clock Cfg	1b	When this bit is set, the Ethernet controller uses the PCIe* reference clock supplied on the connector. This is primarily used for add-in solutions. For the **631xESB/632xESB** and **82571EB/82572EI**, this bit is reserved and should be set to 1b.
0	Loopback Polarity Inversion	0b	This field verifies the polarity inversion in loopback master entry. For the **631xESB/632xESB** and **82571EB/82572EI**, this bit is reserved and should be set to 0b.

a. **631xESB/632xESB** only.

5.6.1.25 PCIe* Control (Word 1Bh)

This word configures initial settings for the PCIe* default functionality.

Table 5-21. PCIe* Control (Word 1Bh)

Bit(s)	Name	Default	Description
15	GIO Receive Valid[a]	1b	This bit identifies if a receiver presence is detected. When set, the **82573E/82573V/82573L** overrides the receiver (partner) detection status. Reserved for the **631xESB/632xESB**. Should always be set to 0b. Reserved for **82571EB/82572EI**. Should always be set to 1b.
14	Reserved	0b	Reserved. Should be set to 0b. Reserved for the **82573L**. Should be set to 1b.
13	GIO Down Reset Disable	0b	This bit disables a core reset when the PCIe* link goes down. 0b = Enable. 1b = Disable. Reserved for the **631xESB/632xESB**. Should always be set to 1b. Reserved for **82571EB/82572EI**. Should always be set to 0b.
12	Lane Reversal Disable GIO LTSSM[a]	0b	This bit disables the ability to negotiate lane reversal. 0b = Enable. 1b = Disable. Reserved for the **631xESB/632xESB**. Should always be set to 1b. For the **82573E/82573V/82573L**, when this bit is cleared, LTSSM complies with the SlimPIPE specification (power mode transition). Reserved for **82571EB/82572EI**. Should always be set to 0b.

Table 5-21. PCIe* Control (Word 1Bh)

Bit(s)	Name	Default	Description
11	Good Recovery Extended FTS[a]	0b	When set to 1b, the LTSSM Recovery states always progresses towards LinkUp (force a good recovery, when a recovery occurs). 0b = Normal mode. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should always be set to 0b. **Note:** For the **82573E/82573V/82573L**, when this bit is set, the Ethernet controller sets the Extended NFTS bit in TS1/TS2 according to the PCIe* specification. Also, when the bit is set the upstream sends 4096 NFTS to the Ethernet controller.
10	Reserved	1b	Reserved. Should always be set to 1b.
9:7	Reserved	000b	Reserved, always set to 000b.
6	Reserved	0b	Reserved. Should always be set to 0b.
5	L2 Disable	0b	This bit disables the link from entering L2 state. 0b = Enable. 1b = Disable. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should always be set to 0b.
4	Skip Disable	0b	This bit disables the SKIP symbol insertion in the elastic buffer. 0b = Enable. 1b = Disable. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should always be set to 0b.
3	L0s Clock Gate Enable	0b	This bit disables the clock gating when the device enters the L0s state. Its default value of 0b disables the clock gating. 0b = Disables clock gating. 1b = Enables clock gating. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should always be set to 0b.
2	Electrical Idle Electrical Idle Mask[a]	0b	Electrical Idle Mask. When set to 1b, disables the check for illegal electrical idle sequence (for example, eidle ordered set without common mode and vise versa). Also excepts any of them as a correct eidle sequence. 0b = Enable. 1b = Disable **Note:** Specification can be interpreted so that the eidle ordered set is sufficient for transition to power management states. The use of this bit allows an exception for such interpretation and avoids the possibility of correct behavior being understood as illegal sequences. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should always be set to 0b.
1:0	Latency to Enter L1	11b	These bits identify the period in L0s state before transitioning into an L1 state. 00b = 64 µs 01b = 256 µs 10b = 1 ms 11b = 4 ms Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should always be set to 11b.

a. **82573E/82573V/82573L** only.

5.6.1.26 PHY Configuration Word 1Ch, High Byte (82573E/82573V/82573L)

The high byte of this word contains the PHY configuration bits loaded to register 25 in page 0 of the PHY configuration space.

Table 5-22. PHY Configuration Word 1Ch, High Byte (82573E/82573V/82573L)

Bit	Name	Default	Description
15	Reserved	0b	This bit is reserved and should be set to 0b.
14	Gigabit Disable	0b	When this bit is set, Gigabit Ethernet operation is disabled in all modes.
13	Reserved	0b	This bit is reserved and should be set to 0b.
12	Class AB	0b	When this bit is set, the PHY operates in Class A mode rather than Class B mode. This mode only applies for 1000BASE-T operation. 10BASE-T and 100BASE-TX operation continues to run in Class B mode by default regardless of this signal value.
11	Disable Gigabit in Non-D0a	1b	This bit disables Gigabit Ethernet operation in non-D0 active states.
10	LPLU	0b	This bit represents the low power link up status. When this bit is set, it enables the decrease in link speed in non-D0 active states when the power policy and power management states require it.
9	D0 LPLU	0b	This bit represents the low power link up status in D0 active states. When this bit is set, it enables the decrease in link speed in D0 active states when the power policy and power management states require it.
8	Reserved	1b	This bit is reserved and should be set to 1b.

5.6.1.27 LED Control (82573E/82573V/82573L)

Table 5-23. LED Control (82573E/82573V/82573L)

Field	Bits	Initial Value	Description
Reserved	31:24	0b	These bits are reserved and should be set to 0b.
LED 2 Blink	23	0b[a]	LED2 (Link 100) Blink.
LED 2 Invert	22	0b[a]	LED2 (Link 100) Invert.
LED 2 Blink Mode	21	0b[a]	LED2 (Link 100) Blink Mode. This field must be configured with the same value as the Global Blink Mode field. It specifies the LED blink mode. 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off.
Reserved	20	0b	This bit is reserved and should be set to 0b.
LED 2 Mode	19:16	0110b[a]	LED2 (Link 100) Mode. This field specifies the control source for the LED2 output. An initial value of 0110b selects Link 100 indication.
LED 1 Blink	15	0b[a]	LED1 (Activity) Blink.
LED 1 Invert	14	0b[a]	LED1 (Activity) Invert.

Table 5-23. LED Control (82573E/82573V/82573L)

Field	Bits	Initial Value	Description
LED 1 Blink Mode	13	0b[a]	LED1 (Activity) Blink Mode. This field must be configured with the same value as the Global Blink Mode field. It specifies the LED blink mode. 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off.
Reserved	12	0b	This bit is reserved and should be set to 0b.
LED 1 Mode	11:8	0011b[a]	LED1 (Activity) Mode. This field specifies the control source for the LED1 output. An initial value of 0011b selects Activity indication.
LED 0 Blink	7	b0[a]	LED0 (Link Up) Blink. This field controls the blink logic for the (inverted) LED control source prior to LED output. 0b = Do not blink asserted LED output. 1b = Blink asserted LED output.
LED 0 Invert	6	0b[a]	LED0 (Link Up) Invert. This field controls the blink logic for the (inverted) LED control source prior to LED output. 0b = Do not invert LED source. 1b = Invert LED source.
Global Blink Mode	5	0b[a]	Global Blink Mode. This field specifies the blink mode of all LEDs. 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off.
Reserved	4	0b	This bit is reserved and should be set to 0.
LED 0 Mode	3:0	0010b[a]	LED0 (Link Up) Mode. This field specifies the control source for LED0. An initial value of 0010b selects Link Up indication.

a. These bits are read from the NVM.

Table 5-24. LED Control Source (82573E/82573V/82573L)

Mode	Selected Mode	Source Indication
0000b	Link 10/1000	Asserted when either a 10 Mb/s or a 1000 Mb/s link is established and maintained.
0001b	Link 100/1000	Asserted when either a 100 Mb/s or a 1000 Mb/s link is established and maintained.
0010b	Link Up	Asserted when any speed link is established and maintained.
0011b	Reserved	Reserved.
0100b	Link/Activity	Asserted when any speed link is established and when no transmit or receive activity is present.
0101b	Link 10	Asserted when a 10 Mb/s link is established and maintained.
0110b	Link 100	Asserted when a 100 Mb/s link is established and maintained.
0111b	Link 1000	Asserted when a 1000 Mb/s link is established and maintained.

Table 5-24. LED Control Source (82573E/82573V/82573L)

Mode	Selected Mode	Source Indication
1000b	Reserved	Reserved.
1001b	Full Duplex	Asserted when the link is configured for full duplex operation.
1010b	Reserved	Reserved.
1011b	Activity	Asserted when link is established and packets are being transmitted or received.
1100b	Reserved	Reserved.
1101b	Reserved	Reserved.
1110b	LED On	Always asserted.
1111b	LED Off	Always de-asserted.

NOTES:
1. When LED Blink is enabled, the appropriate LED Invert bit should be set to 0b.
2. The dynamic LED modes (Link/Activity and Activity) should be used with LED Blink enabled.
3. When LED Blink is enabled and CCM PLL is stopped, the blinking frequencies will be one-fifth of the rates stated in Table 5-24.

5.6.1.28 LED 1-3 Configuration Defaults (Word 1Ch)

This EEPROM/NVM word specifies the hardware defaults for the LEDCTL register fields controlling the LED1 (ACTIVITY indication) and LED3 (LINK_1000 indication) output behaviors.

Table 5-25. LED 1-3 Configuration Defaults (Word 1Ch)

Bit(s)	Name	Default	Description
15	LED3 Blink	0b	This bit represents the initial value of the LED3_BLINK field. If it equals 0b, the LED is non-blinking. Not applicable to the **82573E/82573V/82573L**.
14	LED3 Invert[a]	0b	This bit represents the initial value of the LED3_IVRT field. If it equals 0b, it is an active low output. Not applicable to the **82573E/82573V/82573L**.
13	LED3 Blink Mode	0b[b]	This field represents the LED3 Blink Mode. It must be configured with the same value as Global Blink Mode (word 1Fh, bit 5). 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off. For the **631xESB/632xESB**, Blink at 83 ms on with no defined off time. Not applicable to the **82573E/82573V/82573L**.
12	Reserved	0b	This bit is reserved and should be set to 0b. Not applicable to the **82573E/82573V/82573L**.
11:8	LED3 Mode	0111b	This field represents the initial value of the LED2_MODE specifying the event, state, and pattern displayed on the LED2 (LINK_1000) output. A value of 0111b (or 7h) causes this to indicate 1000 Mb/s operation. See Table 5-26 for all available LED modes. Not applicable to the **82573E/82573V/82573L**.
7	LED1 Blink	1b	This field holds the initial value of LED1_BLINK field and is equal to 0b for non-blinking.
6	LED1 Invert[a]	0b	This field holds the initial value of LED1_IVRT field and is equal to 0b for an active low output.
5	LED1 Blink Mode	0b[b]	This field represents the status of LED1 Blink Mode. It must be configured with the same value as Global Blink Mode (word 1Fh, bit 5). 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off. For the **631xESB/632xESB**, Blink at 83 ms on with no defined off time.
4	Reserved	0b	This bit is reserved and should be set to 0b.
3:0	LED1 Mode	0011b	This field represents the initial value of the LED1_MODE specifying the event, state, and pattern displayed on the LED1 (ACTIVITY) output. A value of 0011b (3h) causes this to indicate ACTIVITY state. See Table 5-26 for all available LED modes.

a. When LED Blink mode is enabled, the appropriate LED invert bit should be set to 0b.
b. These bits are read from the EEPROM/NVM.

Note: For the **82571EB/82572EI**, the LINK/ACTIVITY source functions are slightly different from the others when BLINK is enabled. The LED is off if there is no LINK, on if there is LINK and no ACTIVITY, and blinks if there is LINK and ACTIVITY.

Note: Asserted = active low.

Table 5-26. LED Modes[a]

Mode	Selected Mode	Source Indication
0000b	LINK_10/1000	Asserted when either 10 or 1000 Mb/s link is established and maintained.
0001b	LINK_100/1000	Asserted when either 100 or 1000 Mb/s link is established and maintained.
0010b	LINK_UP	Asserted when any speed link is established and maintained.
0011b	FILTER_ACTIVITY	Asserted when link is established and packets are being transmitted or received that passed MAC filtering.
0100b	LINK/ACTIVITY	Asserted when link is established AND when there is NO transmit or receive activity.
0101b	LINK_10	Asserted when a 10 Mb/s link is established and maintained.
0110b	LINK_100	Asserted when a 100 Mb/s link is established and maintained.
0111b	LINK_1000	Asserted when a 1000 Mb/s link is established and maintained.
1000b	SDP_MODE	LED activation is a reflection of the SDP signal. SDP0, SDP1, SDP2, SDP3 are reflected to LED0, LED1, LED2, LED3, respectively.
1001b	FULL_DUPLEX	Asserted when the link is configured for full duplex operation.
1010b	COLLISION	Asserted when a collision is observed.
1011b	ACTIVITY	Asserted when link is established and packets are being transmitted or received.
1100b	BUS_SIZE	Asserted when the device detects a 1 Lane PCIe* connection.
1101b	PAUSED	Asserted when the device's transmitter is flow controlled.
1110b	LED_ON	Always asserted.
1111b	LED_OFF	Always de-asserted.

a. Not applicable to the **82573E/82573V/82573L**.

NOTE:
1. The dynamic LED modes (Link/Activity and Activity) should be used with LED Blink mode enabled.

5.6.1.29 Device Revision ID (Word 1Eh)

Table 5-27. Device Revision ID (Word 1Eh)

Bit(s)	Name	Default	Description
15	Device Power-Down Disable Device Off Enable[a]		When set, enables the Ethernet controller to enter power down. 0b = Disable. 1b = Enable. Reserved for the **631xESB/632xESB**. Should always be set to 0b. For the **82571EB/82572EI**, always set to 0b when enabling any manageability modes. **Note:** When this bit is set, the **82573E/82573V/82573L** can enter the Device Disable mode and the Dr Disable mode. When it is cleared, both modes are disabled.
14	MNG_ RST_FIX	1b	Manageability Reset Change Enable. When set to 1b, enables the reset scheme change so that the MNG block can access the CSR registers at Dr. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Must be set to 1b.
13	SMBus Timeout Cfg	0b	0b = Timeout determined by clock or by data low. 1b = Determined by clock low. Reserved for the **631xESB/632xESB** and **82573E/82573V/82573L**. Should always be set to 0b.
12	iSCSI LAN 1	0b	When set to 1b, the function 1 class code is iSCSI class code (010000h). When set to 0b, the function 1 class code is LAN class code (020000h). Reserved for the **631xESB/632xESB** and **82573E/82573V/82573L**. Should always be set to 0b.
11	iSCSI LAN 0	0b	When set 1b, the function 0 class code is iSCSI class code (010000h). When set to 0b, the function 0 class code is LAN class code (020000h). Reserved for the **631xESB/632xESB** and **82573E/82573V/82573L**. Should always be set to 0b.
10:8	High Power Control	011b	Reserved. Always set to 011b. Reserved for the **631xESB/632xESB**. Always set to 100b.
8[a]	GIO D3 Gate	0b	This bit enables or disables GIO gate clocking in the D3 power state. This feature is enabled in a future release of the Ethernet controller.
7:0	DEVREVID	03h	Device Rev ID. The actual device revision ID is the EEPROM value. The value is XORed with 05h. Always set to 03h. Reserved for the **631xESB/632xESB**, the actual device ID is the EEPROM value. Should be set to FFh. Reserved for **82573E/82573V/82573L**, the actual device ID is the EEPROM value. Should be set to 0h.

a. **82573E/82573V/82573L** only.

5.6.1.30 LED 0-2 Configuration Defaults (Word 1Fh)

This EEPROM word specifies the hardware defaults for the LEDCTL register fields controlling the LED0 (LINK_UP) and LED2 (LINK_100) output behaviors.

Table 5-28. LED 0-2 Configuration Defaults (Word 1Fh)

Bit(s)	Name	Default	Description
15	LED2 Blink	0b	This bit represents the initial value of the LED2_BLINK field. If it equals 0b, the LED is non-blinking.
14	LED2 Invert[a]	0b	This bit represents the initial value of the LED2_IVRT field. If it equals 0b, it is an active low output.
13	LED2 Blink Mode	0b[b]	This field represents the LED2 Blink Mode. It must be configured with the same value as Global Blink Mode (word 1Fh, bit 5). 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off. For the **631xESB/632xESB**, Blink at 83 ms on with no defined off time.
12	Reserved	0b	This bit is reserved and should be set to 0b.
11:8	LED2 Mode	0110b	This field represents the initial value of the LED2_MODE specifying the event, state, and pattern displayed on the LED2 (LINK_1000) output. A value of 0110b (or 6h) causes this to indicate 100 Mb/s operation. See Table 5-26 for all available LED modes.
7	LED0 Blink	0b	This field holds the initial value of LED0_BLINK field and is equal to 0b for non-blinking.
6	LED0 Invert[a]	0b	This field holds the initial value of LED0_IVRT field and is equal to 0b for an active low output.
5	LED0 Blink Mode	0b[b]	This field represents the LED0 blink mode. 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off. For the **631xESB/632xESB**, Blink at 83 ms on with no defined off time.
4	Reserved	0b	This bit is reserved and should be set to 0b.
3:0	LED0 Mode	0010b	This field represents the initial value of the LED0_MODE specifying the event, state, and pattern displayed on the LED0 (ACTIVITY) output. A value of 0010b (2h) causes this to indicate link up state. See Table 5-26 for all available LED modes.

a. When LED Blink mode is enabled, the appropriate LED invert bit should be set to 0b.
b. These bits are read from the EEPROM/NVM.

5.6.1.31 Firmware Configuration Word 20h (82573E/82573V/82573L)

Table 5-29. Firmware Configuration Word 20h (82573E/82573V/82573L)

Bit	Name	Default	Description
15	Power Save Enable	0b	This bit indicates whether power save is enabled. 0b = Power Save disabled. 1b = Power Save enabled.
14	IDE Register Enable	0b	This bit indicates whether the IDE registers are enabled. 0b = IDE firmware default mode is disabled (no IDE device). 1b = IDE firmware default mode is enabled (IDE device present).
13:1	Reserved	0h	These bits are reserved and should be set to 0h.
0	Force TCO Reset Disable	0b	This bit indicates whether a Force TCO Reset is enabled or disabled. 0b = LAN Force TCO enabled. 1b = LAN Force TCO disabled.

5.6.1.32 Functions Control (Word 21h)[1]

Table 5-30. Functions Control (Word 21h)

Bit(s)	Name	Default	Description
15	IDE Enable	0b	This bit enables the IDE function in the PCI Configuration Space. When this bit is cleared, the IDE configuration space is not visible to the system. This bit is reflected in the Function Active and Power State to Management (FACTPS) register.
14	Serial Enable[a]	0b	This bit enables the Serial Port function in the PCI Configuration Space. When this bit is cleared, the Serial Port configuration space is not visible to the system. This bit is reflected in the FACTPS register.
13	IPMI/KCS Enable	0b	This bit enables the IPMI/KCS function in the PCI Configuration Space. When this bit is cleared, the IPMI/KCS configuration space is not visible to the system. This bit is reflected in the FACTPS register. For the **82571EB/82572EI**, must be cleared (0b).
12	LAN Function Select	0b	When both LAN ports are enabled and the LAN function select equals 0b, LAN 0 is routed to PCI function 0 and LAN 1 is routed to PCI function 1. If the LAN function select bit equals 1b, LAN 0 is routed to PCI function 1 and LAN 1 is routed to PCI function 0. This bit is mapped to FACTPS[30] register.
11:10	IDE INT Select	11b	This bit reflects the default setup of the IDE Interrupt Pin. The value is loaded to the Interrupt Pin register in the PCI configuration space. The default value is INT D. 00b = INT A. 01b = INT B. 10b = INT C. 11b = IND D.

1. Not applicable to the **82573E/82573V/82573L**.

Table 5-30. Functions Control (Word 21h)

Bit(s)	Name	Default	Description
9:8	Serial INT Select	10b	This bit reflects the default setup of the Serial INT Interrupt Pin. The value is loaded to the Interrupt Pin register in the PCI configuration space. The default value is INT C. 00b = INT A. 01b = INT B. 10b = INT C. 11b = IND D.
7:6	IPMI/KCS INT Select	11b	This bit reflects the default setup of the IPMI/KCS Interrupt Pin. The value is loaded to the Interrupt Pin register in the PCI configuration space. The default value is INT D. 00b = INT A. 01b = INT B. 10b = INT C. 11b = IND D.
5	Reserved	X	Reserved for the **631xESB/632xESB**. Should be set to 0b. Reserved for the **82571EB/82572EI**. Should be set to 1b.
4:0	Reserved	00000b 10000b[b]	This field is reserved and should be set to 00000b.

a. SOL enable over PCI (bit 14) and over LPC are **mutually exclusive**. As a result, only one should be enabled in the EEPROM.
b. **82571EB/82572EI** only.

5.6.1.33 LAN Power Consumption (Word 22h)

This word is meaningful only if the EEPROM/NVM signature in word 0Ah is valid and Power Management is enabled.

Table 5-31. LAN Power Consumption (Word 22h)

Bit(s)	Name	Default	Description
15:8	LAN D0 Power	0h	The value in this field is reflected in the PCI Power Management Data Register of the LAN functions for D0 power consumption and dissipation (Data_Select = 0 or 4). Power is defined in 100 mW units and includes the external logic required for the LAN function.
7:5	Function 0 Common Power	0h	The value in this field is reflected in the PCI Power Management Data Register of function 0 when the Data_Select field is set to 8 (common function). The most significant bits in the Data Register that reflect the power values are padded with zeros.
4:0	LAN D3 Power	0h	The value in this field is reflected in the PCI Power Management Data Register of the LAN functions for D3 power consumption and dissipation (Data_Select = 3 or 7). Power is defined in 100 mW units and includes the external logic required for the LAN function. The most significant bits in the Data Register that reflect the power values are padded with zeros.

5.6.2 Flash Software Detection Word 23h (82573E/82573V/82573L)

Table 5-32. Flash Software Detection Word 23h (82573E/82573V/82573L)

Bit	Name	Default	Description
15	Checksum Validity	0b	This bit indicates whether the checksum has been verified and updated by software. 0b = Software has not yet updated the checksum. 1b = Software has already updated the checksum.
14:8	Reserved	1111111b	This field is reserved and all bits should be set to 1b.
7:0	Reserved	1111b 1111b	This field must have all bits set to 1b.

5.6.2.1 Management Hardware Configuration Control (Word 23h)

This word contains bits that direct special firmware behavior when configuring the PHY/PCIe*/ SerDes. Contact your Intel Field Service Representative for more information.

5.6.3 CRID3 Word 27h, High (631xESB/632xESB)

Table 5-33. CRID3 Word 27h, High (631xESB/632xESB)

Bit	Name	Default	Description
15:8	CRID3	00h	CRID value read in case the key written in Rev ID is 3Dh.

5.6.4 CRID2 Word 27h, Low (631xESB/632xESB)

Table 5-34. CRID2 Word 27h, Low (631xESB/632xESB)

Bit	Name	Default	Description
7:0	CRID2	00h	CRID value read in case the key written in Rev ID is 2Dh.

5.6.4.1 CRID1 Word 28h, High (631xESB/632xESB)

Table 5-35. CRID1 Word 28h, High (631xESB/632xESB)

Bit	Name	Default	Description
15:8	CRID1	00h	CRID value read in case the key written in Rev ID is 1Dh.

5.6.4.2 631xESB/632xESB Specific Hardware Setup (Word 29h)

These bits are used for the **631xESB/632xESB** specific hardware setups (read by autoload).

Table 5-36. 631xESB/632xESB Specific Hardware Setup (Word 29h)

Bit	Name	Default	Description
15	MSI En	1b	Enable MSI in all functions. Influence only the capability declaration in PCIe* config space. 0b = Disable. 1b = Enable.
14	Reserved	0b	Reserved. Should be set to 0b.
13	NMI Enable	1b	If enabled, allows sending IOCHK interrupts through SERIRQ. 0b = Disable. 1b = Enable.
12	SDP Pull Up Enable	1b	If set, all SDP pads wake up with pull up, until released by firmware. If cleared, SDP pads are without pull up. 0b = Disable. 1b = Enable.
11:10	Reserved	00b	Reserved. Should be set to 00b.
9	LAN 1 iSCSI Enable	0b	When set, LAN 1 class code is set to 010000h (SCSI). When reset, LAN 1 class code is set to 020000h (LAN).
8	LAN 0 iSCSI Enable	0b	When set, LAN 0 class code is set to 010000h (SCSI). When reset, LAN 0 class code is set to 020000h (LAN).
7:4	Reserved	8h	Reserved. Should be set to 8h.
3	Reserved	0b	Reserved. Should be set 0b.
2	Reserved	1b	Reserved. Should be set 1b.
1	Reserved	0b	Reserved. Should be set 0b.
0	Reserved	1b	Reserved. Should be set 1b.

5.6.4.3 Flash Info Word 2Bh (631xESB/632xESB)

This word is read by the autoloader.

Table 5-37. Flash Info Word 2Bh (631xESB/632xESB)

Bit	Name	Default	Description
15:12	Reserved	0000b	Reserved. Should be set to 0000b.
11:8	Expansion ROM BAR size	0000b	Expansion ROM BAR Size (LAN and IDE) 0000b = 64 KB. 0001b = 128 KB. 0010b = 256 KB. 0011b = 512 KB. 0100b = 1 MB. 0101b = 2 MB. 0110b = 4 MB. 0111b = 8 MB. 1000b = 16 MB. Others = 64 KB.
7	Reserved	0b	Reserved.
6:3	Pflash_size	0000b	Parallel Flash Size 0000b = 64 KB. 0001b = 128 KB. 0010b = 256 KB. 0011b = 512 KB. 0100b = 1 MB. 0101b = 2 MB. 0110b = 4 MB. 0111b = 8 MB. 1000b = 16 MB. Others = 64 KB
2	SW Flash Access Enable	1b	This bit sets the default of bit 7 in the FLA register. 0b = Disabled 1b = Enabled.
1	Mapped Flash in Host	0b	0b = Serial. 1b = Parallel.
0	Reserved	0b	Reserved. Should be set to 0b.

5.6.4.4 End of Read Only (RO) Area Word 2Ch (631xESB/632xESB)

Defines the end of area in the EEPROM that is RO.

Table 5-38. End of RO Area Word 2Ch (631xESB/632xESB)

Bit	Name	Default	Description
15	Reserved	0b	Reserved.
14:0	EORO_ area	0h	Defines the end of area in the EEPROM that is RO. The resolution is one word. Can be up to byte address FFFFh (7FFFh words). A value of 0h, indicates no RO area.

5.6.4.5 LAN Boot Control Word 2Dh (631xESB/632xESB)

This word sets the defaults for some internal registers (read by autoload).

Table 5-39. LAN Boot Control Word 2Dh (631xESB/632xESB)

Bit	Name	Default	Description
15	Reserved	0b	Reserved.
14:0	Flash Address	0h	Defines the base address of the LAN Boot expansion ROM in the physical Flash device. Base address in bytes equals 256 times the field value.

5.6.4.6 Functions Control 2 Word 2Eh (631xESB/632xESB)

Enables an override of the default BT function subsystem ID. This word is read by autoload and reflected in the FACTPS2 register.

Table 5-40. Functions Control 2 Word 2Eh (631xESB/632xESB)

Bit	Name	Default	Description
15	UHCI Ena	0b	Enables the UHCI Function in the PCI Configuration Space. When this bit is cleared, the UHCI configuration space is not visible to the system. This bit is reflected in the FACTPS2 register. 0b = Disable. 1b = Enable.
14	Reserved	0b	Reserved. Must be set to 0b.
13	BT Ena	0b	Enables the BT Function in the PCI Configuration Space. When this bit is cleared, the BT configuration space is not visible to the system. This bit is reflected in the FACTPS2 register. 0b = Disable. 1b = Enable.
12:11	UHCI INT Sel	10b	Default setup of the UHCI Interrupt Pin. The value is loaded to the Interrupt Pin register in the PCI configuration space. Default value in INT C. 00 = INT A. 01 = INT B. 10 = INT C. 11 = INT D.
10:9	Reserved	10b	Reserved. Should be set to 10b.

Table 5-40. Functions Control 2 Word 2Eh (631xESB/632xESB)

Bit	Name	Default	Description
8:7	BT INT Sel	11b	Default setup of the BT Interrupt Pin. The value is loaded to the Interrupt Pin register in the PCI configuration space. Default value in INT D. 00 = INT A. 01 = INT B. 10 = INT C. 11 = INT D.
6	UHCI Device type	0b	0b = Declare UHCI function as type 0000b (native) PCIe* device. 1b = Declare UHCI as a type 0001b (legacy) PCIe* device.
5	UHCI memory BAR enable	1b	1b = Memory BAR enabled in function 5. 0b = Disabled.
4:0	Reserved	1Fh	Reserved.

5.6.4.7 PXE Code (Words 30h – 35h)

Words 30h through 35h have been reserved for configuration and version values to be used by PXE code.

5.6.4.8 Alternate MAC Address (Word 37h)

This word is used as a pointer to an EEPROM block that contains the space for two MAC addresses. The first three words of the EEPROM block is used to store the MAC address for the first port (PCI Function 0). The second three words of the EEPROM block is used to store the MAC address for the second port (PCI Function 1). Initial and default values in the EEPROM block should be set to FFFFh (for both addresses) indicating that no alternate MAC address is present.

Note: Word 37h must be set to FFFFh if alternate MAC addresses are not used. Also, alternate MAC addresses are ignored by hardware and require specific software support for activation.

5.6.4.9 iSCSI Boot (Word 3Dh)[1]

Word 3Dh is reserved for configuration and version values to be used by iSCSI boot code.

Table 5-41. iSCSI Boot Configuration Start Address (Word 3Dh)

Bit(s)	Name	Default	Description
15:0	Address	0b	This field defines the word address in the EEPROM space of the iSCSI Boot Configuration structure.

The iSCSI module structure is listed in the following table.

1. Not applicable to the **82573E/82573V/82573L**.

Configuration Item	Max Size (Byte)	Comments
iSCSI Boot Signature	2	This is the ASCII characters “iS”.
Total Size	2	The structure size is stored in this field and is set depending on the amount of free EEPROM space available. The total size of this structure, including variable length fields, must fit within this space.
Structure Version	1	The version number of this structure is stored in this field and should be set to 1b.
Checksum	1	This field holds the 8-bit checksum of this structure.
Flags	2	01h → Enable iSCSI Boot 02h → Valid Configuration This should be set to 1b if the configuration information in this structure is valid, and 0b, otherwise. 03h → Enable DHCP 04h:05h → Configuration Prompt 0 = 0 seconds 1 = 2 seconds 2 = 3 seconds 3 = 5 seconds 06h:0Ah → Number of Connection Retries
Initiator IP	4	If the DHCP flag is not set, this field should contain the configured IP address. If the DHCP flag is set, this field should be set to 0b or the last configured IP address should be saved.
Initiator Subnet Mask	4	If the DHCP flag is not set, this field should contain the configured subnet mask. If the DHCP flag is set, this field should be set to 0b or the last configured subnet mask should be saved.
Target IP	4	This is the IP address of the iSCSI target.
Target Port	2	This is the IP port of the iSCSI target. The default is 3260.
CHAP Password	32	
CHAP User Name	255 + 1	This is a variable length field.
Initiator Name	255 + 1	This is a variable length field.
User Name	255 + 1	This is a variable length field.

The maximum amount of boot configuration information stored is 834 bytes (417 words); however, the iSCSI boot implementation can limit this value in order to work with a smaller EEPROM.

Variable length fields are used to limit the total amount of EEPROM that is used for iSCSI boot information. Each field is preceded by a single byte that indicates how much space is available for that field. For example, if the Initiator Name field is being limited to 128 bytes, then it is preceded with a single byte with the value 128. The following field begins 128 bytes after the beginning of the Initiator Name field regardless of the actual size of the field. The variable length fields must be null terminated unless they reach the maximum size specified in the length byte.

5.6.4.10 Manageability D0 Power Consumption (Word 40h/100h)

Note: Section 5.6.4.10 through Section 5.6.4.33 only apply when manageability is used.

These words set default values for some internal registers.

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-42. Manageability D0 Power Consumption (Word 40h/100h)

Bit(s)	Name	Default	Description
15	Reserved	1b	This bit is reserved and should be set to 1b. 0b for the **631xESB/632xESB**.
14:10	IDE D0 PWR	00000b	This field holds the Power Consumption value reflected in the Data Register of the IDE function in the Power Management registers at D0 power state. The same value is reflected in the power consumption and power dissipation.
9:5	Serial D0 PWR	00000b	This field holds the Power Consumption value reflected in the Data Register of the Serial Port function in the Power Management registers at D0 power state. The same value is reflected in the power consumption and power dissipation.
4:0	SMS/KCS D0 PWR	00000b	This field holds the Power Consumption value reflected in the Data Register of the IPMI/KCS function in the Power Management registers at D0 power state. The same value is reflected in the power consumption and power dissipation.

5.6.4.11 Manageability D3 Power Consumption (Word 41h/101h)

Set defaults for some internal registers.

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-43. Manageability D3 Power Consumption (Word 41h/101h)

Bit(s)	Name	Default	Description
15	Reserved	1b	This bit is reserved and should be set to 1b.
14:10	IDE D3 PWR	00000b	This field holds the Power Consumption value reflected in the Data Register of the IDE function in the Power Management registers at D3 power state. The same value is reflected in the power consumption and power dissipation.
9:5	Serial D3 PWR	00000b	This field holds the Power Consumption value reflected in the Data Register of the Serial Port function in the Power Management registers at D3 power state. The same value is reflected in the power consumption and power dissipation.
4:0	SMS/KCS D3 PWR	00000b	This field holds the Power Consumption value reflected in the Data Register of the IPMI/KCS function in the Power Management registers at D3 power state. The same value is reflected in the power consumption and power dissipation.

5.6.4.12 IDE Device ID (Word 42h/102h)

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-44. IDE Device ID (Word 42h/102h)

Bit(s)	Name	Default	Description
15:0	IDEDID	1061h 1084h for the **631xESB/ 632xESB** 108Dh[a]	IDE Device ID. If the Load Vendor/Device IDs bit in word 0Ah is set, this word is read in to initialize the Device ID of the IDE functions.

a. **82573E/82573V/82573L** default; Intel® AMT = IDE-R.

5.6.4.13 Serial Port Device ID (Word 43h/103h)

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-45. Serial Port Device ID (Word 43h/103h)

Bit(s)	Name	Default	Description
15:0	SerialDID	1062h 1085h for the **631xESB/ 632xESB** 108Fh[a]	Serial Port Device ID. If the Load Vendor/Device IDs bit in word 0Ah is set, this word is read in to initialize the Device ID of the Serial Port functions.

a. **82573E/82573V/82573L** default; Intel® AMT = SOL.

5.6.4.14 IPMI/KCS Device ID (Word 44h/104h)

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-46. IPMI/KCS Device ID (Word 44h/104h)

Bit(s)	Name	Default	Description
15:0	IPMIDID	1063h 1086h for the **631xESB/ 632xESB** 108Eh[a]	IPMI/KCS Device ID. If the Load Vendor/Device IDs bit in word 0Ah is set, this word is read in to initialize the Device ID of the IPMI/KCS functions.

a. **82573E** default; Intel® AMT = KCS-R.

5.6.4.15 IDE Subsystem ID (Word 45h/105h)[1]

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-47. IDE Subsystem ID (Word 45h/105h)

Bit(s)	Name	Default	Description
15:0	IDESubID	0h	IDE Subvendor ID. If the Load Subsystem IDs bit in word 0Ah is set, this word is read in to initialize the IDE Subsystem ID.

5.6.4.16 Serial Port Subsystem ID (Word 46h/106h)[1]

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-48. Serial Port Subsystem ID (Word 46h/106h)

Bit(s)	Name	Default	Description
15:0	Serial SubID	0000h	Serial Port Subvendor ID. If the Load Subsystem IDs bit in word 0Ah is set, this word is read in to initialize the Serial Port Subsystem ID.

5.6.4.17 IPMI/KCS Subsystem ID (Word 47h/107h)[1]

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-49. IPMI/KCS Subsystem ID (Word 47h/107h)

Bit(s)	Name	Default	Description
15:0	SMS/KCS SubID	0000h	IPMI/KCS Subvendor ID. If the Load Subsystem IDs bit in word 0Ah is set, this word is read in to initialize the IPMI/KCS Subsystem ID.

5.6.4.18 IDE Boot Control (Word 48h/108h)

These fields set default values for some internal registers.

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-50. IDE Boot Control (Word 48h/108h)

Bit(s)	Name	Default	Description
15	IDE Boot Disable	0b	This bit disables the IDE boot expansion register in the PCI configuration space. When it is set, the IDE expansion ROM is a read only register with a zero value. By default, the IDE expansion ROM register is enabled.
14:0	Flash Address	0h	This field defines the base address of the IDE boot expansion ROM in the physical Flash device. The base address in bytes equals 256 times the field value.

1. Not applicable to the **82573V/82573L**.

5.6.4.19 Future Request Time-Out Word 48h (82573E/82573V/82573L)

Table 5-51. Future Request Time-Out Word 48h (82573E/82573V/82573L)

Bit	Name	Default	Description
15:8	Reserved	0h	These bits are reserved and should be set to 0h.
7:4	FRBTO	1111b	This field represents the boot time future request time-out. A future request locks time-out after the de-assertion of the PCIe Reset. During this time, the firmware minimizes its activity in order to maximize Flash bandwidth for the BIOS during boot code execution.[a] This time is defined as 1 s * FRBTO.
3:0	FRRTO	0000b	This field represents the run time future request time-out. A future request locks time-out during run time.[a] This time is defined as 100 ms * FRRTO.

a. Future request time-outs are implemented by firmware. After the timer expires, firmware can override the future request and initiate a long, single SPI cycle.

5.6.4.20 Functions Control Word 49h (82573E/82573V/82573L)

Table 5-52. Functions Control Word 49h (82573E/82573V/82573L)

Bit	Name	Default	Description
15	IDE Enable	0b	This bit enables the IDE function in the PCI Configuration Space. When it is set to 0b (default), the IDE configuration space is not visible to the system. This bit is reflected in the FACTPS register.
14	Serial Enable	0b	This bit enables the serial port function in the PCI Configuration Space. When it is set to 0b (default), the serial port configuration space is not visible to the system. This bit is reflected in the FACTPS register.
13	KCS Enable	0b	This bit enables the KCS function in the PCI Configuration Space. When it is set to 0b (default), the KCS configuration space is not visible to the system. This bit is reflected in the FACTPS register.
12	Reserved	0b	This bit is reserved and should be set to 0b.
11:10	IDE INT Select	01b	This field sets the default for the IDE Interrupt Pin. The value is loaded to the interrupt pin register in the PCI Configuration Space. Its default value is INT D (01b).
9:8	Serial INT Select	10b	This field sets the default for the Serial Interrupt Pin. The value is loaded to the interrupt pin register in the PCI Configuration Space. Its default value is INT C (10b).
7:6	KCS INT Select	11b	This field sets the default for the KCS Interrupt Pin. The value is loaded to the interrupt pin register in the PCI Configuration Space. Its default value is INT D (11b).
5:0	Reserved	0b	These bits are reserved and should be set to 0b.

5.6.4.21 Flash Parameters Word 4Ah (82573E/82573V/82573L)

Table 5-53. Flash Parameters Word 4Ah (82573E/82573V/82573L)

Bit	Name	Default	Description
15:8	FDEVER	1111b 1111b	This field defines the instruction code for the Flash device erase. A value of 00h indicates that the device does not support device erase. Its default value is FFh.
7:5	Reserved	000b	These bits are reserved and should be set to 000b.
4	LONGC	0b	When this bit equals 0b (default), a write cycle to the Flash is not a long cycle. When set to 1b, this indicates that the Flash write instruction is long.
3:0	Reserved	0000b	These bits are reserved and should be set to 0000b.

5.6.4.22 Boot Expansion Address Word 4Bh (82573E/82573V/82573L)

Table 5-54. Boot Expansion Address Word 4Bh (82573E/82573V/82573L)

Bit	Name	Default	Description
15:0	Reserved	0h	These bits are reserved and should be set to 0h.

5.6.4.23 Boot Expansion Size Word 4Ch (82573E/82573V/82573L)

Table 5-55. Boot Expansion Size Word 4Ch (82573E/82573V/82573L)

Bit	Name	Default	Description
15	Disable KCS Expansion	1b	When this bit is set to 1b (default), the KCS Expansion Boot ROM BAR is disabled.
14	Disable IDE Expansion	1b	When this bit is set to 1b (default), the IDE Expansion Boot ROM BAR is disabled.
13	Disable LAN Expansion	1b	When this bit is set to 1b (default), the LAN Expansion Boot ROM BAR is disabled.
12	Disable LAN Flash BAR	0b	When this bit is set to 1b (default), the LAN Flash BAR is disabled.
11:0	Reserved	00h	This bit is reserved and should be set to 00h.

5.6.4.24 KCS Device Class Code Low (Word 4Eh/10Eh)[1]

This word specifies the device class code of the KCS function. It can be either IPMI/KCS or ASF/KCS.

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-56. KCS Device Class Code Low Byte (Word 4Eh/10Eh)

Bit(s)	Name	Default	Description
15:8	Class Code Middle Word	07h	This field defines the middle byte of the IPMI class code: IPMI/KCS mode = Subclass equals 07h.
7:0	Class Code LSB	01h	This field defines the least significant byte of the IPMI class code: IPMI/KCS mode = Interface code 01h for KCS.

5.6.4.25 KCS Device Class Code High (Word 4Fh/10Fh)[1]

This word specifies the device class code of the KCS function. It may be either IPMI/KCS or ASF/KCS.

Note: The lower word is used for LAN 0, and the higher word for LAN 1.

Table 5-57. KCS Device Class Code High Byte (Word 4Fh/10Fh)

Bit(s)	Name	Default	Description
15:8	Reserved	0b	This field is reserved and should be set to 0b.
7:0	Class Code MSB	0Ch	This field defines the most significant byte of the IPMI class code: IPMI/KCS mode = Base class code equals 0Ch.

5.6.4.26 UHCI Device ID Word 110h/50h (631xESB/632xESB)

This word allows for an override of the default UHCI function device ID, is read by autoload, and is reflected in the function 5 config space device ID field.

Table 5-58. UHCI Device ID Word 110h/50h (631xESB/632xESB)

Bit	Name	Default	Description
15:0	UHCI ID	1087h	UHCI function device ID.

1. Not applicable to the **82573V/82573L**.

5.6.4.27 BT Device ID Word 112h/52h (631xESB/632xESB)

This word allows for an override of the default BT function device ID, is read by autoload, and is reflected in function 7 config space device ID field.

Table 5-59. BT Device ID Word 112h/52h (631xESB/632xESB)

Bit	Name	Default	Description
15:0	BT ID	1089h	BT function device ID.

5.6.4.28 UHCI Subsystem ID Word 113h/53h (631xESB/632xESB)

This word allows for an override of the default UHCI function subsystem ID, is read by autoload, and is reflected in function 5 config space subsystem ID field.

Table 5-60. UHCI Subsystem ID Word 113h/53h (631xESB/632xESB)

Bit	Name	Default	Description
15:0	UHCI ID	0h	UHCI function subsystem ID.

5.6.4.29 BT Subsystem ID Word 115h/55h (631xESB/632xESB)

This word allows for an override of the default BT function subsystem ID, is read by autoload, and is reflected in function 7 config space subsystem ID field

Table 5-61. BT Subsystem ID Word 115h/55h (631xESB/632xESB)

Bit	Name	Default	Description
15:0	BT ID	0h	BT function subsystem ID.

5.6.4.30 BT Device Class Code Low Word 117h/57H (631xESB/632xESB)

This word specifies the device class code of the BT function.

Table 5-62. BT Device Class Code Low Word 117h/57h (631xESB/632xESB)

Bit	Name	Default	Description
15:8	Class Code Middle Word	07h	Middle byte of the IPMI/BT class code.
7:0	Class Code LSB	02h	LSB byte of the IPMI class code.

5.6.4.31 BT Device Class Code High Word 118h/58h (631xESB/632xESB)

This word specifies the device class code of the BT function.

Table 5-63. BT Device Class Code High Word 118h/58h (631xESB/632xESB)

Bit	Name	Default	Description
15:9	Reserved	0	Reserved. Should be set to 0b.
7:0	Class Code MSB	0Ch	MSB byte of the IPMI class code. In IPMI/BT mode, these bits are the Base Class code that equals 0Ch.

5.6.4.32 Mng D0 Pwr Consumption 2 Word 119h/59h (631xESB/632xESB)

This word is used to set the defaults for some internal registers.

Table 5-64. Mng D0 Pwr Consumption 2 Word 119h/59h (631xESB/632xESB)

Bit	Name	Default	Description
15	Reserved	1b	Reserved. Should be set to 1b.
14:10	UHCID0PWR	00000b	Power Consumption value that is reflected in the Data Register of the UHCI function in the Power Management registers at D0 power state. The same value is reflected in the Power consumption and Power dissipation.
9:5	Reserved	00000b	Reserved. Should be set to 0b.
4:0	BTD0PWR	00000b	Power Consumption value that is reflected in the Data Register of the IPMI/BT function in the Power Management registers at D0 power state. The same value is reflected in the Power consumption and Power dissipation.

5.6.4.33 Mng D3 Pwr Consumption 2 Word 11Ah/5Ah (631xESB/632xESB)

This word is used to set the defaults for some internal registers (read by autoload).

Table 5-65. Mng D3 Pwr Consumption 2 Word 11Ah/5Ah (631xESB/632xESB)

Bit	Name	Default	Description
15	Reserved	1b	Reserved.
14:10	UHCID3PWR	00000b	Power Consumption value that is reflected in the Data Register of the UHCI function in the Power Management registers at D3 power state. The same value is reflected in the Power consumption and Power dissipation.
9:5	Reserved	00000b	Reserved. Should be set to 0b.
4:0	BTD3PWR	00000b	Power Consumption value that is reflected in the Data Register of the IPMI/BT function in the Power Management registers at D3 power state. The same value is reflected in the Power consumption and Power dissipation.

5.6.5 Intel® AMT MAC Address Words 80h - 82h (82573E)

The Ethernet controller requires an Intel® AMT dedicated Ethernet address (also known as the manageability MAC address). This feature supports dedicated static IP mode only if the Ethernet controller has an assigned dedicated Intel® AMT Ethernet address.

The manageability MAC address is the Intel® AMT Ethernet Individual Address (IA) and is a 6-byte field reserved for a dedicated Intel® AMT MAC address. Intel's default factory settings for these offsets are: FFh, FFh, FFh, FFh, FF, FFh.

An identical MAC address should not be used in both the LAN MAC section at word 00h to 02h and the manageability dedicated MAC section at word 80h to 82h. This type of configuration is considered invalid.

The Intel® AMT Ethernet controller assumes correct image configuration and does not perform any checksum.

5.6.6 Vital Product Data Pointer (Word 2Fh)

For the **631xESB/632xESB** and **82571EB/82572EI**, this word can be used to point to a customer writeable, 64-word Vital Product Data (VPD) block; a value of 0000h or FFFFh indicates that this field is not used.

5.6.7 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.

5.6.8 ASF Controller Words[1]

When the Ethernet controller is in ASF mode, the Ethernet controller ASF function reads the ASF section from the EEPROM/NVM. For the 82571EB/82572EI, this section is in words 40h through F7h. For the **631xESB/632xESB**, word 44h contains a pointer to the base address and the ASF section is in relative offset B6h:00h. These words are read after power up (LAN_PWR_GOOD assertion), ASF Soft Reset (ASF FRC_RST), or software commanded ASF EEPROM read (ASF FRC_EELD).

Note: These words should be programmed by ASF configuration software. The value of the words from the factory should be all FFh.

For the **82573E/82573V**, it reads the ASF words from the management section starting at the address contained in the START field. These words are read after power-up, ASF Soft Reset, or software commanded ASF NVM read. These words should be programmed by the ASF configuration software. Their values from the factory should all equal FFh.

5.6.8.1 ASF Words – Content

The interpretation of these words depends on the ASF mode functionality.

5.6.8.2 ASF Words – EEPROM Checksum (CRC)

While the ASF EEPROM/NVM words are read, the Ethernet controller also computes the ASF CRC word. Words 40h to F7h (for the **631xESB/632xESB**, words in relative offset B7h:00h) are included in the CRC calculation and compared against the CRC value present in the high byte of word F7h (relative word B7h for the **631xESB/632xESB**). If the CRC values do not match, the Ethernet controller does not overwrite the ASF configuration registers with the EEPROM/NVM values. Therefore, if the ASF CRC is invalid, hardware default values are initially present in ASF registers and any subsequent re-read of EEPROM/NVM leaves the ASF registers unchanged (from values current at the time of the EEPROM /NVM read).

The details of this CRC can be found at:

http://cell-relay.indiana.edu/cell-relay/publications/software/CRC/32bitCRC.tutorial.html

For the **82573E/82573V**, while reading the ASF NVM words, the Ethernet controller computes the ASF CRC word. Words 00h to B7h (relative to PSTART) are included in the ASF CRC calculation. The result is compared against the CRC value present in word B7h. If the CRC values mismatch, the Ethernet controller does not overwrite the ASF Configuration Registers with the NVM values. Therefore, if the ASF CRC is invalid, hardware default values will initially be present in ASF registers and any subsequent re-read of NVM leaves the ASF registers unchanged from the values at the time of the NVM read.

1. Not applicable to the **82573L**.

5.7 Software Owned EEPROM Words Description

This section describes the software owned EEPROM words (words 03h:09h). Table 5-66 lists the software owned area and the sections that follow detail the specific words.

5.7.1 EEPROM Map for Words 03h:09h

Table 5-66. EEPROM Map for Words 03h:09h

Word	Used By	15 8	7 0	Image Value	LAN 0/1
03h: 07h	SW	Compatibility 1 High (**82573E/82573V/82573L**)	Compatibility 1 Low (**82573E/82573V/82573L**)		
03h	SW	Compatibility 1 High	Compatibility 1 Low	0000h	LAN 0/1 (both)
04h	SW	OEM LED 2, 3 Configuration	OEM LED 0, 1 Configuration	0000h	
05h	SW	EEPROM Major Version	EEPROM Minor Version	0000h	
06h	SW	Reserved OEM Configuration for the **82571EB/82572EI**	Bit 3:0 for SERDES Amplitude Adjustment. Bit 7:4 Reserved OEM Configuration for the **82571EB/82572EI**	FFFFh[a] 0000h[b]	
07h	SW	Bit 15 for PHY Class A Enable, Bit 14:8 Reserved OEM Configuration for the **82571EB/82572EI**	Bit 7:0 Reserved OEM Configuration for the **82571EB/82572EI**	FFFFh[a] 0000h[b]	
08h 09h	SW	PBA, Byte 1 PBA, Byte 3	PBA, Byte 2 PBA, Byte 4		

a. **82571EB/82572EI** only.
b. **82573E/82573V/82573L** and **631xESB/632xESB** only.

5.7.2 Software Compatibility Word 1 (Word 03h)

Table 5-67. Software Compatibility Word 1 (Word 03h)

Bit	Name	Description
15:13	Reserved	Reserved. Should be set to 0b.
12	ASF SMBus Connect	0b = ASF SMBus connected. 1b = ASF SMBus not connected. Reserved for the **631xESB/632xESB** and **82571EB/82572EI**. Should be set to 0b.
11	NIC	1b = LOM. 0b = NIC.
10	Server	1b = Server. 0b = Client.
9	Client	1b = Client. 0b = Server.
8	OEM/Retail	1b = OEM. 0b = Retail.
7:6	Reserved	Reserved. Should be set to 00b.
5	Reserved	Should be set to 1b for the **82573E/82573V/82573L** and **82571EB/82572EI**. Should be set to 0b for the **631xESB/632xESB**.
4	SMBus Connected	1b = SMBus connected. 0b = SMBus not connected.
3	Reserved	Reserved. Should be set to 0b.
2	PCI Bridge	1b = PCI bridge present. 0b = PCI bridge not present.
1:0	Reserved	Reserved. Should be set to 00b.

5.7.3 OEM LED Configuration Word (Word 04h)

Table 5-68. OEM LED Configuration Word (Word 04h)

Bit	Name	Description		
15:12[a]	LED 3 Config	Value 1h 2h 3h 4h 5h 6h 7h 8h 9h	Mode 1 Default Default Default On On On Off Off Off	Mode 2 Default On Off Default On Off Default On Off
11:8[b]	LED 2 Config LED 2 Control[c]	Same as LED 3 Config. LED 2 control field (**82573E/82573V/82573L**)		
7:4[b]	LED 1 Config LED 1 Control[c]	Same as LED 3 Config. LED 1 control field (**82573E/82573V/82573L**)		
3:0[b]	LED 0 Config LED 0 Control[c]	Same as LED 3 Config. LED 0 control field (**82573E/82573V/82573L**)		

a. Reserved for the **82573E/82573V/82573L**. Bits should be set to 0000b.
b. Not applicable to the **82573E/82573V/82573L**.
c. **82573E/82573V/82573L** only.

5.7.4 EEPROM Image Version Word (Word 05h)

Table 5-69. EEPROM Image Version Word (Word 05h)

Bit	Name	Description
15:8	EEPROM Image Major Version	EEPROM image major version number.
7:0	EEPROM Image Minor Version	EEPROM image minor version number.

5.7.5 631xESB/632xESB SerDes Adjustment (Word 06h)

This word is for platform / LOM specific usage. Word 06h is used to store the value for the SERDES amplitude adjustment.

Note: A default value of FFFFh means the word is not used for any purpose.

Table 5-70. SerDes Adjustment (Word 06h)

Bit	Name	Description
15:8	Reserved	Default to 00h if word 06h is used (word value is not FFFFh).
7:4	Reserved	Default to 0h if word 06h is used (word value is not FFFFh).
3:0	SERDES amplitude	Value of the SERDES amplitude that is written into the PHY register if word 06h is used (i.e. word value is not FFFFh).

5.7.6 631xESB/632xESB Software Configuration (Word 07h)

This word is for platform / LOM specific usage. This word is used to enable the PHY class A.

Note: A default value of FFFFh means this word is not used for any purpose.

Table 5-71. Software Configuration (Word 07h)

Bit	Name	Description
15	PHY Class A Enable	If word value is not FFFFh: Bit 15 = 1b. Enable PHY class A. Bit 15 = 0b. Class AB mode (i.e. default PHY class mode).
14:1	Reserved	Reserved for future software configurations. Default to 0b if this word is used for any software configuration (i.e. the word value is not FFFFh).
0	Reserved	Default to 0b if this word is being used for any software configuration (i.e. the word value is not FFFFh).

5.7.7 OEM Configuration (Words 06h:07h)

These words are for OEM configuration usage.

Note: A default value of FFFFh means the word is not used for any purpose.

5.7.8 PBA Number or OEM Version (Words 08h, 09h)

Words 08h and 09h are dual-purpose words: PBA number (used for Intel manufactured adapter cards) and OEM version.

The nine-digit Printed Board Assembly (PBA) number (Table 5-72), used for Intel manufactured adapter cards, is stored in a four-byte field. The dash itself is not stored, neither is the first digit of the 3-digit suffix, as it will always be 0b for the affected products. Note that through the course of hardware ECOs, the suffix field (byte 4) is incremented. The purpose of this information is to allow Customer Support (or any user) to identify the exact revision level of a product. Network driver software should not rely on this field to identify the product or its capabilities.

Note: The PBA number is not related to the Packet Buffer Allocation (PBA) register or to the MSI-X PBA (Pending Bit Array).

Table 5-72. PBA Number (Words 08h, 09h)

Product	PWA Number	Byte 1	Byte 2	Byte 3	Byte 4
Example	123456-003	12	34	56	03

Note: A default value of FFFFh means this word is not used for any purpose.

OEMs can also use these words for the revisioning level of their products. The use of these words is completely up to the OEM. However, if the EEPROM image is created using Intel's internal tools, this is where the contents of the OEM_version keyword is placed. The internal tools writes these values after all other rules are applied. This enables the internal tools to overwrite all values. Writes are denoted as follows:

;0x8 [15:12] - 0x9 - ;OEM Version Major <Set By Definition File: OEM_Version>

;0x8 [11:4] - 0x1 - ;OEM Version Minor <Set By Definition File: OEM_Version>

;0x8 [3:0] - 0x2 - ;OEM ID 2 = OEM <Set By Definition File: OEM_ID>

OEM_Version: Major #. Minor # - The major # is written to the top four bits of word 8h. The minor # is written to the middle eight bits of word 8h.

OEM_ID: OEM ID - The internal tools converts this string to a given value mapping and writes it to the last four bits of word 8h.

Note: Network driver software should not rely on this field to identify the product or its capabilities.

Power Management 6

6.1 Introduction

The PCIe* 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 each of the Ethernet controllers.

Implementation requirements were obtained from the following documents:

- PCI Bus Power Management Interface Specification Revision 1.1
- PCI Express Base Specification Revision 1.0a
- ACPI Specification Revision 2.0
- PCI Express Card Electromechanical Specification Revision 1.0
- Mobile PCIe Communications Specification Revision 0.80KD

Note: Power management can be disabled through bits in the Initialization Control Word, which is loaded from the EEPROM/NVM during power-up reset. Even when disabled, the power management register set is still present. Power management support is required by the PCIe* Specification.

6.1.1 Assumptions

The following assumptions apply to the implementation of power management for the Ethernet controllers:

- The driver sets the filters up prior to the system transitioning the Ethernet controller to the D3 state.
- Before a transition from D0 to the D3 state, the operating system ensures the device driver has been disabled.
- No wake-up capability, except APM wakeup if it is enabled in the EEPROM/NVM, is required after the system puts the Ethernet controller into the D3 state and then returns it to D0.
- If the APMPME bit in the Wake Up Control Register (WUC.APMPME) is 1b, it is permissible to assert GIO_WAKE_N even when PME_En is 0b.

6.1.2 Power Targets[1]

The following table shows the power targets for the Ethernet controller. The numbers apply to the Ethernet controller power and do not include power losses on external voltage regulators.

1. Not applicable to the **631xESB/632xESB** or **82563EB/82564EB**.

State	Conditions	EAS Target	Comments
D0active (1000 Mb/s active)	Maximum load at 1000 Mb/s	1900/1600[a] mW 1400/860[a] mA	
D3cold with Wake	Link at 10 Mb/s	100/130[a] mW 75/65[a] mA	
D3cold without Wake	No ASF, WOL, or PME	20/3[a] mA	

a. **82573E/82573V/82573L** only.

The following table determines power targets per power supply[1].

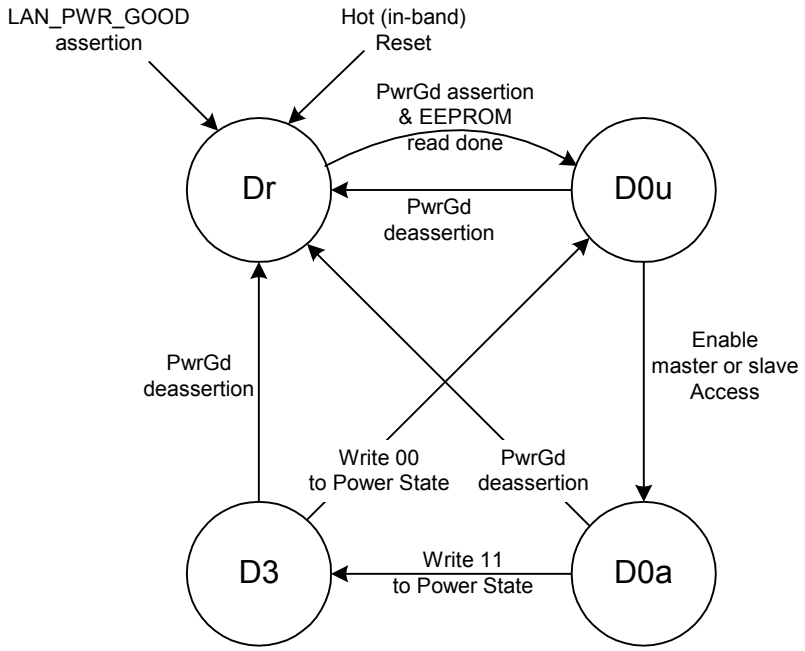
Power State	Units	1.0V Supply	1.8V Supply	Total
D0active (1000 Mb/s active)	mW	800	600	1400
D3cold, Dr with Wake	mA	40	35	75
D3cold without Wake	mA	17	3	20

6.1.3 Introduction to Power States

The Ethernet controller supports D0 and D3 power states defined in the PCI Power Management and PCIe* Specifications. D0 is divided into two sub-states: D0u and D0a. In addition, it supports a Dr state that is entered when the power good signal is de-asserted (including the D3cold state).

1. Not applicable to the **82573E/82573V/82573L**.

Figure 6-1 shows the power states and transitions between them[1].



NOTE: For the **631xESB/632xESB**, substitute LAN_PWR_GOOD and PwrGd with GIO_PWR_GOOD.

Figure 6-1. Power States and Transitions

1. Not applicable to the **82573E/82573V/82573L**.

Figure 6-2 shows the **82573E/82573V/82573L** power states and transitions between them.

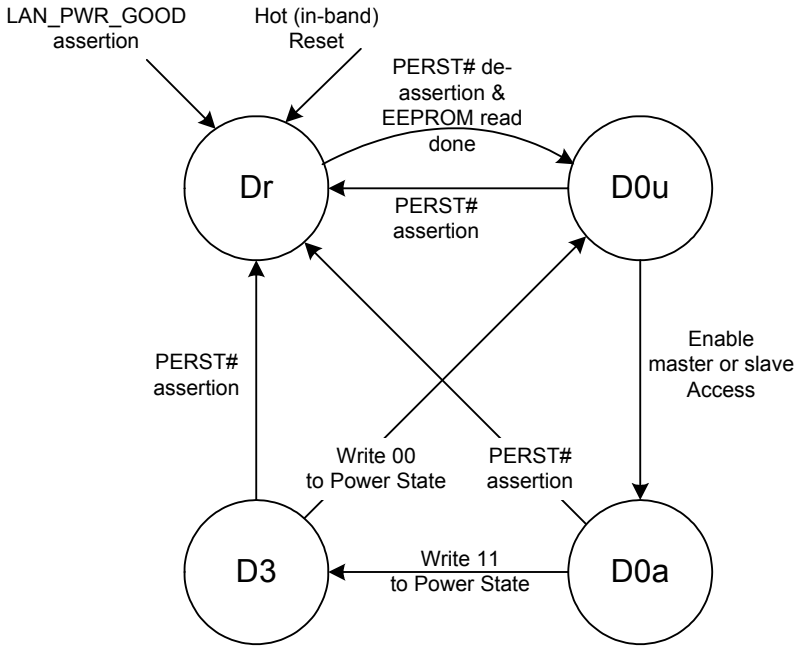


Figure 6-2. Power States and Transitions (82573E/82573V/82573L)

6.1.4 Auxiliary Power Usage[1]

If DisableD3Cold equals 0b and the auxiliary power (AUX_PWR) signal indicates available standby power, the Ethernet controller advertises $D3_{cold}$ wakeup support. The amount of power required for the function (this includes the entire NIC or LOM) is advertised in the Power Management Data Register, which is loaded from the EEPROM/NVM.

If D3cold is supported, the PME_En and PME_Status bits of the Power Management Control/ Status Register (PMCSR) and their shadow bits in the Wake Up Control Register (WUC) are reset only by the power up reset (detection of power rising).

The only effect of setting AUX_PWR to 1b is advertising $D3_{cold}$ wakeup support and changing the reset function of PME_En and PME_Status.

The Ethernet controller tracks the PME_En bit of the PMCSR and the auxiliary power's power management enable bit of the PCIe* Device Control Register to determine the power consumption (and its power state) in the D3cold state (internal Dr state). The actual amount of power differs between form factors.

The PCIE_Aux bit in the EEPROM/NVM determines if the Ethernet controller complies with the auxiliary power regime defined in the PCIe* Specification. If it is set, the Ethernet controller might consume higher auxiliary power according to the following settings:

1. Not applicable to the **631xESB/632xESB** or **82563EB/82564EB**. See Section 6.1.5.

- If the Auxiliary Power PM Enable bit of the PCIe* Device Control Register is set, the Ethernet controller might consume higher power for any purpose (even if PME_En is not set).
- If the Auxiliary Power PM Enable bit of the PCIe* Device Control Register is cleared, higher power consumption is determined by the PCI power management legacy PME_En bit of the PMCSR.

If the PCIE_AUX bit in the EEPROM/NVM is cleared, the Ethernet controller consumes auxiliary power in Dr state independent of the setting of either the PME_En bit or the Auxiliary Power PM Enable bit.

6.1.5 Auxiliary Power Usage (631xESB/632xESB)

The amount of power required for the function (which includes the entire LAN and BMC subsystems) is advertised in the Power Management Data Register, which is loaded from the EEPROM.

The PME_En and PME_Status bits of the Power Management Control/Status Register (PMCSR), as well as their shadow bits in the Wake Up Control Register (WUC) will be reset only by the power up reset (assertion of LAN_PWR_GOOD).

Note: The **631xESB/632xESB** tracks the PME_En bit of the Power Management Control / Status Register (PMCSR) and the Auxiliary (AUX) Power PM Enable bit of the PCIe* Device Control Register to determine the power it might consume (and therefore its power state) in the D3cold state (internal Dr state).

6.1.6 Form Factor Power Limits[1]

The following table summarizes power limitations introduced by some of the different form factors.

	Form Factor		
	PCI Express Add-In Card (x1 Connector)	PCI Express Mobile Communication Slot	PCI Express NEWCARD[a]
Main	3 A @ 3.3V	500 mA @ 3.3V	900 mA @ 3.3v
Auxiliary (aux enabled)	375 mA @ 3.3V	200 mA @ 3.3V[b]	250 mA @3.3v
Auxiliary (aux disabled)	20 mA @ 3.3V		5 mA @ 3.3v

a. "NEWCARD" power refers to total card power. The maximum thermal dissipation directly within the slot from any single wide (software) NEWCARD module is 2.0 Watts.
b. This auxiliary current limit only applies when the primary 3.3V source is not available (for example, the card is in a low power D3 state).

1. Not applicable to the **631xESB/632xESB** or **82563EB/82564EB**.

The Ethernet controller exceeds the allowed power consumption in the following cases:

1. PCIe* add-in card. 1000 Mb/s speed cannot run from auxiliary power, which restricts the Ethernet controller in the Dr state.
2. PCI e* mobile communication slot. 1000 Mb/s operation cannot run from auxiliary power, which might restrict the Ethernet controller speed in any state (including D0).
3. PCIe* NEWCARD. 1000 Mb/s speed cannot run from auxiliary power, which might restrict the Ethernet controller speed in all states or in Dr only, depending on the power delivery scheme.

The Ethernet controller implements two EEPROM/NVM bits to disable gigabit operation in certain cases:

1. The Disable 1000 Mb/s EEPROM/NVM bit disables 1000 Mb/s operation under all conditions.
2. The Disable 1000 Mb/s in non-D0a CSR bit disables 1000 Mb/s operation in non-D0a states. If Disable 1000 Mb/s in non-D0a is set and the Ethernet controller is at gigabit speed on entry to a non-D0a state, device advertisement for 1000 Mb/s is removed and the Ethernet controller must go through Auto-Negotiation. The Disable 1000 Mb/s in non-D0a bit is loaded from the EEPROM/NVM.

The Ethernet controller restarts link Auto-Negotiation each time it transitions from a state where gigabit speed is enabled to a state where gigabit speed is disabled or vice versa. For example, if Disable 1000 Mb/s in non-D0a is set but Disable 1000 Mb/s is clear, the Ethernet controller restarts link Auto-Negotiation on transition from the D0 state to D3 or Dr states.

6.1.7 Unit Power Reduction Measures

This section describes the power reduction techniques employed by Ethernet controller.

6.1.7.1 Manageability Power Management

The Manageability Unit (MMS) is in one of three power management states:

- Full performance — The unit is active with a supplied 62.5 MHz clock.
- Low performance — The unit is active with a supplied 3.125 MHz clock (1:20 of full performance).
- Disabled — A reset is applied to the unit and the clock is gated.

Note: The MMS high performance clock switches to 25 MHz in some cases.

The MMS unit serves several functions and is active based on the requirement of each function:

- ASF/pass-through Manageability — If ASF manageability is enabled (EEPROM/NVM SMB Enable bit set), then the MMS unit is active at all times.
- Intel® AMT — If Intel® AMT manageability is enabled, the MMS unit is active at all times (**82573E** only).
- Serial Port — If the serial port (keyboard/text redirect) function is enabled (EEPROM Serial Enable bit set), then the MMS unit is enabled when the serial port function is in the D0 state (D0u and D0a) and in D3 and Dr states if ACPI PME for this function is enabled (not applicable to the **82573E/82573V/82573L**).

- IDE — If the IDE function is enabled (EEPROM IDE Enable bit set), then the MMS unit is enabled when the IDE function is in D0 state (both D0u and D0a). Not applicable to the **82573E/82573V/82573L**.
- Initialization — The MMS unit is involved in Ethernet controller initialization and is enabled following a configuration reset until the MMS has been configured (not applicable to the **82573E/82573V/82573L**).

When it is enabled, the MMS unit is in low performance mode by default. The unit transitions to full performance in the following cases. It is expected that the MMS unit remains in a high performance mode for a small period of time (power estimates assume low performance mode):

- Receive packet in process in the MMS unit.
- Transmit packet in process in the MMS unit.
- Host accesses to and from the MMS unit.
- Accesses to the EEPROM/NVM, Flash, and CSRs.
- SMB (and FML for the **631xESB/632xESB**) accesses.[1]
- MMS microcontroller requires high performance mode.
- UART, LPC, UHCI, or BT functions active (**631xESB/632xESB** only).

6.1.7.2 PHY Power Management[2]

Link Speed Control

Normal PHY speed negotiation drives to establish a link at the highest possible speed. The Ethernet controller supports an additional mode of operation where the PHY drives to establish a link at a low speed. The link-up process allows a link to come up at the lowest possible speed in cases where power is more important than performance. Different behavior is defined for the D0 state and the other non-D0 states.

Note: The Low Power Link Up (LPLU) feature just described should be disabled (in both D0a and non-D0a states) when the user advertisement is anything other than 10/100/1000 Mb/s. This avoids negotiating through the LPLU procedure a link speed that is not advertised by the user.

The LPLU negotiation process is as follows:

1. The PHY tries to negotiate at 10 Mb/s speed regardless of Auto-Negotiation.
2. If link establishment fails, the PHY tries to negotiate at additional speeds. It enables all speeds to the lowest speed supported by the partner.

For example, if the Ethernet controller advertises 10 Mb/s only and its partner supports 1000 Mb/s only, after the first try fails, the Ethernet controller enables 10/100/1000 Mb/s and tries again. The PHY continues to try and establish a link until it succeeds or until it is instructed otherwise. In the second step (adjusting to partner speed), the PHY also enables parallel detection if it is needed. Automatic MDI/MDI-X resolution is done during the first Auto-Negotiation stage.

The following table lists the link speed as a function of a power management state, link speed control, and gigabit speed enabling:

1. With a 400 KB/s transfer rate, there are less than 62 cycles per byte running at slow clock. This is not enough for processing.
2. Not applicable to the **631xESB/632xESB** or **82563EB/82564EB**.

Power Management State	Low Power Link Up (PHYREG 25.2:1)	1000 Mb/s Disable Bits		PHY Speed Negotiation
		Disable 1000 Mb/s (PHYREG 25.6)	Disable 1000 Mb/s in Non-D0a (PHYREG 25.3)	
D0a	0	0	X	The PHY negotiates to the highest speed advertised (normal operation).
		1		The PHY negotiates to highest speed advertised except 1000 Mb/s.
	1	0	X	The PHY goes through the LPLU procedure, starting with the advertised values.
		1		The PHY goes through the LPLU procedure, starting with advertised values but does not advertise 1000 Mb/s.
Non-D0a	0	0	0	The PHY negotiates to highest speed advertised.
		0	1	The PHY negotiates to highest speed advertised except 1000 Mb/s.
		1	X	
	1	0	0	The PHY goes through the LPLU procedure, starting at 10 Mb/s.
		0	1	The PHY goes through LPLU procedure, starting at 10 Mb/s but does not advertise 1000 Mb/s.

Table 6-1 lists the link speed as a function of power management state, link speed control, and gigabit speed enabling.

D0a State

A power managed link speed control lowers link speed (and power) when the highest link performance is not required. When it is enabled to this D0 Low Power Link Up mode, any link negotiation tries to establish a low link speed, starting with an initial advertisement defined by software.

The D0 LPLU configuration bit enables D0 Low Power Link Up. Before enabling the feature, software must advertise one of the following speed combinations: 10 Mb/s only, 10/100 Mb/s, or 10/100/1000 Mb/s.

Non-D0a State

The PHY can negotiate to a low speed while in a non-D0a state (Dr, D0u, or D3). This applies only when the link is required by SMB manageability, APM wake up, or a power management event. Otherwise, the PHY is disabled during the non-D0a state.

The EEPROM/NVM LPLU bit enables reduction in link speed:

- On power-up entry to Dr state, the PHY advertises support for 10 Mb/s only and goes through the link up process.

- On any entry to a non-D0a state (Dr, D0u, or D3), the PHY advertises support for 10 Mb/s only and goes through the link up process described as follows.
- While in a non-D0 state, if auto-negotiation is required, the PHY advertises support for 10 Mb/s only and goes through the link up process.

The EEPROM/NVM LPLU bit is loaded into the LPLU configuration bit. Software can set or clear this bit at any time. From that point on, the Ethernet controller acts according to the latest value of the LPLU bit.

Link-Disconnect Smart Power Down (SPD) State[1]

The link disconnect state applies to all power management states (Dr, D0u, D0a, and D3). The link can resume and or disconnect in any of these states. This section defines the minimal power management capabilities in the link-disconnect state. In some cases, the PHY enters a deeper power management state than defined in this section. For example, if there is no requirement to maintain the link, the PHY enters a deeper power management state in the non-D0 states.

PHY Smart Power Down (SPD) is a PHY mechanism to reduce PHY power consumption when the link is lost. The PHY enters this state when it detects a lost link state and the PHY SPD bit is set.

Note: The PHY does not enter the SPD state unless Auto-Negotiation is enabled.

While in the SPD state, the PHY powers down all circuits not required to detect link activity. The PHY must still be able to detect link pulses (including parallel detection) and wake up to engage in link negotiation. The PHY does not send link pulses (NLP) in the SPD state.

While the PHY is in the link-disconnect state, it must allow software access to its registers.

Link Energy Detect

The Link Energy Detect MDIO bit is set each time energy is detected on the link. This includes the period of time during link negotiation and when link is established. This bit should be valid immediately after a reset of the device or the PHY.

6.1.7.3 PHY Power Management (631xESB/632xESB and 82563EB/82564EB)

The power state of the MAC is communicated to the PHY through in-band GLCI protocol. The PHY negotiates the link speed and status, based on the following table:

1. Not applicable to the **631xESB/632xESB**.

Power Management State	Low Power Link Up (PHYREG 193.20:3)	1000 Mb/s Disable Bits		PHY Speed Negotiation
		Disable 1000 Mb/s (PHYREG 193.20.5)	Disable 1000 Mb/s in Non-D0a (PHYREG 193.20.4)	
D0a	0	0	X	The PHY negotiates to the highest speed advertised (normal operation).
		1		The PHY negotiates to highest speed advertised except 1000 Mb/s.
	1	0	X	The PHY goes through the LPLU procedure, starting with the advertised values.
		1		The PHY goes through the LPLU procedure, starting with advertised values but does not advertise 1000 Mb/s.
Non-D0a	0	0	0	The PHY negotiates to highest speed advertised.
		0	1	The PHY negotiates to highest speed advertised except 1000 Mb/s.
		1	X	
	1	0	0	The PHY goes through the LPLU procedure, starting at 10 Mb/s.
		0	1	The PHY goes through LPLU procedure, starting at 10 Mb/s but does not advertise 1000 Mb/s.

Auto-negotiation restarts any time one of the following occurs:

- When the state of 1000disable changes. For example, if 1000 is disabled on D3 or Dr entry (but not in D0a), the PHY will Auto-Negotiate on entry.
- When LPLU (Low Power Link Up) changes state with a change in PM state. For example, on transition from D0a w/o LPLU to D3 with LPLU. Or, on transition from D3 w LPLU to D0 w/o LPLU.
- On a transition from D0a state to a non-D0a state, or from a non-D0a state to D0a state, and LPLU is set.

Note: Enabling either LPLU or D0LPLU enables the PHY to negotiate to a speed higher than programmed into the Status and Extended Status registers. Software should take care to ensure that the link speed doesn't end up at a speed the user hasn't enabled.

For example, if a programmer hasn't enabled 100 Mb/s operation then LPLU and D0LPLU wouldn't be usable. If a programmer hasn't enabled 1000 Mb/s then LPLU or D0LPLU should only be used in conjunction with an1000_dis which takes priority.

6.1.7.4 GLCI Power Management (631xESB/632xESB and 82563EB/82564EB)

The GLCI interface has an electrical-idle mechanism, in order to reduce power consumption when possible.

The **82563EB/82564EB** enters the electrical-idle state after recognizing cable disconnected or link down state (sent in-band) or during D3 power-saving state.

Following is the power saving sequence for the **82563EB/82564EB**:

- After power-up, the GLCI interface is active to allow PHY initialization.
- After PHY initialization, the **82563EB/82564EB** might power-down the PHY and enter D3 state.
- The GLCI RX SerDes traces the MAC-PHY interface, waking-up the system under data receiving event.

The following factors trigger a GLCI electrical idle:

- Link down
- PHY power down

The following factors trigger exit from electrical idle:

- All of the above conditions are not true
- MDIO access

6.1.7.5 PHY, SerDes, and Ethernet Controller Power Down

PHY Power-Down State

The PHY enters a power-down state when none of its clients are enabled. In this case, a link is not required to be maintained. The following conditions must be met for the PHY to enter this power-down state:

- PHY power down is enabled through the EEPROM/NVM PHY Power Down Enable bit.
- The LAN function associated with this PHY is in a non-D0 state.
- Both the Serial Port and IDE functions are in a non-D0 state.
- APM Wake on LAN* (WOL) is inactive.
- Intel® AMT[1]/ASF/Pass-through manageability is disabled.
- ACPI PME is disabled for both the LAN and serial port functions. (In some Dr cases, ACPI power management is irrelevant.)

Note: The **631xESB/632xESB** can also send a power-down command to a GLCI-compliant PHY through a GLCI in-band message.

The PHY also enters a power-down state in the following cases (not applicable to the **82573E/82573V/82573L**):

- When the Ethernet controller is configured to SerDes only operation (not applicable to the **631xESB/632xESB**).
- When the LAN0_DIS_N (or LAN1_DIS_N) signal for this LAN function indicates that the relevant function should be disabled (not applicable to the **631xESB/632xESB**).

1. **82573E** only.

SerDes Power-Down State

The Serializer/Deserializer (SerDes) enters a power-down state when none of its clients are enabled. This case does not require a link to be maintained. The following conditions must be met for the SerDes to enter this power-down state:

- SerDes power down is enabled through the EEPROM SerDes Low Power Enable bit.
- The LAN function associated with this SerDes is in a non-D0 state.
- Both the serial port and IDE functions are in a non-D0 state.
- APM WOL is inactive.
- ASF/Pass-through manageability is disabled.
- ACPI PME is disabled for both the LAN and serial port functions. (In some Dr cases, ACPI power management is irrelevant.)

The SerDes also enters a power-down state in the following cases:

- When the Ethernet controller is configured to PHY only operation.
- When the LAN0_DIS_N (or LAN1_DIS_N) signal for this LAN function indicates that the relevant function should be disabled (not applicable to the **631xESB/632xESB**).

Ethernet Controller Power-Down State[1]

The Ethernet controller enters a global power-down state if all of the following conditions are met:

- The device Dr power down enable EEPROM/NVM bit was set (hardware default value is disabled).
- The EE_VR_PWR_DWN bit loaded from the EEPROM/NVM is cleared.
- The Ethernet controller is in Dr state.
- The PHY is in power-down state for any of the reasons mentioned in the PHY Power-Down State section above.
- The SerDes is in power-down state for any of the reasons mentioned in the SerDes Power-Down State section previously described.
- The Ethernet controller also enters a power-down state when the DEV_OFF_N signal is active.

1. **82571EB** and **82572EI** only.

Ethernet Controller Shutdown State[1]

The Ethernet controller enters a device shutdown mode on transition to the D3cold state when it does not need to maintain any functionality. The conditions to enter this state are:

- The EE_VR_PWR_DWN bit from the EEPROM is set.
- The conditions for the Ethernet controller power-down state are met. These conditions are listed in the Device Power-Down State section previously described.

Entry into Ethernet controller shutdown typically occurs on the assertion of GIO Power Good. If the above conditions are met, the Ethernet controller enters shutdown mode. It is also possible to enter Ethernet controller shutdown mode by reading the EEPROM while already in Dr state (GIO Power Good already asserted). The usage model for this case is on system power up, assuming that ASF, APM, or ACPI are not required. After the Ethernet controller enters Dr state on power-up, the EEPROM is read. If the EEPROM contents determine that the conditions to enter Ethernet controller shutdown are met, the Ethernet controller enters Ethernet controller shutdown mode, assuming that GIO Power Good has not been asserted.

Output buffers are tri-stated while the device is in Ethernet controller shutdown mode.

Exit from the Ethernet controller shutdown mode is through the de-assertion of GIO Power Good. The following table lists the stages during a device shutdown exit and compares it to system power up.

Table 6-1. Exit from Device Shutdown in D3 Cold State

Power Up			Exit from VR Shutdown		
Step	Time (delta)	Time (total)	Step	Time (delta)	Time (total)
			GIO Power Good		0 ms
Power ramp (all)		0 ms	Power ramp (1.8V/1.0V)	5 ms	5 ms
Xosc stable	10 ms	10 ms	Xosc stable	10 ms	15 ms
POR / LAN Power Good	50 ms	50 ms	POR / Internal LAN Power Good	0 ms	15 ms
EEPROM Read	20 ms	110 ms	EEPROM Read for GIO	2 ms	17 ms
GIO Power Good	N/A	100 ms			
GIO PLL lock	50 µs	100.05 ms	GIO PLL lock	50 us	15.05 ms
Internal GIO Power Good	50 µs	100.05 ms	Internal GIO Power Good	50 us	15.05 ms
EEPROM Read for GIO	2 ms	102.05 ms			
PCIe link training	N/A	< 20 ms	PCIe* link training	N/A	< 20 ms

1. **82571EB** and **82572EI** only.

6.1.7.6 Common Resources[1]

The Ethernet controller Common Clock Module (CCM) unit provides a clock to both LAN functions. The CMM unit is power managed based on the state of both functions. As an exception, the CCM unit does not stop the internal PLL when the link is in link-disconnect mode (even if the two links are in link-disconnect state).

If shared resources are required for both functions, these resources do not always power down; the resource only powers down when both ports are powered down.

6.1.8 Power States

6.1.8.1 Dr State

Transition to Dr state is initiated on three occasions:

- On system power up. The Dr state starts with the assertion of the internal power detection circuit (LAN_POWER_GOOD) and ends with the assertion of the internal power good signal, indicating that the system asserted its PE_RST# (GIO_PWR_GOOD for the **631xESB/632xESB**) signal (de-asserted its PERST# signal for the **82573E/82573V/82573L**).
- On transition from a D0a state. During operation, the system may de-assert PE_RST# (GIO_PWR_GOOD for the **631xESB/632xESB**) at any time (assert PERST# for the **82573E/82573V/82573L**). In an ACPI system, a system transition to the G2/S5 state causes a transition from D0a to Dr.
- On transition from a D3 state. The system transitions the Ethernet controller into the Dr state by removing the PCIe* power good signal (GIO_PWR_GOOD for the **631xESB/632xESB**) (asserting PERST# for the **82573E/82573V/82573L**).

Any wake-up filter settings that were enabled before entering this reset state are maintained.

The power good signal (GIO_PWR_GOOD for the **631xESB/632xESB**) might remain de-asserted by the system for an arbitrary time. The assertion (rising edge) of the power good signal (GIO_PWR_GOOD for the **631xESB/632xESB**) causes a transition to D0u state (not applicable to the **82573E/82573V/82573L**).

Entry to Dr State

Dr entry on platform power up starts with the assertion of the internal power detection circuit (LAN_POWER_GOOD). The EEPROM/NVM is read and determines the Ethernet controller configuration. If the APM Enable bit in the EEPROM/NVM Initialization Control Word 2 is set, then APM wake up is enabled. The MAC and PHY state is determined by the manageability state and APM wake. To reduce power consumption, if ASF manageability or APM wake is enabled, the PHY auto-negotiates to a lower link speed on Dr entry. The PCIe* link is not enabled in Dr state following system power up because the power good signal (GIO_PWR_GOOD for the **631xESB/632xESB**) is de-asserted (because the PERST# signal is asserted for the **82573E/82573V/82573L**).

1. **82571EB** and **82572EI** only.

Entry to Dr state from D0a state is through the de-assertion of the power good signal (GIO_PWR_GOOD for the **631xESB/632xESB**) (assertion of the PERST# signal for the **82573E/ 82573V/82573L**). An ACPI transition to the G2/S5 state is reflected in a device transition from D0a to Dr. The transition can be orderly (programmer selected the shut down option), where the software device driver might have the opportunity to intervene. Otherwise, it might be an emergency transition (for example, power button override), where the software device driver is not notified.

To reduce power consumption if ASF manageability, APM wake or PCI PM PME[1] is enabled, the PHY auto-negotiates to a lower link speed on D0a to Dr transition.

Transition from the D3 state to the Dr state is accomplished by de-asserting the power good signal (GIO_PWR_GOOD for the **631xESB/632xESB**) (asserting the PERST# signal for the **82573E/ 82573V/82573L**). Before the transition, the system initiates a transition of the PCIe* link from the L1 state to either the L2 or L3 state, assuming all functions were already in D3 state. The link enters the L2 state if PCI PM PME is enabled.

6.1.8.2 D0 Uninitialized State

The D0u state is a low-power state used after the power good signal (GIO_PWR_GOOD for the **631xESB/632xESB**) is asserted following power up on hot reset (in-band reset through PCIe* physical layer message) or on D3 exit.

When entering D0u, the Ethernet controller disables wake ups and asserts a reset to the PHY while the EEPROM/NVM is being read. If the APM mode bit in the EEPROM/NVM Initialization Control Word 2 is set, then APM wake up is enabled.

D0u is reached from either the Dr state, upon assertion of an internal power good signal (GIO_PWR_GOOD for the **631xESB/632xESB**), or the D3hot state. The transition from the D3hot state to the D0u state can occur by configuration software writing a value of 00b to the power state field of the PCI PM registers).

Assertion of internal power good (GIO_PWR_GOOD for the **631xESB/632xESB**) means that the entire state of the Ethernet controller is cleared except for the sticky bits. The state is loaded from the EEPROM/NVM. Afterwards, the PCIe* link is established. When this completes, the configuration software can access the Ethernet controller.

On a transition from D3 to D0u, the Ethernet controller PCI Configuration space is not reset. According to the PCI Power Management Specification, Revision 1.1, Section 5.4, software “will need to perform a full re-initialization of the function including its PCI Configuration Space.”

6.1.8.3 D0 Active State

When memory space is enabled, all internal clocks are activated and the Ethernet controller enters an active state. It can transmit and receive packets if it is properly configured by the software device driver. The PHY is enabled or re-enabled by the software device driver to operate or auto-negotiate to full line speed and power if it is not already operating at full capability. Any APM wake up previously active remains active. The software device driver can deactivate APM wake up by writing to the Wake Up Control Register (WUC). It can also activate other wake-up filters by writing to the Wake Up Filter Control Register (WUFC).

1. ACPI 2.0 specifies that “OSPM will not disable wake events before setting the SLP_EN bit when entering the S5 sleeping state. This provides support for remote management initiatives by enabling Remote Power On (RPO) capability. This is a change from ACPI 1.0 behavior.

D0a is entered from the D0u state by writing a 1b to the Memory Access Enable or the I/O Access Enable bit of the PCI Command Register. The DMA, MAC, and PHY of the appropriate LAN function are enabled. Manageability is also enabled if any of the following conditions apply:

- Intel® AMT[1]/ASF/Pass-through manageability is enabled.
- Serial port is enabled.
- IDE function is enabled.
- UHCI function is enabled (**631xESB/632xESB** only).
- BT function is enabled (**631xESB/632xESB** only).

6.1.8.4 D3 State

The D3 state referred to for this section is PCI PM D3hot.

When the system writes 11b to the Power State field of the PMCSR, the Ethernet controller transitions to D3. Any wake-up filter settings enabled before entering this reset state are maintained. Upon transition to the D3 state, the Ethernet controller clears the Memory Access Enable and I/O Access Enable bits of the PCI Command Register, which disables memory access decode. In D3, the Ethernet controller only responds to PCI configuration accesses and does not generate master cycles.

A D3 state is followed by either a D0u state (in preparation for a D0a state) or by a transition to Dr state (PCI PM D3cold state). To transition back to D0u, the system writes 00b to the Power State field of the PMCSR. Transition to the Dr state is through power good (GIO_PWR_GOOD for the **631xESB/632xESB**) de-assertion.

Entry to D3 State

Transition to D3 state is through a configuration write to the Power State field of the PCI PM registers.

Prior to transition from D0 to the D3 state, the software device driver disables scheduling of further tasks to the Ethernet controller. It masks all interrupts and does not write to the transmit descriptor tail register or to the receive descriptor tail register and operates the master disable algorithm. If wake up capability is needed, the software device driver should set up the appropriate wake up registers and the system should write a 1b to the PME_En bit of the PMCSR or to the Auxiliary Power PM Enable bit of the PCIe* Device Control Register before the transition to D3.

As a response to being programmed into the D3 state, the Ethernet controller brings its PCI e* link into the L1 link state. As part of the transition into the L1 state, the Ethernet controller suspends scheduling of new TLPs and waits for the completion of all previous TLPs it has sent. The Ethernet controller clears the Memory Access Enable and I/O Access Enable bits of the PCI Command Register, which disables memory access decode. Any receive packets that have not been transferred into system memory is kept in the Ethernet controller and discarded later on D3 exit. Any transmit packets that have not been sent can still be transmitted, assuming the Ethernet link is valid.

To reduce power consumption, if ASF manageability, APM wake or PCI PM PME is enabled, the PHY will auto-negotiate to a lower link speed on D3 entry.

1. **82573E** only.

Master Disable

System software can disable master accesses on the PCIe* link by either clearing the PCI Bus Master bit or by bringing the function into a D3 state. From this point on, the Ethernet controller must not issue master accesses for this function. Due to the full duplex nature of PCIe* and the pipelined design in the Ethernet controller, multiple requests from several functions might be pending when the master disable request arrives. The protocol described in this section ensures that a function does not issue master requests to the PCIe* link after its master enable bit is cleared or after entry to D3 state.

Two configuration bits are provided for the handshake between the Ethernet controller function and its driver:

- GIO Master Disable bit in the Device Control Register (CTRL). When the GIO Master Disable bit is set, the Ethernet controller blocks new master requests, including manageability requests, by this function. The Ethernet controller then proceeds to issue any pending requests by this function. This bit is cleared on master reset (LAN Power Good all the way to Software Reset) to allow master accesses.
- GIO Master Enable Status bits in the Device Status Register. These bits are cleared by the Ethernet controller when the GIO Master Disable bit is set and no master requests are pending by the relevant function. Otherwise, these bits are set. This indicates that no master requests is issued by this function as long as the GIO Master Disable bit is set.

Note: The software device driver sets the GIO Master Disable bit when it is notified of a pending master disable or D3 entry. The Ethernet controller blocks new requests and proceeds to issue any pending requests by this function. Afterwards, the software device driver polls the GIO Master Enable Status bit. When the bit is cleared, it is guaranteed that no requests are pending from this function. The software device driver can time out if the GIO Master Enable Status bit is not cleared within a given time period.

Note: The GIO Master Disable bit must be cleared to enable a master request to the PCIe* link. This can be done either through reset or by the software device driver.

6.1.8.5 Link-Disconnect

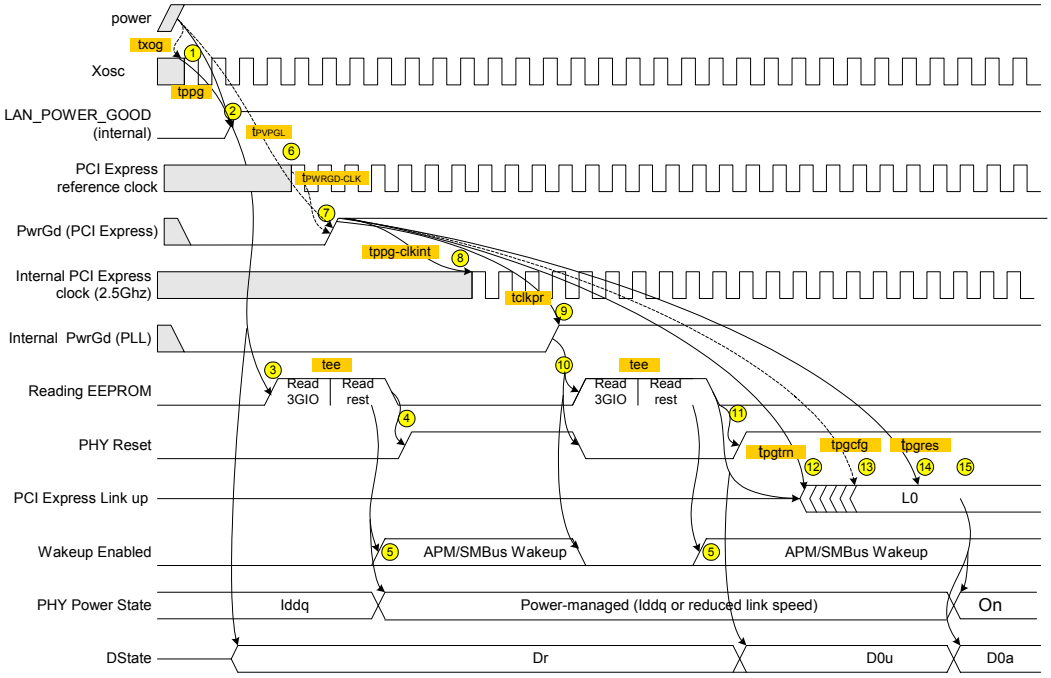
In any of D0u, D0a, D3, or Dr states, the Ethernet controller enters a link-disconnect state if it detects a link-disconnect condition on the Ethernet link. The link-disconnect state is invisible to software (other than the Link Energy Detect bit state). In particular, when the Ethernet controller is in the D0 state, software can access any of the Ethernet controller registers as in a link-connect state.

6.1.9 Power-State Transitions Timing

The following sections give detailed timing for the state transitions. The timing diagrams are not to scale, and the dotted connecting lines represent the Ethernet controller requirements, and the solid connecting lines, Ethernet controller guarantees. The clocks edges are shown to indicate running clocks only. They are not used to indicate the actual number of cycles for any operation.

6.1.9.1 Power Up[1]

Note: For the **631xESB/632xESB**, substitute GIO_PWR_GOOD for PwrGd.

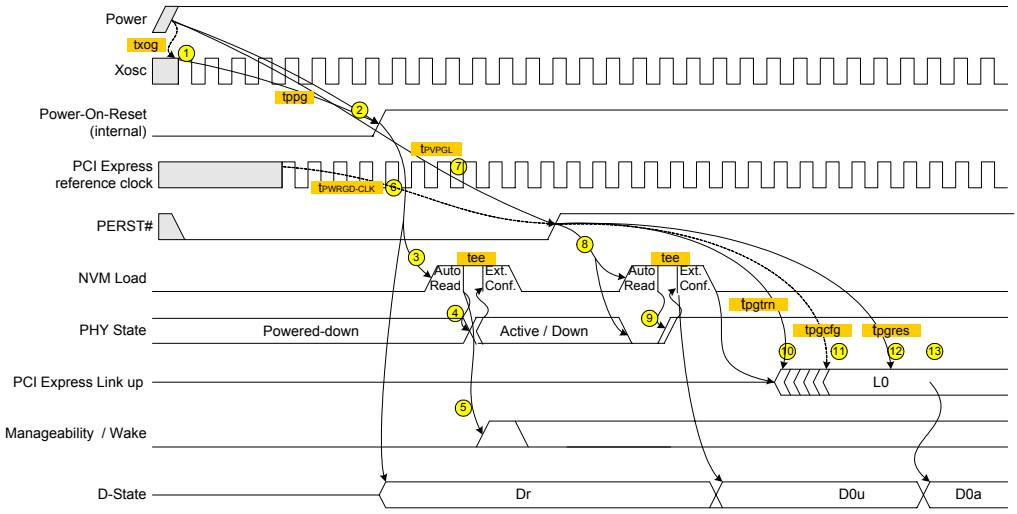


Notes	
1	Xosc is stable t_{xog} after the power is stable.
2	LAN_PWR_GOOD is asserted after all power supplies are good and t_{ppg}, after Xosc, is stable.
3	An EEPROM read starts on the rising edge of LAN_PWR_GOOD.
4	After reading the EEPROM, the PHY reset is de-asserted.
5	APM wake-up mode may be enabled based on what is read from the EEPROM.
6	The PCIe* reference clock is valid $t_{PWRGD-CLK}$ before the assertion of PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**). Per the PCIe* Specification.
7	PE_RST# (GIO_PWR_GOOD for the **631xESB/632xESB**) is asserted t_{PVPGL} after power is stable.
8	The Internal PCIe* clock is valid and stable $t_{ppg-clkint}$ from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
9	The PCIe* internal PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) signal is asserted t_{clkpr} after the external PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) signal.
10	Assertion of the internal PCIe* PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) causes the EEPROM to be re-read, asserts PHY reset, and disables wake up.
11	After reading the EEPROM, PHY reset is de-asserted.

1. Not applicable to the **82573E/82573V/82573L**. See Section 6.1.9.2.

Notes	
12	Link training starts after t_{pgtrn} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
13	A first PCIe* configuration access might arrive after t_{pgcfg} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
14	A first PCIe* configuration response can be sent after t_{pgres} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
15	Writing a 1b to the Memory Access Enable bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state.

6.1.9.2 Power Up (82573E/82573V/82573L)

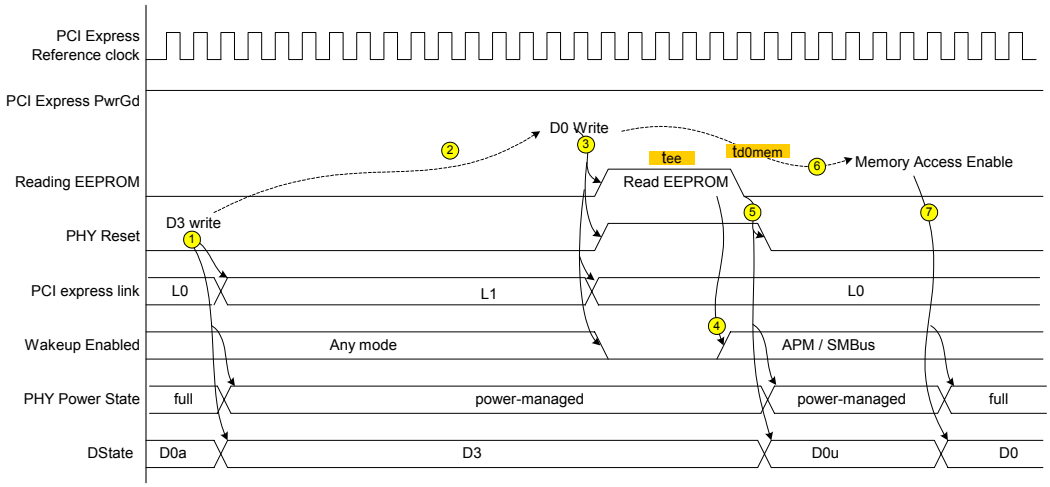


Notes	
1	Xosc is stable t_{xog} after the power is stable.
2	Internal Reset is released after all power supplies are good and t_{ppg}, after Xosc, is stable.
3	An NVM read starts on the rising edge of the Internal Reset or LAN_PWR_GOOD.
4	After reading the NVM, the PHY can exit power down mode.
5	APM wake-up and/or manageability can be enabled based on NVM contents.
6	The PCIe* reference clock is valid $t_{PWRGD-CLK}$ before the de-assertion of PERST# (as per the PCIe* Specification).
7	PERST# is de-asserted t_{PVPGL} after power is stable.
8	De-asserting PERST# causes the NVM to be re-read, asserts PHY power-down, and disables Wake Up.
9	After reading the NVM, the PHY exits power-down mode.
10	Link training starts after t_{pgtrn} from PERST# de-assertion.

Notes	
11	A first PCIe* configuration access might arrive after t_{pgcfg} from PERST# de-assertion.
12	A first PCIe* configuration response can be sent after t_{pgres} from PERST# de-assertion.
13	Writing a 1b to the Memory Access Enable bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state.

6.1.9.3 Transition from D0a to D3 and Back without PWRGD[1]

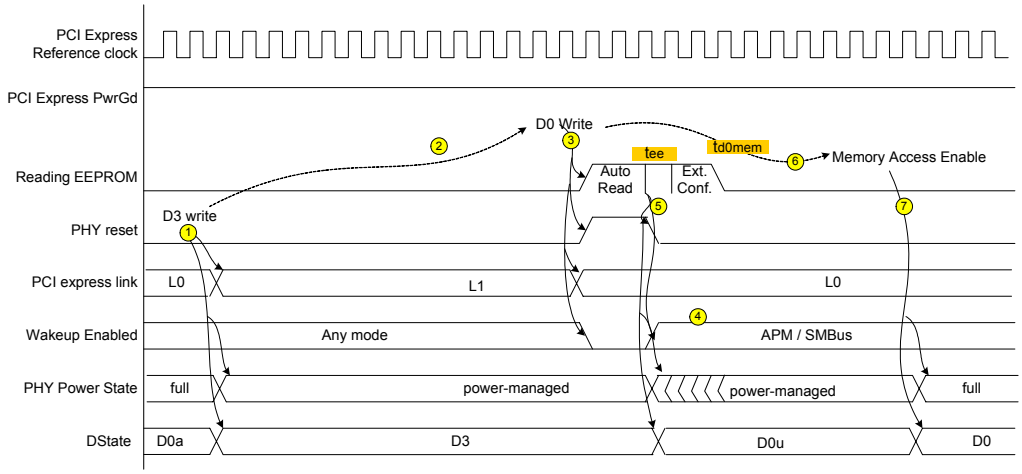
Note: For the **631xESB/632xESB**, substitute GIO_PWR_GOOD for PwrGd.



Notes	
1	Writing 11b to the Power State field of the Power Management Control/Status Register (PMCSR) transitions the Ethernet controller to D3.
2	The system keeps the Ethernet controller in D3 for an arbitrary amount of time.
3	To exit D3, the system writes 00b to the Power State field of the Power Management Control/Status Register (PMCSR).
4	APM wake up or SMB mode can be enabled based on the EEPROM contents.
5	After reading the EEPROM, the reset to the PHY is de-asserted. The PHY operates at a reduced speed if APM wake up or SMB is enabled. Otherwise, it is powered down.
6	The system can delay an arbitrary time before enabling memory access.
7	Writing a 1b to the Memory Access Enable bit or to the I/O Access Enable bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 and returns the PHY to full power and full speed operation.

1. Not applicable to the **82573E/82573V/82573L**. See Section 6.1.9.4.

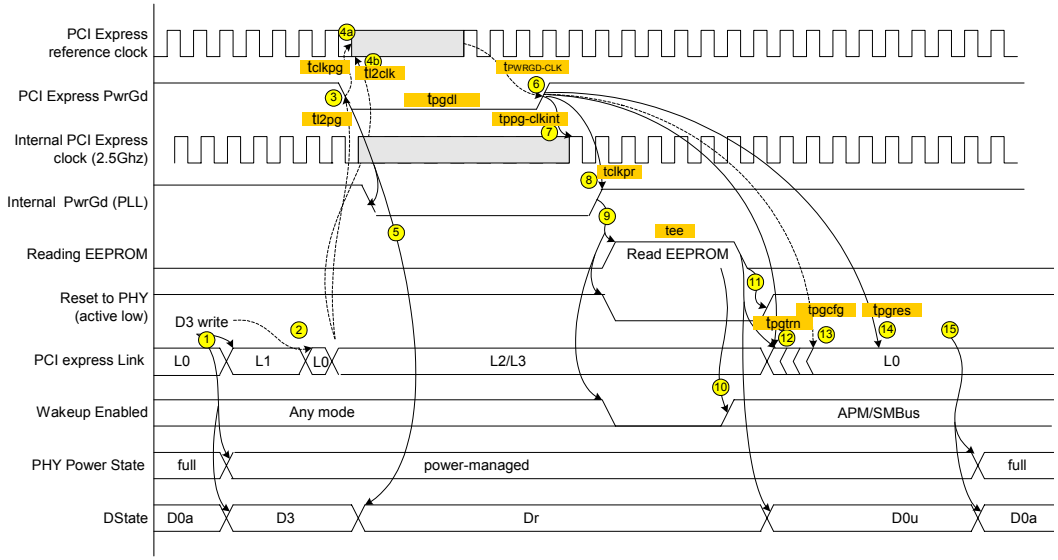
6.1.9.4 Transition from D0a to D3 and Back without PERST# (82573E/82573V/82573L)



Notes	
1	Writing 11b to the Power State field of the Power Management Control/Status Register (PMCSR) transitions the Ethernet controller to D3.
2	The system keeps the Ethernet controller in D3 for an arbitrary amount of time.
3	To exit D3, the system writes 00b to the Power State field of the Power Management Control/Status Register (PMCSR).
4	APM wake up or SMB mode can be enabled based on the NVM contents.
5	After reading the NVM, the reset to the PHY is de-asserted. The PHY operates at a reduced speed if APM wake up or SMB is enabled. Otherwise, it is powered down.
6	The system can delay an arbitrary time before enabling memory access.
7	Writing a 1b to the Memory Access Enable bit or to the I/O Access Enable bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 and returns the PHY to full power and full speed operation.

6.1.9.5 Transition from D0a to D3 and Back with PWRGD[1]

Note: For the **631xESB/632xESB**, substitute GIO_PWR_GOOD for Internal PwrGd (PLL).

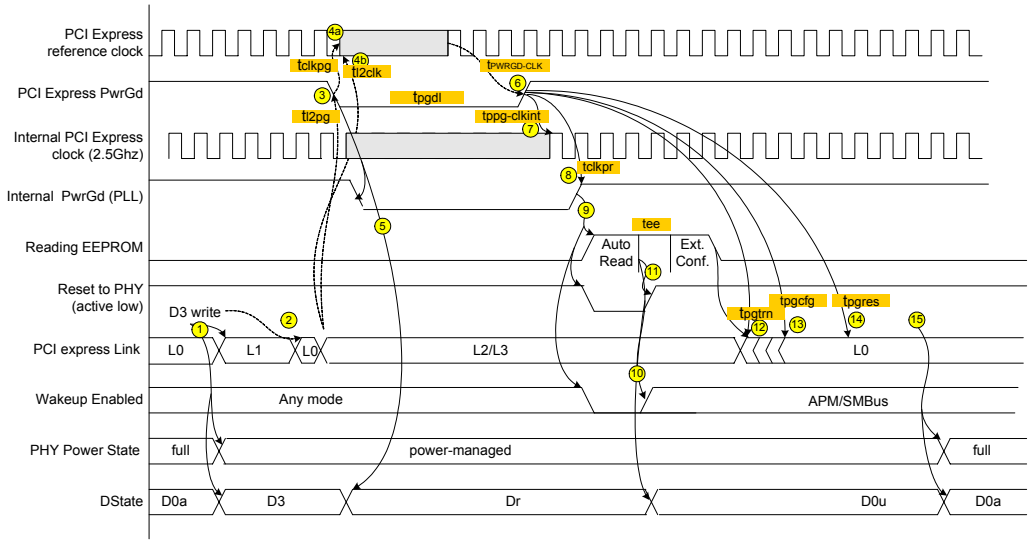


Notes	
1	Writing 11b to the Power State field of the Power Management Control/Status Register (PMCSR) transitions the Ethernet controller to D3. The PCIe* link transitions to L1 state.
2	The system can delay an arbitrary amount of time between setting the D3 mode and transitioning the link to an L2 or L3 state.
3	After a link transition, PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) is de-asserted.
4	The system must de-assert PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) before stopping the PCI e* reference clock. It must also wait t_{l2clk} after link transition to L2 or L3 before stopping the reference clock.
5	When PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) is de-asserted, the Ethernet controller transitions to Dr state.
6	The system starts the PCIe* reference clock $t_{PWRGD-CLK}$ before asserting PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**).
7	The internal PCIe* clock is valid and stable $t_{ppg-clkint}$ from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
8	The PCIe* internal PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) signal is asserted t_{clkpr} after the external PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) signal.
9	Assertion of the internal PCIe* PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) causes the EEPROM to be re-read, asserts a PHY reset, and disables wake up.
10	APM wake-up mode might be enabled based on the EEPROM contents.
11	After reading the EEPROM, the PHY reset is de-asserted.

1. Not applicable to the **82573E/82573V/82573L**. See Section 6.1.9.6.

Notes	
12	Link training starts after t_{pgtrn} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
13	A first PCIe* configuration access might arrive after t_{pgcfg} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
14	A first PCIe* configuration response can be sent after t_{pgres} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
15	Writing a 1b to the Memory Access Enable bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state.

6.1.9.6 Transition from D0a to D3 and Back with PERST# (82573E/82573V/82573L)

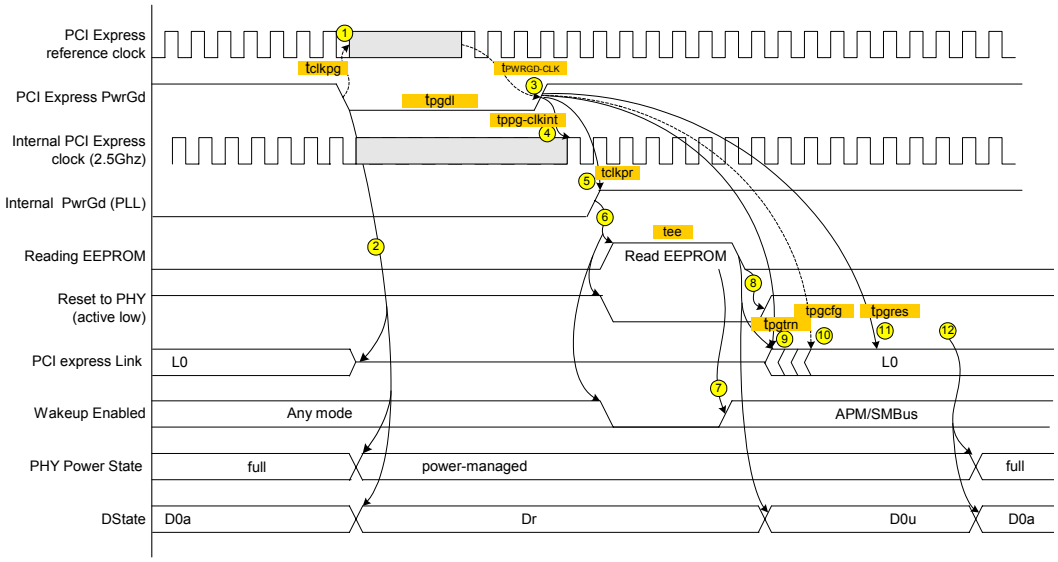


Notes	
1	Writing 11b to the Power State field of the Power Management Control/Status Register (PMCSR) transitions the Ethernet controller to D3. The PCIe* link transitions to L1 state.
2	The system can delay an arbitrary amount of time between setting the D3 mode and transitioning the link to an L2 or L3 state.
3	After a link transition, PERST# is asserted.
4	The system must assert PERST# before stopping the PCI e* reference clock. It must also wait t_{l2clk} after link transition to L2 or L3 before stopping the reference clock.
5	When PERST# is asserted, the Ethernet controller transitions to Dr state.
6	The system starts the PCIe* reference clock $t_{PWRGD-CLK}$ before de-asserting PERST#.
7	The internal PCIe* clock is valid and stable $t_{ppg-clkint}$ from PERST# de-assertion.
8	The PCIe* internal PWRGD signal is asserted t_{clkpr} after the external PERST# signal.
9	Assertion of the internal PCIe* PWRGD causes the NVM to be re-read, asserts a PHY reset, and disables wake up.
10	APM wake-up mode might be enabled based on the NVM contents.

Notes	
11	After reading the NVM, the PHY reset is de-asserted.
12	Link training starts after t_{pgtrn} from PERST# de-assertion.
13	A first PCIe* configuration access might arrive after t_{pgcfg} from PERST# de-assertion.
14	A first PCIe* configuration response can be sent after t_{pgres} from PERST# de-assertion.
15	Writing a 1b to the Memory Access Enable bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state.

6.1.9.7 D0a to Dr and Back without Transition to D3[1]

Note: For the **631xESB/632xESB**, substitute GIO_PWR_GOOD for Internal PwrGd (PLL).



Notes	
1	The system must de-assert PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) before stopping the PCIe* reference clock. It must also wait t_{l2clk} after link transition to L2 or L3 before stopping the reference clock.
2	When PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) is de-asserted, the Ethernet controller transitions to Dr state and the PCIe* link transitions to an electrical idle.
3	The system starts the PCIe* reference clock $t_{PWRGD-CLK}$ before asserting PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**).
4	The internal PCIe* clock is valid and stable $t_{ppg-clkint}$ from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
5	The PCIe* internal PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) signal is asserted t_{clkpr} after the external PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) signal.
6	Assertion of the internal PCIe* PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) causes the EEPROM to be re-read, asserts a PHY reset, and disables wake up.

1. Not applicable to the **82573E/82573V/82573L**.

Notes	
7	APM wake-up mode may be enabled based on the EEPROM contents.
8	After reading the EEPROM, PHY reset is de-asserted.
9	Link training starts after t_{pgtrn} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
10	A first PCIe* configuration access might arrive after t_{pgcfg} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
11	A first PCIe* configuration response can be sent after t_{pgres} from PWRGD (GIO_PWR_GOOD for the **631xESB/632xESB**) assertion.
12	Writing a 1b to the Memory Access Enable bit in the PCI Command Register transitions the Ethernet controller from D0u to D0 state.

6.1.9.8 Timing Requirements

The Ethernet controller requires the following start up and power state transitions.

Parameter	Description	Min	Max	Notes
t_{xog}	Xosc stable from power stable.		10 ms	
$t_{PWRGD-CLK}$	PCIe* clock valid to PCIe* power good.	100 µs	-	As per PCIe* Specification.
t_{PVPGL}	Power rails stable to PCIe* PWRGD[a] active Power rails stable to PCIe* PERST# in active[b]	100 ms	-	As per PCIe* Specification.
T_{pgcfg}	External PWRGD[a] (PERST#[b]) signal to first configuration cycle.	100 ms		As per PCIe* Specification.
t_{d0mem}	Ethernet controller programmed from D3hot to D0 state to next Ethernet controller access.	10 ms		As per PCIe* Management Specification.
t_{l2pg}	L2 link transition to PWRGD[a] de-assertion. L2 link transition to PERST# assertion[b].	0 ns		As per PCIe* Specification.
t_{l2clk}	L2 link transition to removal of PCIe* reference clock.	100 ns		As per PCIe* Specification.
T_{clkpg}	PWRGD[a] de-assertion to removal of PCIe* reference clock. PERST# assertion to removal of PCIe* reference clock[b].	0 ns		As per PCIe* Specification.
T_{pgdl}	PWRGD[a] de-assertion time. PERST# assertion time[b].	100 µs		As per PCIe* Specification.

a. GIO_PWR_GOOD for the **631xESB/632xESB**.
b. **82573E/82573V/82573L** only.

6.1.9.9 Timing Guarantees

The Ethernet controller guarantees the following start up and power state transition related timing parameters.

Parameter	Description	Min	Max	Notes
t_{ppg}	LAN power good delay from valid power rail (not applicable to the **631xESB/632xESB**).	80 ms 50 ms[a]	90 ms 60 ms[a]	
t_{ee}	EEPROM/NVM read duration.		20 ms 15 ms[a]	
$t_{ppg-clkint}$	PCIe* PWRGD[b] to internal PLL lock[c].	-	50 µs	
t_{clkpr}	Internal PCIe* PWRGD from external PCIe* PWRGD[b].		50 µs	
t_{pgtrn}	PCIe* PWRGD[b] to start of link training. PCIe* PERST# to start of link training[a].		20 ms	As per PCIe* Specification.
T_{pgres}	External PWRGD[b] response to first configuration cycle.		1 s	As per PCIe* Specification.

a. **82573E/82573V/82573L** only.
b. GIO_PWR_GOOD for the **631xESB/632xESB**.
c. Not applicable to the **82573E/82573V/82573L**.

6.2 Wake-Up

The Ethernet controller supports two types of wake up mechanisms:

- Advanced Power Management (APM) Wake Up
- PCIe* Power Management Wake Up

The PCIe* Power Management Wake Up uses the GIO_WAKE_N pin to wake the system. The Advanced Power Management Wake Up can be configured to use the GIO_WAKE_N pin.

6.2.1 Advanced Power Management Wakeup

Advanced Power Management Wake Up, or APM Wake Up, was previously known as Wake on LAN (WOL). It is a feature that has been present in the 10/100 Mb/s network adapters for several generations. Its basic function is to receive a broadcast or unicast packet with an explicit data pattern and respond by asserting a wake-up signal to the system. In the earlier generations, this was accomplished by using a special signal that ran across a cable to a defined connector on the motherboard. The adapter asserted the signal for approximately 50 ms to signal a wake up. If the Ethernet controller is configured appropriately, it uses an in-band PM_PME message to achieve this.

On power up, the Ethernet controller reads the APM Enable bits from the EEPROM/NVM Initialization Control Word 2 into the APM Enable (APME) bits of the Wakeup Control Register (WUC). These bits enable APM Wake Up.

When APM Wake Up is enabled, the Ethernet controller checks all incoming packets for Magic Packets*. When the Ethernet controller receives a matching Magic Packet, it acts accordingly. If the Assert PME on APM Wake Up (APMPME) bit is set in the WUC register, the Ethernet controller:

- Sets the PME_Status bit in the PMCSR and issues a PM_PME message. In some cases, this might first require assertion of the WAKE# signal to resume power and clock to the PCIe* interface.
- Stores the first 128 bytes of the packet in the Wake Up Packet Memory (WUPM).
- Sets the Magic Packet Received bit in the Wake Up Status Register (WUS).
- Sets the packet length in the Wake Up Packet Length Register (WUPL).

The Ethernet controller maintains the first magic packet received in the Wake Up Packet Memory (WUPM) until the software device driver writes a 0b to the Magic Packet Received MAG bit in the Wake Up Status Register (WUS).

APM Wake Up is supported in all power states and only disabled if a subsequent EEPROM/NVM read results in a cleared APM Wake Up bit or software explicitly writes a 0b to the APM Wake Up (APM) bit of the WUC register.

Note: For the **82571EB**, it is recommended that WoL be enabled on either port 0 or port 1 but not both. When using Intel's device drivers WoL can only be enabled on port 0.

6.2.2 PCIe Power Management Wakeup

The Ethernet controller supports PCIe* Power Management based wake ups. It can generate system wake-up events from three sources:

- Reception of a Magic Packet.
- Reception of a network wake-up packet.
- Detection of a link change of state.

Activating PCIe* Power Management Wake Up requires the following steps:

- The driver programs the WUFC register indicating the wake-up packets. It also supplies the necessary data to the IPv4 and IPv6 Address Table (IP4AT and IP6AT) and the Flexible Filter Mask Table (FFMT), Flexible Filter Length Table (FFLT), and the Flexible Filter Value Table (FFVT). It can also set the Link Status Change (LNKC) bit in the Wake Up Filter Control Register (WUFC) to cause wake up when the link changes state.
- At configuration time, the operating system writes a 1b to the PME_EN bit of the PMCSR.

After enabling wake up, the operating system typically writes 11b to the lower two bits of the PMCSR placing the Ethernet controller into low-power mode.

After wake up is enabled, the Ethernet controller monitors the incoming packets. First, it filters the packets according to its standard address filtering method and then filtering them with all of the enabled wake-up filters. If a packet passes both the standard address filtering and at least one of the enabled wake-up filters, the Ethernet controller:

- Sets the PME_Status bit in the PMCSR.
- Asserts GIO_WAKE_N (PE_WAKEN for the **631xESB/632xESB**) if the PME_EN bit in the PMCSR is set.

- Stores the first 128 bytes of the packet in the Wake Up Packet Memory.
- Sets one or more of the received bits in the Wake Up Status Register (WUS). (The Ethernet controller sets more than one bit if a packet matches more than one filter.)
- Sets the packet length in the Wake Up Packet Length Register (WUPL).

If it is enabled, a link state change wake up causes similar results, setting PME_Status, asserting GIO_WAKE_N (PE_WAKEN for the **631xESB/632xESB**) and setting the Link Status Changed (LNKC) bit in the Wake Up Status Register (WUS) when the link goes up or down.

GIO_WAKE_N (PE_WAKEN for the **631xESB/632xESB**) remains asserted until the operating system either writes a 1b to the PME_Status bit of the PMCSR or writes a 0b to the PME_EN bit.

After receiving a wake-up packet, the Ethernet controller ignores any subsequent wake-up packets until the software device driver clears all of the received bits in the Wake Up Status Register (WUS). It also ignores link change events until the software device driver clears the Link Status Change (LNKC) bit in the WUS.

Note: Wake on link change is not supported when the Ethernet controller is configured to Serializer/ Deserializer (SerDes) mode (not applicable to the **82573E/82573V/82573L**).

6.2.3 Wake-Up 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 Wake Up Filter Control Register (WUFC) is set to 1b.

6.2.3.1 Pre-Defined Filters

The following packets are supported by the Ethernet controller pre-defined filters:

- Directed Packet (including exact, multicast, and broadcast).
- Magic Packet.
- ARP/IPv4 Request Packet.
- Directed IPv4 Packet.
- Directed IPv6 Packet.

Each of these filters are enabled if the corresponding bit in the Wake Up Filter Control Register (WUFC) is set to 1b.

The explanation of each filter includes a table detailing which bytes at which offsets are compared to determine if the packet passes the filter. Both VLAN frames and LLC/Snap frames can increase the given offsets if they are present.

Directed Exact Packet

The Ethernet controller generates a wake-up event upon reception of any packet whose destination address matches one of the 16 valid programmed Receive Addresses if the Directed Exact Wake Up Enable bit is set in the Wake Up Filter Control Register (WUFC.EX).

Offset	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	Match any pre-programmed address.

Directed Multicast Packet

For multicast packets, the upper bits of the destination address in the incoming packet index a bit vector. This is the Multicast Table Array that indicates whether to accept the packet. If the Directed Multicast Wake Up Enable bit is set in the Wake Up Filter Control Register (WUFC.MC) and the indexed bit in the vector is 1b, then the Ethernet controller generates a wake-up 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	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	See above paragraph.

Broadcast

If the Broadcast Wake Up Enable bit in the Wake Up Filter Control Register (WUFC.BC) is set, the Ethernet controller generates a wake-up event when it receives a broadcast packet.

Offset	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address	FF FF FF FF FF FF	Compare	

Magic Packet*

The definition for a Magic Packet* can be located at http://www.amd.com/products/npd/overview/20212.html. Magic Packets are defined in the following paragraph.

The Ethernet controller expects the destination address to either:

- Be the broadcast address (FF.FF.FF.FF.FF.FF).
- Match the value in the Receive Address Register 0 (RAH0, RAL0). This is initially loaded from the EEPROM/NVM but might be changed by the software device driver.
- Match any other address filtering enabled by the software device driver.

The Ethernet controller looks for the contents of Receive Address Register 0 (RAH0, RAL0) as the embedded IEEE address. It will consider any non-FFh byte after a series of at least 6 FFs to be the start of the IEEE address for comparison purposes. For example, it catches the case of 7 FFs followed by the IEEE address. As soon as one of the first 96 bytes after a string of FFs does not match, it continues to search for another set of at least 6 FFs followed by the 16 copies of the IEEE address later in the packet. It should be noted that this definition precludes the first byte of the destination address from equaling FF.

A Magic Packet destination address must match the address filtering enabled in the configuration registers with the exception that broadcast packets will be considered to match even if the Broadcast Accept bit of the Receive Control Register (RCTL.BAM) is 0b. If APM Wake Up is enabled in the EEPROM/NVM, the Ethernet controller starts with the Receive Address Register 0 (RAH0, RAL0) loaded from the EEPROM/NVM. This allows the Ethernet controller to accept packets with the matching IEEE address before the driver comes up.

Offset	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	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).

Note: Accepting broadcast magic packets for wake up purposes when Broadcast Accept bit of the Receive Control Register (RCTL.BAM) is 0b differs from older generation Intel Gigabit Ethernet components. Previously, these devices initialized RCTL.BAM to 1b if APM was enabled in the EEPROM/NVM but then required that bit to equal 1b to accept broadcast Magic Packets, unless broadcast packets passed another perfect or multicast filter.

ARP/IPv4 Request Packet

The Ethernet controller supports the reception of ARP request packets for wake up if the ARP bit is set in the Wake Up Filter Control Register (WUFC). Four IPv4 addresses are supported and are programmed in the IPv4 Address Table (IP4AT). 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 with VLAN tagging on both Ethernet II and Ethernet SNAP types.

Offset	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	2	Type	0806h	Compare	ARP
14	2	Hardware Type	0001h	Compare	
16	2	Protocol Type	0800h	Compare	
18	1	Hardware Size	06h	Compare	
19	1	Protocol Address Length	04h	Compare	
20	2	Operation	0001h	Compare	
22	6	Sender Hardware Address	-	Ignore	
28	4	Sender IP Address	-	Ignore	
32	6	Target Hardware Address	-	Ignore	
38	4	Target IP Address	IP4AT	Compare	May match any of 4 values in IP4AT.

Directed IPv4 Packet

The Ethernet controller supports the reception of Directed IPv4 packets for wake up if the IPv4 bit is set in the Wake Up Filter Control Register (WUFC). Four IPv4 addresses are supported and are programmed in the IPv4 Address Table (IP4AT). A successfully matched packet must contain the station MAC address, a Protocol Type of 0800h, and one of the four programmed IPv4 addresses. The Ethernet controller also handles Directed IPv4 packets with VLAN tagging on both Ethernet II and Ethernet SNAP types.

Offset	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	2	Type	0800h	Compare	IP
14	1	Version/ HDR length	4Xh	Compare	Check IPv4.
15	1	Type of Service	-	Ignore	
16	2	Packet Length	-	Ignore	
18	2	Identification	-	Ignore	
20	2	Fragment Information	-	Ignore	
22	1	Time to Live	-	Ignore	
23	1	Protocol	-	Ignore	
24	2	Header Checksum	-	Ignore	
26	4	Source IP Address	-	Ignore	
30	4	Destination IP Address	IP4AT	Compare	May match any of 4 values in IP4AT.

Directed IPv6 Packet

The Ethernet controller supports reception of Directed IPv6 packets for wake up if the IPv6 bit is set in the Wake Up Filter Control Register (WUFC). One IPv6 address is supported and it is programmed in the IPv6 Address Table (IP6AT). A successfully matched packet must contain the station MAC address, a Protocol Type of 0800h, and the programmed IPv6 address. The Ethernet controller also handles Directed IPv6 packets with VLAN tagging on both Ethernet II and Ethernet SNAP types.

Offset	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	2	Type	0800h	Compare	IP

Offset	Number of Bytes	Field	Value	Action	Comment
14	1	Version / Priority	6Xh	Compare	Check IPv6.
15	3	Flow Label	-	Ignore	
18	2	Payload Length	-	Ignore	
20	1	Next Header	-	Ignore	
21	1	Hop Limit	-	Ignore	
22	16	Source IP Address	-	Ignore	
38	16	Destination IP Address	IP6AT	Compare	Match value in IP6AT.

6.2.3.2 Flexible Filter

The Ethernet controller supports a total of four flexible filters. Each filter can be configured to recognize any arbitrary pattern within the first 128 bytes of the packet. The flexible filter is configured by software programming mask values into the Flexible Filter Mask Table (FFMT), 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 must also enable the filter in the Wake Up Filter Control Register (WUFC) and enable the overall wake up functionality must be enabled by setting PME_EN in the PMCSR or the WUC register.

When flexible filtering is enabled, the flexible filters scan incoming packets for a match. If the filter encounters any byte in the packet where the mask bit is one and the byte does not match the byte programmed in the Flexible Filter Value Table (FFVT), then the filter will fail that packet. If the filter reaches the required length without failing the packet, it passes the packet and generates a wake-up event. It will ignore any mask bits set to one beyond the required length.

Note: The minimum length of the pattern that is supported is 2 bytes.

The following packets listed in subsequent sections are for reference purposes only. The flexible filter can be used to filter these packets.

IPX Diagnostic Responder Request Packet

An IPX Diagnostic Responder Request Packet must contain a valid MAC address, a Protocol Type of 8137h, and an IPX Diagnostic Socket of 0456h. It might include LLC/SNAP headers and VLAN tags. Since filtering this packet relies on the flexible filters, which use offsets directly specified by the operating system, the operating system must account for the extra offset due to the LLC/SNAP headers and VLAN tags.

Offset	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	

Offset	Number of Bytes	Field	Value	Action	Comment
12	2	Type	8137h	Compare	IPX
14	16	Some IPX Data	-	Ignore	
30	2	IPX Diagnostic Socket	0456h	Compare	

Directed IPX Packet

A valid Directed IPX Packet contains the station MAC address, a Protocol Type of 8137h, and an IPX Node Address equal to the station MAC address. It may include LLC/SNAP headers and VLAN tags. Since filtering this packet relies on the flexible filters, which use offsets directly specified by the operating system, the operating system must account for the extra offset due to the LLC/SNAP headers and VLAN tags.

Offset	Number of Bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC Header processed by main address filter.
6	6	Source Address		Skip	
12	8	Possible LLC/SNAP Header		Skip	
12	4	Possible VLAN Tag		Skip	
12	2	Type	8137h	Compare	IPX.
14	10	Some IPX Data	-	Ignore	
24	6	IPX Node Address	Receive Address 0	Compare	Must match Receive Address 0.

IPv6 Neighbor Discovery Filter

In Ipv6, a Neighbor Discovery packet is used for address resolution. A flexible filter can be used to check for a Neighborhood Discovery Packet.

6.2.3.3 Wake Up Packet Storage

The Ethernet controller saves the first 128 bytes of the wake-up packet in its internal buffer, which can be read through the Wake Up Packet Memory (WUPM) after the system has been woken up.

FLASH Memory Interface 7

7.1 Introduction

The **82571EB**, **82572EI**, and the **631xESB/632xESB** provide an external serial interface to a Flash, or Boot ROM, device such as the Atmel AT25F1024 or AT25FB512. All accesses to this device are controlled 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 serial Flash devices with up to 64 Mb (or 8 MB) of memory. The Flash size implemented with the Ethernet controller device may be encoded into bits in the EEPROM. The Flash and Expansion ROM BARs are reconfigured based on these EEPROM settings. For the **631xESB/632xESB**, the Flash interface type is SPI mode 0.

Note: Although the Ethernet controller supports devices with up to 8 MB of memory, larger devices may also be used. Accesses to memory beyond the Flash device size results in access wrapping since only the lower address bits are used by the Flash control unit.

7.1.1 Flash Interface Operation

The Ethernet controller provides two different methods for software access to the Flash.

1. Legacy Flash transactions. The Flash can be read from or written to any time the host CPU performs a read or a write operation to a memory location that is within the Flash address mapping.
2. Boot time. The Flash is also accessed at boot time in the space indicated by the Expansion ROM Base Address Register.

All accesses to the Flash require the appropriate command sequence for the device used. The specific Flash data sheet from the vendor should be referred to for more details regarding reading from or writing to the Flash. Flash accesses are based on a direct decode of CPU accesses to a memory window defined in either:

1. Ethernet controller Flash Base Address Register (PCIe* Control Register at offset 14h or 18h).
2. Address range of the IOADDR register defined by the IO Base Address Register (PCIe* Configuration Register at offset 18h or 20h).
3. Expansion ROM Base Address Register (PCIe* Control Register at offset 30h).

The Ethernet controller device controls accesses to the Flash when it decodes a valid access.

Notes:

1. Flash read accesses must always be assembled by the Ethernet controller each time the access is greater than a byte-wide access.
2. The Ethernet controller byte reads or byte writes to the Flash require 2 µs. The device continues to issue retry accesses during this time.
3. The Ethernet controller device supports only byte writes to the Flash.

Another method that software can access the Flash is by directly using the Flash 4-wire interface through the Flash Access Register (FLA). It can use this for reads, writes, or other Flash operations (such as accessing the Flash status register, etc.).

The Flash can be directly accessed by following these steps:

1. Write a 1b to the Flash Request bit (*FLA.FL_REQ*).
2. Read the Flash Grant bit (*FLA.FL_GNT*) until it becomes 1b. It remains 0b as long as there are other accesses to the Flash in progress.
3. Write or read the Flash using the direct access to the 4-wire interface as defined in the Flash Access Register (FLA). The exact protocol used depends on the Flash placed on the board and can be found in the appropriate datasheet.
4. Write a 0b to the Flash Request bit (*FLA.FL_REQ*).

7.1.2 Flash Write Control

The Flash is write controlled by the FWE bits in the EEPROM/Flash Control and Data Register (EEC.FWE). Write access to the Flash device when write operations are disabled should not be attempted (FWE not equal to10b). Behavior after this type of an operation is undefined and can result in device or system lock up.

After sending a one byte write to the Flash, software can verify whether the next byte can be sent. In other words, software checks if the write process in the Flash has finished by reading the Flash Access Register. If the *FLA.FL_BUSY* bit in the Flash Access Register is set, the current write operation is not complete. If this bit is cleared, software continues and writes the next byte to the Flash.

Note: Writing to the Serial Flash should take into account the limitations of the page boundaries which are specific to the flash memory type.

7.1.3 Flash Erase Control

If using an Atmel* Flash device, software can erase the Flash by setting the *Flash Erase Command* bit (31) in the Flash Access Register to 1b and the *Flash Write Enable Control* bits (5:4) in the EEPROM/Flash Control Register to 00b. For other Flash devices, the *Flash Erase Command* bit (31) cannot be used. Instead, follow the instructions in the appropriate Flash device specification for Flash erase procedures using a bit-banging mechanism.

Note: The Flash erase action (for Flash devices other than Atmel*) can only be done by using direct SPI commands.

Hardware gets this command and then sends the Erase command to the Flash. The erase process finishes automatically. Software can continue accessing the Flash after the erase process finishes. The process can also be checked by using the Flash Write control mechanism described in Section 7.1.2.

Ethernet Interface 8

8.1 Introduction

The PCIe* 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.

Note: The Ethernet controllers are optimized for full-duplex operation in 1000 Mb/s mode. Half-duplex 1000 Mb/s operation is not supported and is not recommended.

The PHY features 10/100/1000-BaseT signaling 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 the ability to shut down to an extremely low (powered-down) state when not needed, as well as ability to auto-negotiate to lower 10/100 Mb/s speeds when the system is in low power-states.

Note: The **82563EB/82564EB** (PHY) communicates with the **631xESB/632xESB** (MAC) over the GLCI interface. See Section 8.5 through Section 8.7 for details.

8.1.1 82571EB/82572EI GMII/MII Interface

The internal GMII/MII interface supports the onboard 10/100/1000 Mb/s BASE-T transceivers in full duplex operation and supports the onboard 10/100 Mb/s BASE-T transceivers in full or half duplex operation.

The MAC(s) might be configured to use a different media interface. While the most likely application is expected to be based on use of the internal copper PHY, the **82571EB/82572EI** supports the following potential configurations:

- Internal Copper PHY (GMII/MII mode)
- External TBI device such as an optical SerDes (TBI mode).

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 interface, used to communicate between the MAC and the PHY, supports 10/100/1000 Mb/s operation, with both half- and full-duplex operation at 10/100 Mb/s, and full-duplex operation at 1000 Mb/s.

The TBI mode used to communicate between the MAC and the external TBI interface device supports 1000 Mb/s at full-duplex operation.

8.1.2 82573E/82573V/82573L GMII/MII Interface

The GMII/MII interface, used to communicate between the MAC and the PHY, supports 10/100/1000 Mb/s operation, with both half- and full-duplex operation at 10/100 Mb/s, and full-duplex operation at 1000 Mb/s.

8.1.3 Internal MAC/PHY GMII/MII Interface[1]

The MAC and PHY communicate through an internal GMII/MII interface that can be configured for either 1000 Mb/s operation (GMII) or 10/100 Mb/s (MII) mode of operation. For proper network operation, both MAC and PHY must be properly configured (either explicitly via software or via hardware auto-negotiation) to identical speed and duplex settings. All MAC configuration is performed using device control registers mapped into system memory or I/O space; an internal MDIO/MDC interface accessible via software is used to configure the PHY operation.

The internal GMII mode of operation is similar to the MII mode of operation. GMII uses the same MDIO/MDC management interface and registers for PHY configuration as MII mode. These common elements of operation enable the MAC and PHY to cooperatively determine the link partner's operational capability and configure the hardware based on those capabilities.

Note: For the **631xESB/632xESB**, see Section 8.5.

8.1.4 MDIO/MDC[1]

The Ethernet controller implements an internal IEEE 802.3 MII Management Interface (also known as the Management Data Input/Output or MDIO Interface) between the MAC and PHY. This interface provides the MAC and software the ability to monitor and control the state of the PHY. The internal MDIO interface defines a physical connection, a special protocol that runs across the connection, and an internal set of addressable registers. The internal interface consists of a data line (MDIO) and clock line (MDC), which are accessible by software via the MAC register space.

- MDC (management data clock) — This signal is used by the PHY as a clock timing reference for information transfer on the MDIO signal. The MDC is not required to be a continuous signal and can be frozen when no management data is transferred. The MDC signal has a maximum operating frequency of 2.5 MHz.
- MDIO (management data I/O) — This internal signaling between the MAC and PHY logically represents a bi-directional data signal used to transfer control information and status to and from the PHY (to read and write the PHY management registers). Asserting and interpreting value(s) on this interface requires knowledge of the special MDIO protocol to avoid possible internal signal contention or miscommunication to/from the PHY.

Software may use MDIO accesses to read or write registers in either GMII/MII mode by accessing the Ethernet controller's MDIC register.

Note: For the **631xESB/632xESB**, see Section 8.5.

1. Not applicable to the **631xESB/632xESB**.

8.2 Duplex Operation for Copper

The Ethernet controller supports half-duplex and full-duplex 10/100 Mb/s MII mode or full-duplex 1000 Mb/s GMII mode. However, only full-duplex mode is supported when SerDes mode is used.

Configuration of the duplex operation of the Ethernet controller can be forced or determined via the Auto-Negotiation process. See Section 8.3 for details on link configuration setup and resolution.

8.2.1 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 full-duplex 10/100/1000Base-T mode, transmission and reception are delineated independently by the 10/100/1000Base-T control signals. Transmission starts upon the assertion of TX_EN, which indicates there is valid data on the TX_DATA bus driven from the MAC to the PHY. Reception is signaled by the PHY by the assertion of the RX_DV signal which indicates valid receive data on the RX_DATA lines to the MAC.

In SerDes mode[1], the transmission and reception of packets is indicated by symbols imbedded in the data stream. These symbols delineate the packet encapsulation and the protocol does not rely on other control signals.

8.2.2 Half Duplex

Note: The Ethernet controller operates in half duplex mode only when configured for 10/100Base-T (internal PHY mode for the **82571EB/82572EI**). 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 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. The MAC signals the PHY with TX_EN at the start of transmission.

In the case of a collision, the 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.

1. Not applicable to the **82573E/82573V/82573L**.

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 13.8.12).

8.2.3 Gigabit Physical Coding Sub-Layer (PCS) for TBI/SerDes[1]

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 SerDes mode and is bypassed in 10/100/1000Base-T (internal PHY for the **82571EB/82572EI**) 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.

8.2.3.1 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. Not applicable to the **82573E/82573V/82573L**.

8.2.3.2 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.
/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.

8.3 Auto-Negotiation and Link Setup

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 SerDes mode, the Ethernet controller provides the complete 802.3z PCS function in the MAC. For 10/100/1000Base-T mode, the PCS and Auto-Negotiation functions are maintained within the PHY.

Configuring the link can be accomplished by several methods ranging from software's forcing link settings, software-controlled negotiation, MAC-controlled auto-negotiation, to Auto-Negotiation initiated by a PHY. The following sections describe processes of bringing the link up including configuration of the Ethernet controller and the transceiver, as well as the various methods of determining duplex and speed configuration.

The process of determining link configuration differs slightly based on the specific link mode being used.

When operating in a 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 Ethernet controller.

When operating in a 10/100/1000Base-T mode (internal PHY mode for the **82571EB/82572EI**), the PHY performs Auto-Negotiation per 802.3ab clause 40 and extensions to clause 28. Link resolution is obtained by the MAC from the PHY after the link has been established. The MAC accomplishes this via the MDIO interface, via specific signals from the PHY to the MAC, or by MAC auto detection functions.

8.3.1 Fiber/TBI Link Configuration[1]

When using TBI link mode, link mode configuration may be performed using the PCS function in the **82571EB/82572EI** MAC. The hardware supports both hardware and software Auto-Negotiation methods for determining the link configuration, as well as allowing for manually configuration to force the link. Hardware Auto-Negotiation is the preferred method.

8.3.1.1 MAC Link Speed

TBI operations are only defined for 1000 Mb/s operation. Other link speeds are not supported. When configured for either of these Ethernet interfaces, the MAC speed-determination function is disabled and the Device Status register bits (STATUS.SPEED) bits indicate a value of 10b for 1000 Mb/s.

8.3.1.2 TBI/SerDes Mode Auto-Negotiation

At power up, or Ethernet controller reset via the GIO_PWR_GOOD 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.

1. Not applicable to the **82573E/82573V/82573L**.

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 on-chip PCS and TBI interface to an internal SerDes.

TBI Mode Auto-Negotiation is used to determine the following information:

- Duplex resolution (though the Ethernet controller only supports full-duplex in TBI mode)
- Flow control configuration

Speed for TBI modes is fixed at 1000 Mb/s, so speed settings in the Device Control register are unaffected by the Auto-Negotiation process.

The following implementation is accessible in the design:

- A full hardware Auto-Negotiation implementation that does not require software intervention in order to successfully reach a negotiated link configuration.

A set of registers is provided to facilitate hardware Auto-Negotiation.

The IEEE 802.3z specification defines a set of resources that can be used to control a hardware implementation of Auto-Negotiation, but this definition is sub-optimal for TBI interfaces. It assumes the existence of a complete MII Management Register Set, that the Ethernet controller MAC does not implement.

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 TBI mode, 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.

8.3.1.3 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 ANE 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 software 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 8-1 and Figure 8-2 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 8-1. 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 8-2. 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.

8.3.1.4 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.

Note: 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.

Note: 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 13.3.19 and 13.3.20, respectively.

8.3.1.5 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. Forcing link may be accomplished by software writing a one to CTRL.SLU, which forces the MAC PCS logic into a link up state (enables listening to incoming characters when LOS is de-asserted by the internal SerDes).

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.

8.3.2 Link Configuration in 10/100/1000Base-T Mode

When operating with the 10/100/1000Base-T mode, link configuration is generally determined by PHY Auto-Negotiation. The software device driver must intervene in cases where a successful link is not negotiated or the user desires to manually configure the link. The following sections discuss the methods of link configuration for copper PHY operation.

8.3.2.1 Auto-Negotiation

In 10/100/1000Base-T 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.

Note: 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).

Note: 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 enables the MAC to recognize the LINK signal from the PHY, which indicates the PHY has established a valid link connection, and to receive and transmit data.

8.3.2.2 Forcing Link 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 PHY 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 13.3.1 for details.

When forcing the Ethernet controller to a specific speed configuration, the driver must 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.

Note: 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.

8.3.2.3 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. 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.

8.3.2.4 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

The standard set of PHY management registers can be found in the IEEE P802.3ab standard.

8.3.2.5 Comments Regarding Forcing Link

Forcing link in 10/100/1000Base-T 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/external 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.

8.3.3 Loss of Signal/Link Status Indication

The PHY LOS/LINK signal provides an indication of physical link status to the MAC. For the **82571EB/82572EI**, the LOS/LINK signal is through internal PHY mode or external TBI modes of operation). In 10/100/1000Base-T mode, this signal from the PHY indicates whether the link is up or down; typically indicated after successful Auto-Negotiation. Assuming that the MAC has been configured with CTRL.SLU = 1b, the MAC status bit STATUS.LU, when read generally reflects whether the PHY or SerDes has link (except under forced-link setup where even the PHY link indication might have been forced).

Note: When the MAC is configured for TBI mode[1], the input reflects loss-of-signal connection from the optics.

1. Not applicable to the **82573E/82573V/82573L**.

When the link indication from the PHY is de-asserted (or the loss-of-signal asserted from the SerDes[1]), the MAC considers this to be a transition to a link-down situation (cable unplugged, loss of link partner, etc.). If the LSC (Link Status Change) interrupt is enabled, the MAC generates an interrupt to be serviced by the software device driver. See Section 3.3.3 for more details.

Note: When operating in 10 Mb/s mode, the **631xESB/632xESB's** MAC should be reset after a link failure to prevent received packet data corruption after recovering from a link loss event.

8.4 10/100 Mb/s Specific Performance Enhancements

8.4.1 Adaptive IFS[1]

Adaptive IFS throttles back-to-back transmissions in the transmit packet buffer and delays their transfer to the CSMA/CD transmit function, and thus can be used to delay the transmission of back-to-back packets on the wire. This register also helps reduce collisions while in half duplex mode. 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.

The Adaptive IFS field provides a similar function to the IPGT field in the TIPG register (see Section 13.3.59). However, it only affects the initial transmission timing, not re-transmission timing.

Note: If the value of the AdaptiveIFS field is less than the IPGTransmitTime field in the Transmit IPG register then it has no effect, as the **82573E/82573V/82573L** selects the maximum of the two values.

8.4.2 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 8-3. 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

1. Not applicable to the **631xESB/632xESB**.

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.

8.4.3 MAC Control Frames & Reception of Flow Control Packets

Three comparisons are used to determine the validity of a flow control frame:

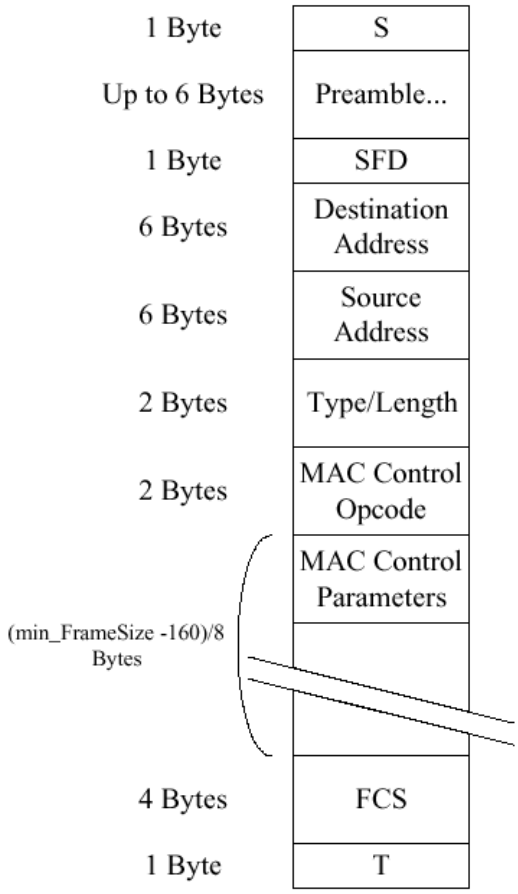
1. A match on the 6-byte multicast address for MAC Control Frames or to the station address of the device (Receive Address Register 0).
2. A match on the type field.
3. A comparison of the MAC Control Opcode field.

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 a 64-byte time or 512-bit time.



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 8-3. 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).

Note: 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.

8.4.4 Discard PAUSE Frames and Pass MAC Control Frames

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 13.3.33 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. When DPF is cleared to 0b, a valid flow control packet is transferred via DMA. This bit 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 a logic high.

8.4.5 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 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.

Note: 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.

8.4.6 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.

Note: Software sends an XON frame by programming a zero in the PAUSE timer field of the FCTTV register.

8.5 GLCI Interface (631xESB/632xESB and 82563EB/82564EB Only)

The **82563EB/82564EB** communicates with the **631xESB/632xESB** over the GLCI interface (GLCI mode). The **82563EB/82564EB** serial interface is electrically compatible with the SerDes implemented in 1000Base-BX applications, as defined in the PICMG 3.1 specification, version 1.0, chapter 5, Backplane physical layers interfaces. There is one serial interface per port.

8.5.1 GLCI Operation

GLCI is an Intel proprietary SerDes-based MAC-PHY interface. It is a 4-pin (RX +/-, TX +/-) differential interface for each Ethernet port. Each port has a separate GLCI interface that operates independently. The clock is embedded within the differential signals and runs at 1.25 GHz, although the bus is capable of passing data at 10, 100, or 1000 Mb/s. There is no need for the additional MDIO/MDC signals as the PHY management information is passed in-band. Special encoded packets messages carry the MDIO information. The **82563EB/82564EB** implements revision 1.71 of the GLCI specification.

8.5.2 Inband Messages

Each port receives the following information from the **631xESB/632xESB** over each GLCI bus:

- Port Reset bit.
- D-state (ACPI power state) encoded as either D0u, D0a, D3, or Dr4.
- LED indications for four LEDs.
- MDIO requests (including register address, read/write indication, and write data for writes)

The **82563EB/82564EB** transmits the following information back to the **631xESB/632xESB** over each GLCI bus for each port:

- PHY link up/down
- Duplex mode (full/half)
- Speed (10 Mb/s, 100 Mb/s, or 1000 Mb/s)
- Acknowledge of MDIO requests, including read data for reads.

When the PHY/MAC's status changes, the port sends a GLCI status message indicating the status change. It expects to receive an acknowledgement from the other side (the **82563EB/82564EB** expects an acknowledge from the **631xESB/632xESB** and it expects to get an acknowledge from the **82563EB/82564EB**). If it doesn't receive an acknowledgement during the Inband Status Acknowledge Timeout period, it re-sends the status message up to the Max. Retries limit. Even if there is no change, or if the Max. Retries fails, the **82563EB/82564EB** and the **631xESB/632xESB** periodically re-sends their PHY/MAC status every link status retransmission period. They also expect an acknowledge from this transmission and retransmit it until an acknowledge is received or the maximum number of retries is reached.

Note: For more details on the operation of the GLCI bus, see the GLCI specification.

8.6 Ethernet Port

Each copper 1000Base-T compatible port is implemented by an integrated PHY. Each port conforms to the 1000Base-T protocol and electricals, which are defined in clause 40 of the 802.3-2002 specification. The port consists of four bi-directional, differential twisted pair signals that can be over 100 m long. Data is signaled at 125 million symbols/second per pair. Using 4 pairs and a complex symbol encoding allows an actual data rate of 1Gb/s to be achieved.

The same signals can be used to operate in 100Base-TX and 10Base-T modes for interfacing with lower performance link partners at 100 Mb/s and 10 Mb/s, respectively. When running at 100 Mb/s or 10 Mb/s one signal pair is use for transmit and a different one is used for receive.

8.7 GLCI In-Band Access

In GLCI mode, a port's registers are normally accessed through GLCI in-band MDIO messages. This includes the PHY registers, GLCI/Serial registers, and other top-level port registers. By default this is enabled, although it is disabled when top-level MDIO access is enabled.

The GLCI bus interface can only be used to access registers associated with a single port. Because of this no PHY address is required. Just like a standard MDIO access, the GLCI bus carries inband messages that carry a 5-bit register address and uses the same paging mechanism described in Section 13.

802.1q VLAN Support 9

The PCIe* 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

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	802.1q VLAN Packet	#Octets
DA	6	DA	6
SA	6	SA	6
Type/Length	2	8021.q Tag	4
Data	46-1500	Type/Length	2
CRC	4	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 (not applicable to the **82573E/82573V**), 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.

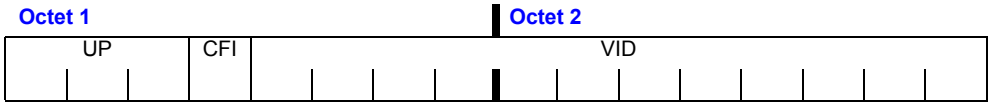
9.1.1 802.1q Tagged Frames

For 802.1q, the Tag Header field consists of four octets comprised of the Tag Protocol Identifier (TPID) and Tag Control Information (TCI), each taking 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 13.3.33.
- VLAN Identifier (VID)

Table 9-2. 802.1q Tagged Frames



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.

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 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 comprised of 128 32-bit registers. Matching to this bit vector follows the same algorithm as indicated in Section 13.4.1 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 13.3.33), 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 is intentionally left blank.

Configurable LED Outputs 10

10.1 Configurable LED Outputs

The PCIe* Family of Gigabit Ethernet Controllers implements four (three for the **82573E/82573V/82573L**) output drivers intended for driving external LED circuits per port. Each of the four (three for the **82573E/82573V/82573L**) LED outputs can be individually configured to select the particular event, state, or activity, which is indicated on that output. In addition, each LED can be individually configured for output polarity as well as for blinking versus non-blinking (steady-state) indication.

The configuration for LED outputs is specified via the LEDCTL register. In addition, the hardware-default configuration for all the LED outputs, can be specified via EEPROM/NVM fields, thereby supporting LED displays configurable to a particular OEM preference.

Each of the four (three for the **82573E/82573V/82573L**) LED's can be configured to use one of a variety of sources for output indication. The MODE bits control the LED source:

- LINK_100/1000 is asserted when link is established at either 100 or 1000 Mb/s.
- LINK_10/1000 is asserted when link is established at either 10 or 1000 Mb/s.
- LINK_UP is asserted when any speed link is established and maintained.
- ACTIVITY is asserted when link is established and packets are being transmitted or received.
- LINK/ACTIVITY is asserted when link is established AND there is NO transmit or receive activity
- LINK_10 is asserted when a 10 Mb/s link is established and maintained.
- LINK_100 is asserted when a 100 Mb/s link is established and maintained.
- LINK_1000 is asserted when a 1000 Mb/s link is established and maintained.
- FULL_DUPLEX is asserted when the link is configured for full duplex operation.
- COLLISION is asserted when a collision is observed.
- PAUSED is asserted when the Ethernet controller's transmitter is flow controlled.
- LED_ON is always asserted; LED_OFF is always de-asserted.

The IVRT bits enable the LED source to be inverted before being output or observed by the blink-control logic. LED outputs are assumed to normally be connected to the negative side (cathode) of an external LED.

The BLINK bits control whether the LED should be blinked (either 200 ms on and 200 ms off or 83 ms on and 83 ms off[1]) while the LED source is asserted. The blink control might be especially useful for ensuring that certain events, such as ACTIVITY indication, cause LED transitions, which are sufficiently visible to a human eye.

Note: The LINK/ACTIVITY source functions slightly different from the others when BLINK is enabled. The LED is off if there is no LINK, on if there is LINK and no ACTIVITY, and blinking if there is LINK and ACTIVITY.

1. While in Smart Power-Down mode (not applicable to the **631xESB/632xESB**), the blinking durations are increased by 5 times to 1 s and 415 ms respectively.

PHY Functionality and Features 11

Note: This section is not applicable to the **82573E/82573V/82573L**.

11.1 Auto-Negotiation

Auto-Negotiation between the PCIe* 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 as 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
- PHY detects loss of link
- PHY detects re-appearance of energy on the link
- 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
- PHY hardware reset asserted using the MAC CTRL.PHYRST bit
- PHY soft-reset initiated via the PHY Control Register (bit 15)
- Explicit Auto-Negotiation Re-Start initiated via the PHY Control Register (bit 9)
- Explicit transition of PHY from internal IEEE power-down to normal mode via the PHY Control Register (bit 11) or the Copper Specific Control Register (bit 2) for the **631xESB/632xESB**.

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 without any software intervention. If the link partner 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.

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.

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 (Auto-Negotiation Expansion Register bit 15) can be set to 1b 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 Auto-Negotiation Next Page Transmit Register (NPT) and Auto-Negotiation Link Partner Next Page Register (LPN), respectively. The PHY stores the previous results from the LPN register in internal registers so that new next pages can overwrite the LPN register.

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 (PCTRL) bits 6, 8, 12, and 13 do not take effect unless one of the following takes place:

- PHY soft reset (PHY Control Register bit 15) for the **631xESB/632xESB**
- 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.

The Auto-Negotiation Expansion and the 100BASE-T Control registers are internally latched every time 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 100BASE-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 in a similar way as the Auto-Negotiation Expansion and 100BASE-T Control registers during additional next page exchanges.

11.1.4 Status

Once the PHY completes auto-negotiation, it updates various statuses in the PHY Status Register, Link Partner Ability Register (Base Page), Auto-Negotiation Expansion Register, and 1000BASE-X 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 and the PHY Interrupt Status Register.

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 completes auto-negotiation.

11.2 MDI/MDI-X Crossover

The Ethernet controller PHY automatically determines whether or not it needs to cross over between pairs as shown in the following table so that a 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 with 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 Port Control register (bits 13:12) for the **82571EB/82572EI** and the Copper Specific Control register (bits 6:5) for the **631xESB/632xESB**.

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

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

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 PHY to auto-negotiate with another gigabit link partner using a two or three pair cable and downshift to link at 100 Mb/s or 10 Mb/s, whichever is the highest speed below gigabit that the link partner is capable of.

By default, downshift is turned on. Refer to Table 13-45 to disable the downshift feature.

11.3 Cable Length Detection

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

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

When the PHY detects loss of link, it initiates the auto-negotiation process. 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 initiates auto-negotiation by sending FLPs for five seconds. If at the end of five seconds auto-negotiation has not completed, then the PHY stops sending FLPs and returns to the energy-detect power down state. While monitoring receive energy, the PHY sends out a single 10 Mb/s Normal Link Pulse (NLP) 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 Power Management register bit 4 for the **82571EB/82572EI** and the Copper Specific Control register (bits 9:8) for the **631xESB/632xESB** are used to configure the specific Energy-Detect mode of behavior for the PHY.

11.4.2 D3 State, No Link Required

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 transmit any NLPs on the wire itself.

When the MAC transitions back to D0 power-states, either through explicit system/software mechanisms or by hardware reset operations, the PHY returns to a functional power-state, re-initiates auto-negotiation, advertises all possible speeds (10/100/1000 Mb/s), and reverts to an energy-detect power down state if unsuccessful in establishing link.

11.4.3 D3 Link-Up, Speed-Management Enabled

If the MAC is configured for PHY power management (CTRL.EN_PHY_PWR_MGMT = 1b) and the PHY is linked at 1000 Mb/s, 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 Mb/s capability. This results in D3 operation at a lower-speed link and a reduced power level.

If wakeup, SmartSpeed, 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 Mb/s speed capability, in order to return to maximum-speed operation.

11.4.4 D3 Link-Up, Speed-Management Disabled

If the MAC is configured for no PHY power management (CTRL.EN_PHY_PWR_MGMT = 0b), and the MAC transitions to D3 power-states (either wakeup or manageability are enabled) where link is required for either wakeup or manageability, then the PHY simply remains operational at its current line speed. This configuration is not recommended, since D3 power consumption at 1000 Mb/s exceeds 20 mA Vaux.

11.5 Initialization

During power-up or reset, the PHY core performs the initialization as shown in Figure 11-1. The software driver has access to PHY register 0d, bits 15 and 11 for PHY reset and PHY Power Down control, respectively.

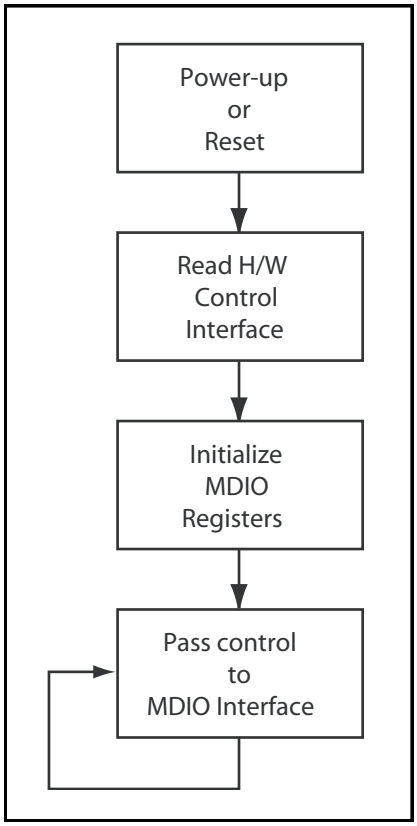


Figure 11-1. PHY Initialization Sequence

11.5.1 MDIO Control Mode

In the MDIO Control mode, the PHY uses the Hardware Control Interface to set up the 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 achieve link using one of the following 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 FLPs. 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.

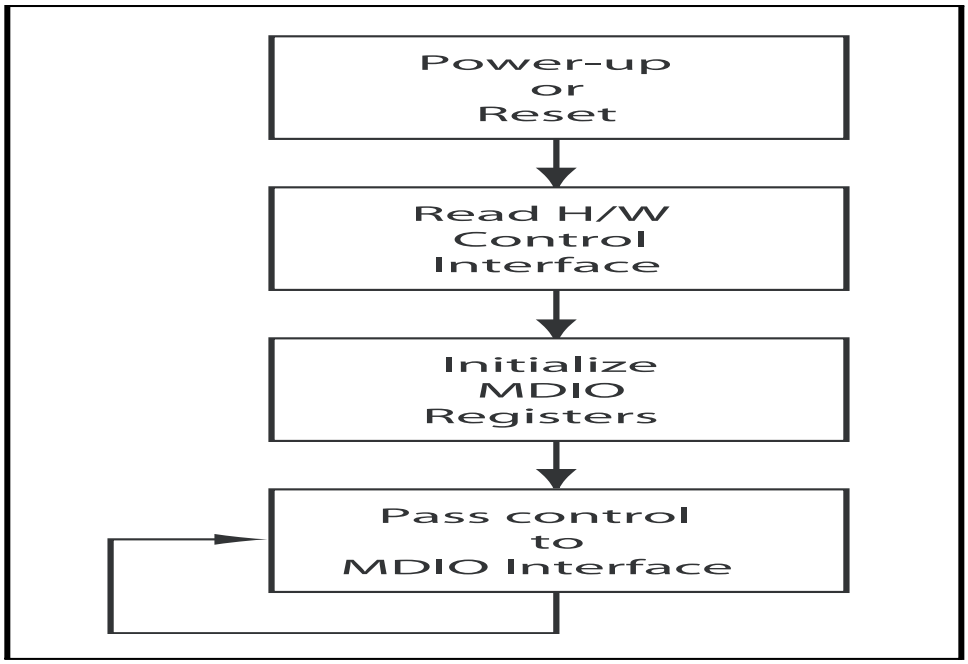


Figure 11-2. Overview of Link Establishment

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 Mb/s and 100 Mb/s links (1000 Mb/s 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 Mb/s, 100 Mb/s, or 1000 Mb/s) and duplex state (full or half). For Gigabit (1000 Mb/s) links, the programmer must explicitly designate one side as the Master and the other as the Slave. Table 11-1 summarizes link establishment procedures.

Table 11-1. 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 NLPs instead. If 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 established 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 or data 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 100 Mb/s 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.

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 11-2 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 11-2. 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 put into a low-power state according to MAC control (Power Management controls) or via PHY register 0d. In two mechanisms to achieve a single power-down state, 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 = 0b), it re-initializes all analog functions, but retains its previous configuration settings**.**

11.10.2 Smart Power-Down[1]

Smart Power-Down (SPD) is a link-disconnect capability applicable to all power management states, and is intended for mobile applications. Smart power down combines a power saving mechanism with the fact that link might disappear and resume.

SPD is enabled by PHY register 25d, bit 0 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.

Two link partners connected back-to-back can experience link failures if both are allowed to enter the SPD state.

11.11 1000 Mb/s Operation

11.11.1 Introduction

This section provides an overview of 1000BASE-T functions.

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.

1. Not applicable to the **631xESB/632xESB**.

11.12 100 Mb/s 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 Mb/s Operation

The PHY operates as a standard 10 Mb/s 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 Mb/s functions.

Note: When operating in 10 Mb/s mode, the **631xESB/632xESB's** MAC should be reset after a link failure to prevent received packet data corruption after recovering from a link loss event.

11.13.1 Link Test[1]

In 10 Mb/s 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[1]

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 (bit 0 for the **631xESB/632xESB**) to 1b.

1. Not applicable to the **631xESB/632xESB**.

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.

Dual Port Characteristics 12

Note: This section applies to the **82571EB**, the **631xESB/632xESB**, and the **82563EB** only.

12.1 Introduction

The Ethernet controller architecture includes two instances both the MAC and PHY. With both MAC/PHY pairs operating, the Ethernet controller appears as a multi-function PCIe* device containing two identically-functioning devices. To avoid confusion, each MAC (when combined with either an internal/external PHY or SerDes) is referred to as "LANx", where x = "A" or x = "B" to refer to each logical LAN device (LAN 0 or LAN 1).

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 PCIe* 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 12.6 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 PCIe* Interface

The Ethernet controller contains a single physical PCIe* core interface. The Ethernet controller is designed so that each of the logical LAN devices (LAN 0, LAN 1) appears as a distinct function implementing, amongst other registers, the following PCIe* 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 (01h:03h)
Ch	BIST 00h	Header Type 00h	Latency Timer	Cache Line Size
10h	Base Address 0			
14h	Base Address 1			
18h	Base Address 2			

Byte Offset	Byte 0	Byte 1	Byte 2	Byte 3
1Ch	Base Address 3			
20h	Base Address 4			
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 02h)	Interrupt Line 00h

Many of the fields of the PCIe* 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 13.3, 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. An outstanding read for one LAN device does not impact the Ethernet controller's ability to accept a register access to the other LAN. PCIe* bus operation, where each register access results in a completion by one LAN device, in no way prevents the other LAN device from accepting and servicing its register space as the Ethernet controller's PCIe* core supports two credits for memory accesses.

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).

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 0 and LAN 1 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 0/1 Shared”. Those words configuring a specific LAN device parameters are identified as either “LAN 0” or “LAN 1”.

The following EEPROM words warrant additional notes specifically related to dual-LAN support:

Ethernet Address (IA) (LAN 0/1 shared)	The EEPROM specifies the IA associated with the LAN 0 device and used as the hardware default of the Receive Address Registers for that device. The hardware-default IA for the LAN 1 device is automatically determined by the same EEPROM word, and is set to the value of (IA LAN 0 XOR 010000000000b)
Initialization Control 1, Initialization Control 2 (LAN 0/1 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, PCIe* bus advertised capabilities, etc.
Initialization Control 3 (LAN 0, LAN 1 unique)	This EEPROM word configures default values associated with each LAN device's hardware connections, including which link mode (internal/external PHY, internal/external TBI or 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 internal TBI transceiver interface at once (**82571EB/82572EI** only).

12.4 Shared FLASH

The Ethernet controller provides an interface to an external FLASH/ROM memory device, as described in Section 5. 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

The Ethernet controller implements 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 size window, The first one is served first and only then the next one, Note that the Ethernet controller does not synchronize between the two entities accessing the Flash though contentions caused from one entity reading and the other modifying the same locations is possible.

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 Link Mode/Configuration[1]

The **82571EB** provides significant amount of flexibility in pairing a LAN device with a particular type of media (copper or fiber-optic) as well as the specific transceiver/interface used to communicate with the media. Each MAC, representing a distinct LAN device, can be coupled with an internal copper PHY (the default) or SerDes interface independently. The link configuration specified for each LAN device can be specified in the LINK_MODE field of the Extended Device Control Register (CTRL_EXT) and initialized from the EEPROM Initialization Control Word 3 associated with each LAN device

12.6 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.

12.6.1 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.

When a particular LAN is disabled, all internal clocks to that LAN are disabled, the device is held in reset, and the internal/external 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: When both LAN functions are disabled, MMS cannot be used.

12.6.2 Values Sampled on Reset

The Ethernet controller samples the LAN0_Dis_N and LAN1_Dis_N pins on the rising edge of the 3GIO_PWR_Good, GIO_PWR_GOOD or In-Band PCIe* Reset. Based on the values sampled, the LAN devices are enabled/disabled according to the following table:

Pin Sampled	LAN Device Controlled	Enable/Disable
LAN0_Dis_N	LAN 0 device	Vcc/logic 1b = enabled Vss/logic 0b = disabled
LAN1_Dis_N	LAN 1 device	

1. Not applicable to the **631xESB/632xESB**.

12.6.3 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 80h 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.

12.6.4 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.

12.6.5 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.

Register Descriptions 13

13.1 Introduction

This section details the state inside the PCIe* 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 internal register/memory space is described in the following sections and divided up into the following categories:

- General
- Interrupt
- MAC Receive
- MAC Transmit
- Statistics
- Wake Up
- Diagnostics (including packet buffer memory access)
- PHY Receive, Transmit and Special Function

Note: The PHY registers are accessed indirectly through the MDI/O interface.

The Ethernet controller's address space is mapped into four regions with PCI Base Address Registers described in the table below. These regions are shown as follows.

Table 13-1. PCI Base Address Registers

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 Windows Mapped	32 Bytes

Both the Flash and 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.

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:

- Register pairs where two 32-bit registers make up a larger logical size
- Accesses to Flash memory (through the Expansion ROM space, secondary BAR space, or the I/O space) can be byte, word, or double word 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 address 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/NVM 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.

Note: Partial reads to read-on-clear registers (like 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. Though some logical statistics registers represent counters in excess of 32-bits in width, registers must be accessed using 32-bit operations. For example, independent access to each 32-bit field.

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). Refer to Table 13-1 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). The Flash is only accessible if enabled through the EEPROM/NVM 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 reference the Flash provided that access is enabled through the EEPROM/NVM 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

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), the BAR contains a valid (non-zero value), and I/O address decoding is enabled in the PCIe* 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
00000008h - 0000001Fh	Reserved	Reserved.	RO	4 bytes

13.2.2.1 IOADDR

The IOADDR register must always be written as a DWORD access (for example, the 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 PCIe* 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.

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. Table 13-2 lists the supported configurations:

Table 13-2. IODATA Register Configurations

Access Type	Ethernet Controller IOADDR Register Bits [1:0]	Target IODATA Access BE[3:0]# Bits in Data Phase
BYTE (8 bits)	00b	1110b
	01b	1101b
	10b	1011b
	11b	0111b
WORD (16 bits)	00b	1100b
	10b	0011b
DWORD (32 bits)	00b	0000b

Note: 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 13-2:

```
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 can be indeterminate.

Note: 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. For example, a split transaction or transaction retry.

Because a register/memory/flash read or write takes two IO cycles to complete, software must provide a guarantee that the two IO cycles occur as an atomic operation. Otherwise, results can be indeterminate.

13.2.2.3 Undefined I/O Offsets

I/O offsets 00000008h through 0000001Fh are considered to be reserved offsets with the I/O window. Dword reads from these addresses will return FFFFh; writes to these addresses are discarded.

Table 13-3. Ethernet Controller Register Summary

Category	Offset	Abbreviation	Name	R/W	Page
General	00000h 00004h	CTRL	Device Control	R/W	278
General	00008h	STATUS	Device Status	RO	282
General	00010h	EEC	EEPROM/Flash Control	R/W	284
General	00014h	EERD	EEPROM Read	R/W	287
General	00018h	CTRL_EXT	Extended Device Control	R/W	288
General	0001Ch	FLA	Flash Access	R/W	292
General	00020h	MDIC	MDI Control	R/W	294
General	00024h	SERDESCTL	Serdes_ana	R/W	355
General	00028h	FCAL	Flow Control Address Low	R/W	355
General	0002Ch	FCAH	Flow Control Address High	R/W	356
General	00030h	FCT	Flow Control Type	R/W	356
General	00034h	KUMCTRLSTA	GLCI Control and Status Register (the **631xESB/632xESB**)	R/W	356
General	00038h	VET	VLAN EtherType	R/W	358
General	00170h	FCTTV	Flow Control Transmit Timer Value	R/W	359
General	00178h	TXCW	Transmit Configuration Word	R/W	360

Table 13-3. (Continued) Ethernet Controller Register Summary

Category	Offset	Abbreviation	Name	R/W	Page
General	00180h	RXCW	Receive Configuration Word	RO	362
General	00E00h	LEDCTL	LED Control	R/W	365
General	01000h	PBA	Packet Buffer Allocation	R/W	369
General	01010h	EEMNGCTL	MNG EEPROM Control (**82571EB**)	R/W	369
General	05B5Ch	SW_FW_SYNC	Software/Firmware Synchronization (**631xESB/632xESB**)	R/W	369
Interrupt	000C0h	ICR	Interrupt Cause Read	R/ CLR	372
Interrupt	000C4h	ITR	Interrupt Throttling Rate	R/W	375
Interrupt	000C8h	ICS	Interrupt Cause Set	WO	376
Interrupt	000D0h	IMS	Interrupt Mask Set/Read	R/W	378
Interrupt	000D8h	IMC	Interrupt Mask Clear	WO	379
Interrupt	000E0h	IAM	Interrupt Acknowledge Auto Mask	R/W	380
Receive	00100h	RCTL	Receive Control	R/W	380
Receive	02008h	ERT	Early Receive Threshold (**82573E/82573V/ 82573L** only)	R/W	385
Receive	02160h	FCRTL	Flow Control Receive Threshold Low	R/W	386
Receive	02168h	FCRTH	Flow Control Receive Threshold High	R/W	387
Receive	02170h	PSRCTL	Packet Split Receive Control	R/W	388
Receive	02800h	RDBAL0	Receive Descriptor Base Low Queue 0	R/W	389
Receive	02804h	RDBAH0	Receive Descriptor Base High Queue 0	R/W	389
Receive	02808h	RDLEN0	Receive Descriptor Length Queue 0	R/W	390
Receive	02810h	RDH0	Receive Descriptor Head Queue 0	R/W	390
Receive	02818h	RDT0	Receive Descriptor Tail Queue 0	R/W	391
Receive	02820h	RDTR	Receive Interrupt Packet Delay Timer	R/W	391
Receive	02828h	RXDCTL	Receive Descriptor Control	R/W	392
Receive	0282Ch	RADV	Receive Interrupt Absolute Delay Timer	R/W	393
Receive	02900h	RDBAL1	Receive Descriptor Base Low Queue 1	R/W	389
Receive	02904h	RDBAH1	Receive Descriptor Base High Queue 1	R/W	389
Receive	02908h	RDLEN1	Receive Descriptor Length Queue 1	R/W	390
Receive	02910h	RDH1	Receive Descriptor Head Queue 1	R/W	390

Table 13-3. (Continued) Ethernet Controller Register Summary

Category	Offset	Abbreviation	Name	R/W	Page
Receive	02918h	RDT1	Receive Descriptor Tail Queue 1	R/W	391
Receive	02928h	RXDCTL1	Receive Descriptor Control 1	R/W	392
Receive	02C00h	RSRPD	Receive Small Packet Detect	R/W	398
Receive	02C08	RAID	Receive ACK Interrupt Delay	R/W	398
Receive	02C10h	CPUVEC	CPU Vector	R/W	399
Receive	05000h	RXCSUM	Receive Checksum Control	R/W	399
Receive	05008h	RFCTL	Receive Filter Control	R/W	400
Receive	05200h-053FCh	MTA[127:0]	Multicast Table Array (n)	R/W	420
Receive	05400h	RAL	Receive Address Low	R/W	422
Receive	05404h	RAH	Receive Address High)	R/W	422
Receive	05600h-057FCh	VFTA[127:0]	VLAN Filter Table Array (n)	R/W	423
Receive	05818h	MRQC	Multiple Receive Queues Command	R/W	424
Receive	05864h	RSSIM	RSS Interrupt Mask	R/W	425
Receive	05868h	RSSIR	RSS Interrupt Request	R/W	425
Receive	05C00-05C3Fh	RETA	Redirection Table	R/W	426
Receive	05C80-05CA7h	RSSRK	RSS Random Key	R/W	426
Transmit	00400h	TCTL	Transmit Control	R/W	401
Transmit	00410h	TIPG	Transmit IPG	R/W	404
Transmit	00458h	AIT	Adaptive IFS Throttle (Not applicable to the **631xESB/632xESB**)	R/W	405
Transmit	03800h	TDBAL	Transmit Descriptor Base Low	R/W	406
Transmit	03804h	TDBAH	Transmit Descriptor Base High	R/W	406
Transmit	03808h	TDLEN	Transmit Descriptor Length	R/W	407
Transmit	03810h	TDH	Transmit Descriptor Head	R/W	407
Transmit	03818h	TDT	Transmit Descriptor Tail	R/W	408
Transmit	03820h	TIDV	Transmit Interrupt Delay Value	R/W	408
Transmit	03828h	TXDCTL	Transmit Descriptor Control	R/W	409
Transmit	0382Ch	TADV	Transmit Absolute Interrupt Delay Value	R/W	411

Table 13-3. (Continued) Ethernet Controller Register Summary

Category	Offset	Abbreviation	Name	R/W	Page
Transmit	03840h	TARC0	Transmit Arbitration Counter Queue 0	R/W	419
Transmit	03900h	TDBAL1	Transmit Descriptor Base Low Queue 1	R/W	406
Transmit	03904h	TDBAH1	Transmit Descriptor Base High Queue 1	R/W	406
Transmit	03908h	TDLEN1	Transmit Descriptor Length Queue 1	R/W	407
Transmit	03910h	TDH1	Transmit Descriptor Head Queue 1	R/W	407
Transmit	03918h	TDT1	Transmit Descriptor Tail Queue 1	R/W	408
Transmit	03928h	TXDCTL1	Transmit Descriptor Control 1	R/W	409
Transmit	03940h	TARC1	Transmit Arbitration Counter Queue 1	R/W	411
Statistics	04000h	CRCERRS	CRC Error Count	R	455
Statistics	04004h	ALGNERRC	Alignment Error Count	R	456
Statistics	04008h	SYMERRS	Symbol Error Count	R	456
Statistics	0400Ch	RXERRC	RX Error Count	R	456
Statistics	04010h	MPC	Missed Packets Count	R	457
Statistics	04014h	SCC	Single Collision Count	R	457
Statistics	04018h	ECOL	Excessive Collisions Count	R	458
Statistics	0401Ch	MCC	Multiple Collision Count	R	458
Statistics	04020h	LATECOL	Late Collisions Count	R	458
Statistics	04028h	COLC	Collision Count	R	459
Statistics	04030h	DC	Defer Count	R	459
Statistics	04034h	TNC-S	Transmit - No CRS	R	460
Statistics	04038h	SEC	Sequence Error Count	R	460
Statistics	04040h	RLEC	Receive Length Error Count	R	461
Statistics	04048h	XONRXC	XON Received Count	R	462
Statistics	0404Ch	XONTXC	XON Transmitted Count	R	462
Statistics	04050h	XOFFRXC	XOFF Received Count	R	463
Statistics	04054h	XOFFTXC	XOFF Transmitted Count	R	463
Statistics	04058h	FCRUC	FC Received Unsupported Count	R	464
Statistics	0405Ch	PRC64	Packets Received (64 Bytes) Count	R	464
Statistics	04060h	PRC127	Packets Received (65-127 Bytes) Count	R	464
Statistics	04064h	PRC255	Packets Received (128-255 Bytes) Count	R	465

Table 13-3. (Continued) Ethernet Controller Register Summary

Category	Offset	Abbreviation	Name	R/W	Page
Statistics	04068h	PRC511	Packets Received (256-511 Bytes) Count	R	465
Statistics	0406Ch	PRC1023	Packets Received (512-1023 Bytes) Count	R	466
Statistics	04070h	PRC1522	Packets Received (1024-Max Bytes)	R	466
Statistics	04074h	GPRC	Good Packets Received Count	R	466
Statistics	04078h	BPRC	Broadcast Packets Received Count	R	467
Statistics	0407Ch	MPRC	Multicast Packets Received Count	R	467
Statistics	04080h	GPTC	Good Packets Transmitted Count	R	468
Statistics	04088h	GORCL	Good Octets Received Count (Low)	R	468
Statistics	0408Ch	GORCH	Good Octets Received Count (Hi)	R	468
Statistics	04090h	GOTCL	Good Octets Transmitted Count (Low)	R	469
Statistics	04094h	GOTCH	Good Octets Transmitted Count (Hi)	R	469
Statistics	040A0h	RNBC	Receive No Buffers Count	R	469
Statistics	040A4h	RUC	Receive Undersize Count	R	470
Statistics	040A8h	RFC	Receive Fragment Count	R	470
Statistics	040ACh	ROC	Receive Oversize Count	R	470
Statistics	040B0h	RJC	Receive Jabber Count	R	471
Statistics	040B4h	MPRC	Management Packets Received Count	R	471
Statistics	040B8h	MPDC	Management Packets Dropped Count	R	471
Statistics	040BCh	MPTC	Management Pkts Transmitted Count	R	472
Statistics	040C0h	TORL	Total Octets Received (Lo)	R	472
Statistics	040C4h	TORH	Total Octets Received (Hi)	R	472
Statistics	040C8h	TOTL	Total Octets Transmitted (Lo)	R	472
Statistics	040CCh	TOTH	Total Octets Transmitted (Hi)	R	472
Statistics	040D0h	TPR	Total Packets Received	R	473
Statistics	040D4h	TPT	Total Packets Transmitted	R	473
Statistics	040D8h	PTC64	Packets Transmitted (64 Bytes) Count	R	474
Statistics	040DCh	PTC127	Packets Transmitted (65-127 Bytes) Count	R	474
Statistics	040E0h	PTC255	Packets Transmitted (128-255 Bytes) Count	R	474
Statistics	040E4h	PTC511	Packets Transmitted (256-511 Bytes) Count	R	475

Table 13-3. (Continued) Ethernet Controller Register Summary

Category	Offset	Abbreviation	Name	R/W	Page
Statistics	040E8h	PTC1023	Packets Transmitted (512-1023 Bytes) Count	R	475
Statistics	040ECh	PTC1522	Packets Transmitted (1024-1522) Count	R	476
Statistics	040F0h	MPTC	Multicast Packets Transmitted Count	R	476
Statistics	040F4h	BPTC	Broadcast Packets Transmitted Count	R	476
Statistics	040F8h	TSCTC	TCP Segmentation Context Transmitted Count	R	477
Statistics	040FCh	TSCTFC	TCP Segmentation Context Tx Fail Count	R	477
Statistics	04100h	IAC	Interrupt Assertion Count (not applicable to the **631xESB/632xESB**)	R	478
Statistics	04104h	ICRXPTC	Interrupt Cause Rx Packet Timer Expire Count	R	478
Statistics	04108h	ICRXATC	Interrupt Cause Rx Absolute Timer Expire Count	R	478
Statistics	0410Ch	ICTXPTC	Interrupt Cause Tx Packet Timer Expire Count	R	478
Statistics	04110h	ICTXATC	Interrupt Cause Tx Absolute Timer Expire Count	R	479
Statistics	04118h	ICTXQEC	Interrupt Cause Tx Queue Empty Count	R	479
Statistics	0411Ch	ICTXDLTC	Interrupt Cause Transmit Descriptor Low Threshold Count	R	480
Statistics	04120h	ICRXDMTC	Interrupt Cause Rx Descriptor Minimum Threshold Count	R	480
Statistics	04124h	ICRXOC	Interrupt Cause Rx Overrun Count	R	480
Wakeup	05800h	WUC	Wakeup Control	R/W	426
Wakeup	05808h	WUFC	Wakeup Filter Control	R/W	427
Wakeup	05810h	WUS	Wakeup Status	R	428
Wakeup	05838h	IPAV	IP Address Valid	R/W	429
Wakeup	05840h-05858h	IP4AT	IPv4 Address Table	R/W	429
Wakeup	05880h-0588Fh	IP6AT	IPv6 Address Table	R/W	430
Wakeup	05900h	WUPL	Wakeup Packet Length	R/W	431
Wakeup	05A00h-05A7Ch	WUPM	Wakeup Packet Memory	R/W	431
Wakeup	05F00h-05F18h	FFLT	Flexible Filter Length Table	R/W	432

Table 13-3. (Continued) Ethernet Controller Register Summary

Category	Offset	Abbreviation	Name	R/W	Page
Wakeup	09000h-093F8h	FFMT	Flexible Filter Mask Table	R/W	433
Wakeup	09800h-09BF8h	FFVT	Flexible Filter Value Table	R/W	434
MNG	05820h	MANC	Management Control Register	R/W	434
PCIe*	05B00h	GCR	PCIe* Control	R/W	437
PCIe*	05B10h	GSCL_1	PCIe* statistics Control #1	R/W	439
PCIe*	05B14h	GSCL_2	PCIe* statistics Control #2	R/W	440
PCIe*	05B18h	GSCL_3	PCIe* statistics Control #3	R/W	444
PCIe*	05B1Ch	GSCL_4	PCIe* statistics Control #4	R/W	445
PCIe*	05B20h	GSCN_0	PCIe* Counter #0	R/W	446
PCIe*	05B24h	GSCN_1	PCIe* Counter #1	R/W	446
PCIe*	05B28h	GSCN_2	PCIe* Counter #2	R/W	446
PCIe*	05B2Ch	GSCN_3	PCIe* Counter #3	R/W	446
PCIe*	05B30h	FACTPS	Function Active and Power State to MNG	R/W	446
PCIe*	05B34h	GIOANACTL0	Serdes/CCM/PCIe* CSR	R/W	449
PCIe*	05B38h	GIOANACTL1	Serdes/CCM/PCIe* CSR	R/W	449
PCIe*	05B3Ch	GIOANACTL2	Serdes/CCM/PCIe* CSR	R/W	451
PCIe*	05B40h	GIOANACTL3	Serdes/CCM/PCIe* CSR	R/W	451
PCIe*	05B44h	GIOANACTLALL	Serdes/CCM/PCIe* CSR	R/W	452
PCIe*	05B48h	CCMCTL	Serdes/CCM/PCIe* CSR	R/W	452
PCIe*	05B4Ch	SCCTL	Serdes/CCM/PCIe* CSR	R/W	452
PCIe*	05B50h	SWSM	Software Semaphore	R/W	453
PCIe*	05B54h	FWSM	Firmware Semaphore	R/W	454

13.3 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.

13.3.1 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.

Table 13-4. CTRL Register Bit Description

Field	Bit(s)	Initial Value	Description
FD	0	1b	Full-Duplex Enables software to override the hardware Auto-Negotiation function. The FD sets the duplex mode only if CTRL.FRCDPLX is set. **Note:** Ethernet controllers DO NOT support half-duplex while in 1000 Mb/s mode. For the **82571EB/82572EI**: When operating in internal SerDes mode, and the AN Hardware is enabled, this bit is ignored. When operating in internal SerDes, and the AN Hardware is disabled, or the link is forced, this bit should be set by software. When operating in internal PHY mode, the FD bit is set by software based on AN and data rate resolution.
Reserved	1	0b	This bit is reserved and should be set to 0b.
GIO Master Disable	2	0b	When set to 1b, the function of this bit blocks new master requests including manageability requests. If no master requests are pending by this function, the *GIO Master Enable Status* bit is set.
LRST	3	1b	Link Reset 0b = Normal; 1b = Link Reset Applicable only in internal SerDes of operation. Used to reset the link control logic and restart the Auto-Negotiation process, when TXCW.ANE is set and internal SerDes is enabled. When set, transmission and reception are halted regardless of 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. This is a reserved bit for the **82573E/82573V/82573L**. Always set to 1b.

Field	Bit(s)	Initial Value	Description
Reserved	5:4	00b	Reserved Factory use only. Should be written with 00b.
SLU	6	0b	Set Link Up When the MAC link mode is set for 10/100/1000Base-T mode (internal/external PHY), Set Link Up must be set to 1b. to permit the MAC to recognize the LINK signal from the PHY, which indicates the PHY has established link, and to receive and transmit data The “Set Link Up” is normally initialized to 0b. However, if bit 10 of Word 14h/24h is set in the EEPROM/NVM then it is initialized to 1b. For the **82571EB/82572EI**: When the MAC link mode is set for TBI mode and if auto-negotiation is disabled (TXCW.ANE = 0b), setting this bit enables the MAC to communicate with the SerDes and enables recognition of the LOS signal. If auto-negotiation is enabled (TXCW.ANE = 1b) Set Link Up is ignored.
Reserved	7	0b	This bit is reserved and should be set to 0b.
SPEED	9:8	10b	Speed selection. These bits determine the speed configuration and are written by software after reading the PHY configuration through the MDIO interface (or GLCI interface for the **631xESB/632xESB**). These signals are ignored in internal Serdes mode (not applicable to the **82573E/82573V/82573L**). 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. For the **82571EB/82572EI**, FRCSPD is applicable only in internal PHY (copper) mode and is configurable through an EEPROM.
FRCDPLX	12	0b	Force Duplex When set to 1b, software can override the duplex indication from the PHY that is indicated in the FDX to the MAC. Otherwise, in 10/100/1000Base-T (internal/external PHY) link mode, the duplex setting is sampled from the PHY FDX indication into the MAC on the asserting edge of the PHY LINK signal. When asserted, the *CTRL.FD* bit sets duplex.
Reserved	14:13	00b	Reserved. Should be set to 00b.

Field	Bit(s)	Initial Value	Description
Reserved	15	0b	Reserved, should be set to 0b (the **631xESB/632xESB**)
Reserved	16	0b	For the **631xESB/632xESB**: Reserved. Must be set to 0b.
Reserved	17:15	0b	These are reserved bits for the **82571EB/82572EI** and the **82573E/82573V/82573L**. Read as 0b. Bit 17 is a reserved bit for the **631xESB/632xESB**.
SDP0_DATA	18	0b[1]	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. If bit 13 of this register is set, than this bit indicates the dock/undock state, and must be configured as input.
D/UD Status	18	0b	Dock/Undock Status (**82573E/82573V/82573L**) Indicates the dock/undock status of the platform. Valid only when the D/UD EN bit is set in the Extended Configuration Control register. The polarity of D/UD Status is determined by the D/UD polarity bit. This bit is pulled up when the pin is unconnected.
SDP1_DATA	19	0b[1]	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. This is a reserved bit for the (**82573E/82573V/82573L**).
ADVD3WUC	20	0b[1]	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.
Reserved	21	0b	This bit is reserved and should be set to 0b.
SDP0_IODIR	22	0b[1]	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. This is a reserved bit for the (**82573E/82573V/82573L**).
SDP1_IODIR	23	0b[1]	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. This is a reserved bit for the (**82573E/82573V/82573L**).
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. Software can reset the Ethernet controller by writing to this bit. The Ethernet controller re-reads the PER-FUNCTION EEPROM/NVM fields after a software reset. Bits that are normally read from the EEPROM/NVM are reset to their default hardware values. Note that this reset is PER FUNCTION and resets only the function that received the software reset. PCI Configuration space (configuration and mapping) of the Ethernet controller is unaffected.
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.
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.
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.
PHY_RST	31	0b	PHY Reset Controls a hardware-level reset to the external PHYs for the **631xESB/632xESB** and to the internal PHY(s) for the **82571EB/82572EI** and **82573E/82573V/82573L**. 0b = Normal operation. 1b = PHYs reset asserted. For the **631xESB/632xESB**, this reset is sent to the PHY in-band through the GLCI interface. Note that after reset, software should write 0b to exit the reset state.

1. If the signature bits of the EEPROM/NVM'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/NVM 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.

13.3.2 Device Status Register

STATUS (00008h; R)

This register provides software status indication about the Ethernet controller's settings and modes of operation.

Table 13-5. Status Register Bit Description

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 the **631xESB/632xESB** and **82573E/82573V/82573L**, this bit is only valid if the Set Link Up bit in the Device Control Register (CTRL.SLU) is set. For the **82571EB/82572EI**: While in 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. While in internal PHY mode, this bit reflects the status of the internal link signal indicating a transition to a Link Up.
LAN ID	3:2	0b	LAN ID. Provides software a mechanism to determine the Ethernet controller LAN identifier for this MAC. Read as: [0b,0b] LAN A, [0b,1b] LAN B.
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.

Field	Bit(s)	Initial Value	Description
TBIMODE	5	X	TBI Mode For the **631xESB/632xESB**, when this bit is asserted, indicates that the MAC is operating in SerDes mode. Based on CTRL_EXT.LINK_MODE. For the **82571EB/82572EI**: Internal SerDes Indication When set, the **82571EB/82572EI** operates in internal SerDes mode. When clear, the **82571EB/82572EI** operates in internal PHY mode. This is a reserved bit for the (**82573E/82573V/82573L**).
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 while in internal SerDes mode.
ASDV	9:8	X	Auto Speed Detection Value For the **631xESB/632xESB** and **82573E/82573V/82573L**, speed result sensed by the MAC auto-detection function. For the **82571EB/82572EI**: Indicates the speed sensed by the **82571EB/82572EI** 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 13.3.5 for details.
PHYRA	10	1b	PHY Reset Asserted Hardware sets this R/W bit following the assertion of a PHY reset. The bit is cleared on writing 0b to it and is used by firmware as an indication for the required initialization of the PHY.
Reserved	11	0b	Reserved. Should be set to 0b.
Reserved	18:12	0	These bits are reserved and should be set to 0b.
GIO Master Enable Status	19	1b	This bit is cleared when the *GIO Master Disable* bit is set and no master requests are pending by this function (otherwise, set). This bit indicates that no master requests are issued by this function as long as the *GIO Master Disable* bit is set.
Reserved	31:20	0b	Reserved Reads as 0b.

13.3.3 EEPROM/Flash Control Register

EEC (00010h; R/W)

This register provides a simplified interface for software accesses to the EEPROM/NVM. Software controls the EEPROM/NVM 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/NVM is latched into bit 3 of this register (via an internal 62.5 MHz clock for the **631xESB/632xESB**) and can be accessed by software through reads of this register.

Note: Attempts to write to the FLASH device when writes are disabled (FWE not equal to 10b) should not be attempted. Behavior after such an operation is undefined and can result in component and/or system hangs.

If LAN disable occurs during bit-banging, the EEPROM and the *EE_GNT* bit remain stuck at 0b. In this case, further accesses to EEPROM fails.

Table 13-6. EEC Register Bit Description

Field	Bit	Initial Value	Description
EE_SK	0	0b	Clock input to the EEPROM When EE_GNT = 1b, the EE_SK output signal is mapped to this bit and provides the serial clock input to the EEPROM. Software clocks the EEPROM via toggling this bit with successive writes.
EE_CS	1	0b	Chip select input to the EEPROM When EE_GNT = 1b, the EE_CS output signal is mapped to the chip select of the EEPROM device. Software enables the EEPROM by writing a 1b to this bit.
EE_DI	2	0b	Data input to the EEPROM When EE_GNT = 1b, the EE_DI output signal is mapped directly to this bit. Software provides data input to the EEPROM via writes to this bit.
EE_DO	3	X	Data output bit from the EEPROM The EE_DO input signal is mapped directly to this bit in the register and contains the EEPROM data output. This bit is RO from a 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 = Flash erase (along with bit 31 in the FLA register) 01b = Flash writes disabled 10b = Flash writes enabled 11b = Not allowed **Note:** For the **631xESB/632xESB**, values 01b and 10b are relevant for both serial and parallel flash accesses. A value of 00b is valid only for serial flash. The parallel flash can be erased using regular reads and writes. If FLA.SWACCEN (FLA[7]) = 0b, then theses bit are read only to software.
EE_REQ	6	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	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	1b	EEPROM Present This bit indicates that an EEPROM is present by monitoring the EE_DO input for an active-low acknowledge by the serial EEPROM during initial EEPROM scan. 1b = EEPROM present.
Auto_RD	9	0b	EEPROM Auto Read Done When set to 1b, this bit indicates that the auto read by hardware from the EEPROM is done. This bit is also set when the EEPROM is not present or when its signature is not valid.

Field	Bit	Initial Value	Description
EE_ADDR_SIZE	10	0b	EEPROM Address Size This field defines the address size of the EEPROM. 0b = 8- and 9-bit. 1b = 16-bit. This is a reserved bit for the (**82573E/82573V/82573L**).
EE_SIZE NVSize (**82573E/ 82573V/82573L**)	14:11[1]	0010b	EEEPROM Size This field defines the size of the EEPROM: Field Value / EEPROM Size / EEPROM Address Size 0000 / 128 bytes / 1 byte 0001 / 256 bytes / 1 byte 0010 / 512 bytes / 1 byte 0011 / 1 KB / 2 bytes 0100 / 2 KB / 2 bytes 0101 / 4 KB / 2 bytes 0110 / 8 KB / 2 bytes 0111 / 16 KB / 2 bytes 1000 / 32 KB / 2 bytes 1001 / 64 KB / 2 bytes 1011:1111 / Reserved / Reserved For the **82573E/82573V/82573L**: NVM Size This field defines the size of the NVM: This field defines the size of the NVM in bytes which equal 128 * 2 ** NVSize. This field is loaded from word 0Fh in the NVM and is read only.
NVADDS	16:15	0b	For the **82573E/82573V/82573L**: NVM Address Size. This field defines the Address size of the NVM: 00b = Reserved. 01b = EEPROM with 1 address byte. 10b = EEPROM with 2 address byte. 11b = FLASH with 3 address byte. This field is set at POR by the NVMT strapping pin. In case of an EEPROM the address length is set following a detection of the signature bits in word 12h in the EEPROM. If an EEPROM is attached to the **82573E/82573V/82573L** and valid signature is not found, software can modify this field enabling parallel access to an empty **82573E/82573V/82573L**. In any other cases writes to this field do not affect the **82573E/82573V/82573L** operation.
Reserved	19:17	0b	For the **82573E/82573V/82573L**, reserved. Reads as 0b.
AUPDEN	20	0b	Enable Autonomous FLASH Update Enables the **82573E/82573V/82573L** to update the FLASH autonomously. The autonomous update is triggered by write cycles and expiration of the FLASH timer.
Reserved	22:21	0b	For the **82573E/82573V/82573L**, reserved. Reads as 0b.

Field	Bit	Initial Value	Description
NVMTYPE	24:23	00b[1]	This RO field indicates the NVM type: 00b = EEPROM. 01b = Stand alone Flash. 10b = Shared SPI Flash. 11b = SIO. These bits are loaded from NVM word 0Fh and are informational only (the platform uses strapping to determine the actual NVM type).
Reserved	31:15 31:25[2]	0b	Reserved Reads as 0b.

1. These bits are red from the EEPROM/NVM.
2. **82573E/82573V/82573L** only.

13.3.4 EEPROM Read Register

EERD (00014h; RW)

Table 13-7. EEPROM Read Register Bit Description

Field	Bit(s)	Initial Value	Description
START	0	0b	Start Read Writing a 1b to this bit causes the EEPROM/NVM 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/NVM read completes. Set to 0b when the EEPROM/NVM 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/NVM 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: For the **82571EB/82572EI**, 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.

13.3.5 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).

Table 13-8. CTRL_EXT Register Bit Description

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0b	Reserved. Should be written as 0b to ensure future compatibility.
SDP2_GPIEN	2	0b	General Purpose Interrupt Detection Enable for SDP2 If software-controllable IO pin SDP2 is configured as an input, this bit (when set to 1b) enables use for GPI interrupt detection. This is a reserved bit for the (**82573E/82573V/82573L**).
SDP3_GPIEN	3	0b	General Purpose Interrupt Detection Enable for SDP3 If software-controllable IO pin SDP3 is configured as an input, this bit (when set to 1b) enables use for GPI interrupt detection. This is a reserved bit for the **82573E/82573V/82573L** and the **631xESB/632xESB**. Should be set to 0b.
Reserved	5:4	0b	Reserved. Reads as 0b.
SDP2_DATA	6	0b[1]	SDP2 Data Value. Used to read (write) the value of software-controllable IO pin SDP2. If SDP2 is configured as an output (SDP2_IODIR = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP2 is configured as an input, reads return the current value of the pin. This is a reserved bit for the (**82573E/82573V/82573L**).

Field	Bit(s)	Initial Value	Description
SDP3_DATA	7	0b[1]	SDP3 Data Value. Used to read (write) the value of software-controllable IO pin SDP3. If SDP3 is configured as an output (SDP3_IODIR = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP3 is configured as an input, reads return the current value of the pin. This is a reserved bit for the **82573E/82573V/82573L** and the **631xESB/632xESB**. Should be set to 0b.
Reserved	9:8	0b[1]	Reserved Should be written as 0b to ensure future compatibility.
SDP2_IODIR	10	0b[1]	SDP2 Pin Directionality. Controls whether software-controllable pin SDP2 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. This is a reserved bit for the (**82573E/82573V/82573L**).
SDP3_IODIR	11	0b[1]	SDP3 Pin Directionality. Controls whether software-controllable pin SDP3 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. This is a reserved bit for the **82573E/82573V/82573L** and the **631xESB/632xESB**. Should be set to 0b.
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. For the **82571EB/82572EI**, this functionality is provided for diagnostic purposes, regardless of whether the Auto Speed Detection feature is enabled. Applicable while in internal PHY mode only.
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	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.

Field	Bit(s)	Initial Value	Description
RO-DIS	17	0b	Relaxed Ordering Disabled When set to 1b, the Ethernet controller does not request any relaxed ordering transactions in PCIe* mode regardless of the state of bit 1 in the PCIe* command register. When this bit is cleared and bit 1 of the PCIe* command register is set, the Ethernet controller requests relaxed ordering transactions as described in Section 4.2.4.2.
SERDES Low Power Enable	18	0b[1]	When set, allows the SERDES to enter a low power state when the function is in Dr state as described in Section 6.1.7.5. This is a reserved bit for the (**82573E/82573V/82573L**).
DMA Dynamic Gating Enable	19	0b[1]	When set, enables dynamic clock gating of the DMA and MAC units. This is a reserved bit for the **631xESB/632xESB**. Must be set to 0b.
PHY Power Down Enable	20	1b[1]	When set, enable the PHY to enter a low-power state as described in Section 6.1.7.2.
Reserved	21	0b	Reserved. Should be set to 0b.
LINK_MODE	23:22	0b[1]	Link Mode. This controls which interface is used to talk to the link. 00b = Direct copper (1000Base-T) interface (10/100/1000Base-T internal PHY mode). 00b = GLCI mode (the **631xESB/632xESB** only). 01b = Reserved. 10b = Reserved. 11b = Internal SerDes (TBI) interface. 11b = SerDes mode (the **631xESB/632xESB** only).
Tx LS	23	0b[1]	For the **82573E/82573V/82573L**, should be set for correct Large Send functionality.
Tx LS Flow	22	0b[1]	For the **82573E/82573V/82573L**, should be cleared for the correct LSO functionality
PB_PAREN	24	0b	Packet Buffer Parity Error detection Enable (the **631xESB/632xESB** only). 1b = Parity errors in the packet buffer cause an error bit to be activated in the descriptor. CRC is also inverted in the case of Tx. 0b = Parity errors have no effect. **Note:** To avoid a false parity error indication at the beginning of operation, the entire space in offsets 10000h:1BFFFh should be written with a value of 0b before setting this bit. Reserved for the **82571EB/82572EI**.
DF_PAREN	25	0b	Descriptor FIFO Parity Error detection Enable (the **631xESB/632xESB** only). 1b = Parity errors in the descriptor FIFO cause a corresponding DMA operation to stop. 0b = Parity errors have no effect. Reserved for the **82571EB/82572EI**.
Reserved	26	0b	Reserved, should be set to 0b.

Field	Bit(s)	Initial Value	Description
IAME	27	0b	Interrupt Acknowledge Auto-Mask Enable When set to 1b, a read or write to the ICR register has the side effect of writing the value in the IAM register to the IMC register. When set to 0b, this feature is disabled.
DRV_LOAD	28	0b	Driver Loaded This bit should be set by the driver after it loaded. This bit should be cleared when the driver unloads or after a PCIe* soft reset. The MNG controller loads this bit to indicate to the manageability controller that the driver has loaded. This is a reserved bit for the (**82573E/82573V/82573L**).
INT_TIMERS_ CLEAR_ENA	29	0b	When set this bit enables the clear of the interrupt timers following an IMS clear. In this state, successive interrupts happen only after the timers expire again. When clear, successive interrupts following IMS clear might happen immediately. Reserved for the **82571EB/82572EI**.
HOST_PAREN	30	0b	Host Data FIFO Parity Enable (the **631xESB/632xESB** only). 1 = Parity errors in the descriptor FIFO are detected. 0b = Parity errors not detected. In addition, when set to 0b, the partial write into this memory is enabled (which might cause false parity errors). Reserved for the **82571EB/82572EI**.
Reserved	31	0b	Reserved. Reads as 0b.

1. These bits are read from the EEPROM/NVM.

The **82571EB/82572EI** allows for up to two externally controlled interrupts. The upper two software-definable pins, SDP[3:2], 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 **631xESB/632xESB** allows for one externally controlled interrupt. This software-definable pin, SDP[2], can be mapped for use as a GPI interrupt bit. This mapping is enabled by the SDP2_GPIEN bits only when this signal is also configured as an input via SDP2_IODIR. When configured to function as an external interrupt pin, a GPI interrupt is generated when the corresponding pin is sampled in an active-high state.

The bit mappings are shown in Table 13-9 for clarity.

Table 13-9. 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	
3[1]	SDP3_IODIR	SDP3_GPIEN	14
2	SDP2_IODIR	SDP2_GPIEN	13

1. **82571EB/82572EI** only.

Note: If software uses the EE_RST function and desires to retain the current configuration information, the contents of the control registers should be read and stored by software. Control register values are changed by a read of the EEPROM which occurs upon assertion of the EE_RST bit.

The EEPROM reset function can read configuration information out of the EEPROM which affects the configuration of PCIe* configuration space BAR settings. The changes to the BARs are not visible unless the system is rebooted and the BIOS is allowed to re-map them.

The *SPD_BYPS* bit performs a similar function to the CTRL.FRCSPD bit in that the Ethernet controller's speed settings are determined by the value software writes to the *CRTL.SPEED* bits. However, with the *SPD_BYPS* bit asserted, the settings in CTRL.SPEED take effect rather than waiting until after the Ethernet controller's clock switching circuitry performs the change.

For the **631xESB/632xESB**, this register includes a PB_ PAREN bit that enables parity error processing in the packet buffer memory.

13.3.6 Flash Access

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 (FL_NVM_SK bit 0 for the **82573E/82573V/82573L**) 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.

Note: The Ethernet controller FLA register is only reset at LAN_PWR_GOOD and not as legacy devices at a software reset.

Table 13-10. Flash Access - FLA

Field	Bit(s)	Initial Value	Description
FL_SCK FL_NVM_SK (**82573E/ 82573V/ 82573L**)	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. For the **82573E/82573V/82573L**, when FL_GNT is 1b, the FL_NVM_SK output signal is mapped to this bit and provides the serial clock input to the Flash. Software clocks the Flash by toggling this bit with successive writes.
FL_CE	1	0b	Chip Select Input to the FLASH. When FL_GNT is 1b, the FL_CE output signal is mapped to the chip select of the FLASH 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_SI output signal is mapped directly to this bit. Software provides data input to the FLASH via writes to this bit.

Field	Bit(s)	Initial Value	Description
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	0b	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_DO bits.
FLA_add_size	6	0b	FLASH Address Size. When Flash_add_size is set, all flashes (including 64 KB) are accessed using 3 bytes of the address. This is a reserved bit for the **82573E/82573V/82573L**.
Reserved	8:7	0b	Reserved. Reads as 0b.
SW_WR_ DONE	9	1b	For the **82573E/82573V/82573L**: Status Bit Indicates that last LAN_BAR, LAN_EXP, IDE or KCS write was done.
RD_STATUS	10	1b	For the **82573E/82573V/82573L**: Flash Status Updated by software or firmware. If set to 0b' the WEL bit in the status register is read, else the BUSY bit.
Reserved	15:9	0b	Reserved. Reads as 0b.
IDE Boot Offset	29:16	0b	Defines the Base address of the IDE Boot expansion ROM in the physical FLASH device. The base address (in bytes) equals 256 x the field value. The field is loaded from EEPROM word 48h/108h. The driver does not have to preserve the value of this register since it is used only after it is loaded from the EEPROM. However, Boot LAN software should not affect this field since Boot IDE software might be executed afterwards. For the **82571EB/82572EI**, software cannot write to this field. These are reserved bits for the **82573E/82573V/82573L**.
FL_BUSY	30	0b	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	0b	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.

13.3.7 MDI Control Register

MDIC (00020h; R/W)

Software uses this register to read or write Management Data Interface (MDI) registers in the PHY.

Note: For the **631xESB/632xESB**, when enabled, this MDIO communication sequence is transmitted through the in-band GLCI signal.

For an MDI read cycle, the sequence of events is as follows:

- The CPU performs a PCIe* write cycle to the MII register with:
  - — Ready = 0b
  - — Interrupt Enable set to 1b or 0b
  - — Opcode = 10b (read)
  - — PHYADD = PHY address from the MDI register
  - — REGADD = Register address of the specific register to be accessed (0 through 31)
- The MAC applies the following sequence on the MDIO signal to the PHY:

  <PREAMBLE><01><10><PHYADD><REGADD><Z> where Z stands for the MAC tri-stating the MDIO signal
- The PHY returns the following sequence on the MDIO signal:

  <0><DATA><IDLE>
- The MAC discards the leading bit and places the following 16 data bits in the MII register
- The MAC asserts an interrupt indicating MDI “Done” if the *Interrupt Enable* bit was set
- The CPU reads the data from the MII register and then issues a new MDI command

For a MDI write cycle, the sequence of events is as follows:

- Ready = 0b
- Interrupt Enable set to 1b or 0b
- Opcode = 01b (write)
- PHYADD = PHY address from the MDI register
- REGADD = Register address of the specific register to be accessed (0 through 31)
- Data = Specific data for desired control of the PHY
- The MAC applies the following sequence on the MDIO signal to the PHY:

  <PREAMBLE><01><01><PHYADD><REGADD><10><DATA><IDLE>
- The MAC asserts an interrupt indicating MDI “Done” if the *Interrupt Enable* bit was set
- The MAC sets the *Ready* bit in the MII register to indicate that the write operation completed
- The CPU might issue a new MDI command

Note: An MDI read or write might take as long as 64 µs from the CPU write to the *Ready* bit assertion.

If an invalid opcode is written by software, the MAC does not execute any accesses to the PHY registers.

If the PHY does not generate a 0b as the second bit of the turn-around cycle for reads, the MAC aborts the access, sets the *E* (error) bit, writes FFFFh to the data field to indicate an error condition, and sets the *Ready* bit.

Note: This note **only** applies if address 26 needs to be written to. For the **631xESB/632xESB**, due to a legacy blocking feature in the **82571EB/82572EI**, address 26 should be written prior to address 31. Failure to do so might result with a failure to write into address 26.

Also, the *R* (ready) bit in MDIC register is asserted when the state machine becomes idle, while the last bit of the transaction on MDC/MDIO external line is still being transmitted. Care must be taken that a new write to the MDIC is not performed before the line is ready.

Note: See Section 13.3.9 for detailed information about register access for the **82563EB/82564EB**.

Note: The PHY register bit descriptions follow Table 13-11.

Table 13-11. MDI Control Register Bit Description

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. For the **82573E/82573V/82573L**: 1 = Gigabit PHY. 2 = PCIe* PHY. This is a reserved bit for the **631xESB/632xESB**.
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.

Field	Bit(s)	Initial Value	Description
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 Write as 0b for future compatibility.

13.3.8 PHY Registers

This document uses a special nomenclature to define the read/write mode of individual bits in each register. See Table 13-12.

For all binary equations appearing in the register map, the symbol "|" is equivalent to a binary OR operation.

Table 13-12. 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.

13.3.8.1 PHY Control Register

PCTRL (00d; R/W)

Table 13-13. 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.	RW	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. **Note:** If auto-negotiation is enabled, this bit is ignored.	R/W	1b	Update
Collision Test	7	1b = Enable COL signal test. 0b = Disable COL signal test. **Note:** This bit is ignored unless loopback is enabled (bit 14 = 1b).	R/W	0b	0b
Duplex Mode	8	1b = Full Duplex. 0b = Half Duplex. **Note:** If auto-negotiation is enabled, this bit is ignored.	R/W	1b	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.	WO, SC	0b	Self Clear
Isolate	10	1b = Isolate. 0b = Normal operation.	R/W	0b	0b
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.	R/W	0b	0b

Table 13-13. 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 not supported.	R/W	1b	Update
Speed Selection (LSB)	13	See Speed Selection (MSB), bit 6. **Note:** If auto-negotiation is enabled, this bit is ignored.	R/W	1b[1]	Update
Loopback	14	1b = Enable loopback. 0b = Disable loopback.	R/W	0b	0b
Reset	15	1b = PHY reset. 0b = Normal operation.	WO, SC	0b	Self Clear

1. 0b for the **82573E/82573V/82573L** and **82563EB/82564EB**.

13.3.8.2 PHY Status Register

PSTATUS (01d; R)

Table 13-14. 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	

Table 13-14. PHY Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
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	0b = PHY does not accept management frames with preamble suppressed.	RO	Always 0b	
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.	RO	1b	
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.	RO	1b	
100BASE-X Half Duplex	13	1b = PHY able to perform half duplex 100BASE-X. 0b = PHY able to perform half duplex 100BASE-X.	RO	1b	
100BASE-X Full Duplex	14	1b = PHY able to perform full duplex 100BASE-X. 0b = PHY not able to perform full duplex 100BASE-X.	RO	1b	
100BASE-T4	15	0b = PHY not able to perform 100BASE-T4. 1b = PHY able to perform 100BASE-T4.	RO	Always 0b	

13.3.8.3 PHY Identifier Register (LSB)

PHY ID (02d; R)

Table 13-15. PHY Identifier Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
PHY ID Number	15:0	The PHY identifier composed of bits 3 through 18 of the Organizationally Unique Identifier (OUI)	RO	Always 02A8h[1]	

1. 0141h for the **82573E/82573V/82573L** and **82563EB/82564EB**.

13.3.8.4 Extended PHY Identifier Register (MSB)

Extended PHY ID (03d; R)

Table 13-16. Extended PHY Identifier Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Manufacturer's Revision Number	3:0	4 bits containing the manufacturer's revision number.	RO	0h	00h
Manufacturer's Model Number	9:4	6 bits containing the manufacturer's part number.	RO	38h (0Ch for the **82573E/ 82573V/ 82573L**)	00h
PHY ID Number	15:10	The PHY identifier composed of bits 19 through 24 of the OUI	RO	00h (03h for the **82573E/ 82573V/ 82573L**)	00h

13.3.8.5 82563EB/82564EB PHY Identifier Register (MSB)

PHY ID (03d; R)

Table 13-17. PHY Identifier Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
PHY Identification	15:0		RO	0CA0h	

13.3.8.6 Auto-Negotiation Advertisement Register

ANA (04d; R/W)

Table 13-18. Auto-Negotiation Advertisement Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Selector Field	4:0	00001b = 802.3 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.	R/W	Always 00001b	
10Base-T[1]	5	1b = DTE is 10BASE-T capable. 0b = DTE is not 10BASE-T capable.	R/W	1b	Retain
10Base-T Full Duplex	6	1b = DTE is 10BASE-T full duplex capable. 0b = DTE is not 10BASE-T full duplex capable.	R/W	1b	Retain
100Base-TX	7	1b = DTE is 100BASE-TX capable. 0b = DTE is not 100BASE-TX capable.	R/W	1b	Retain
100BASE-TX Full Duplex	8	1b = DTE is 100BASE-TX full duplex capable. 0b = DTE is not 100BASE-TX full duplex capable.	R/W	1b	Retain
100BASE-T4	9	0b = Not capable of 100BASE-T4.	R/W	Always 0b	
PAUSE	10	Advertise to Partner that Pause operation (as defined in 802.3x) is desired.	R/W	1b	Retain
ASM_DIR	11	Advertise Asymmetric Pause direction bit. This bit is used in conjunction with PAUSE.	R/W	1b	Retain
Reserved	12	Ignore on read.	R/W	0b	Retain
Remote Fault	13	1b = Set Remote Fault bit. 0b = Do not set Remote Fault bit.	R/W	0b	Retain

Table 13-18. Auto-Negotiation Advertisement Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	14	Always read as 0b. Write to 0b for normal operation.	R/W	Always 0b	
Next Page	15	1b = Manual control of Next Page (Software). 0b = Ethernet controller control of Next Page (Auto).	R/W	0b	Retain

1. 10BASE-TX for the **82573E/82573V/82573L**.

13.3.8.7 Auto-Negotiation Base Page Ability Register (05d; R)

Table 13-19. Auto-Negotiation Base Page Ability Bit Description[1]

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Selector Fields[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 04d, 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[2]	10	Link Partner uses Pause Operation as defined in 802.3x.	RO	N/A	

Table 13-19. Auto-Negotiation Base Page Ability Bit Description[1]

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
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	
Technology Ability Field	12	For the **82573E/82573V/82573L**, this bit is cleared when the link goes down and loaded when a base page is received. Reserved for all other Ethernet controllers. 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. For the **82573E/82573V/82573L**, this bit is cleared when the link goes down and loaded when a base page receives Acknowledge.	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 08d stores the Auto-Negotiation Link Partner Received Next Pages. PHY register 05d is not used to store Next Pages. It contains the information from the last Base Page correctly received.
2. Pause Capable for the **82573E/82573V/82573L**.

13.3.8.8 Auto-Negotiation Expansion Register

ANE (06d; R)

Table 13-20. 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	Indicates that a new page has been received and the received code word has been loaded into PHY register 05d (base pages) or PHY register 08d (next pages) as specified in clause 28 of 802.3. This bit clears on read. If PHY register 16d bit 1 (Alternate NP Feature) is set, the *Page Received* bit also clears when mr_page_rx = false or transmit_disable = true.	RO/ LH	0b	0b

Table 13-20. Auto-Negotiation Expansion Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Next Page Able	2	1b = Local device is next page able. 0b = Local device is not next page able.	RO	1b	
Link Partner Next Page Able	3	1b = Link Partner is next page able. 0b = Link Partner is not next page able.	RO	0b	0b
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	0b	0b
Base Page	5	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 16d, bit 1 (Alternate NP Feature) is set. 1b = base_page = true. 0b = base_page = false. Not applicable to the **82573E/82573V/ 82573L** or the **82563EB/82564EB**.	RO/ LH	0b	0b
Reserved	15:6	Always read as 0b.	RO	Always 0b	

13.3.8.9 Auto-Negotiation Next Page Transmit Register NPT (07d; R/W)

Table 13-21. Auto-Negotiation Next Page Transmit Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Message/ Unformatted Field	10:0	11-bit message code field.	R/W	1b	1b
Toggle	11	1b = Previous value of the transmitted Link Code Word = 0b. 0b = Previous value of the transmitted Link Code Word = 1b.	RO	0b	0b
Acknowledge 2	12	1b = Complies with message. 0b = Cannot comply with message.	R/W	0b	0b
Message Page	13	1b = Message page. 0b = Unformatted page.	R/W	1b	1b
Reserved	14	Always read as 0b. Write to 0b for normal operation.	RO	0b	0b
Next Page	15	1b = Additional next pages follow. 0b = Last page.	R/W	0b	0b

13.3.8.10 Auto-Negotiation Next Page Ability Register

LPN (08d; R)

Table 13-22. Auto-Negotiation Next Page Ability Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Message/ Unformatted Field	10:0	11-bit message code field.	RO	0b	0b
Toggle	11	1b = Previous value of the transmitted Link Code Word = 0b. 0b = Previous value of the transmitted Link Code Word = 1b.	RO	0b	0b
Acknowledge 2	12	1b = Link Partner complies with the message. 0b = Link Partner cannot comply with the message.	RO	0b	0b
Message Page	13	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
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. This is a reserved bit for the **82563EB/ 82564EB**. Always read as 0b. Write to 0b for normal operation.	RO	0b	0b
Next Page	15	1b = Link Partner has additional next pages to send. 0b = Link Partner has no additional next pages to send.	RO	0b	0b

13.3.8.11 1000BASE-T/100BASE-T2 Control Register

GCON (09d; R/W)

Table 13-23. 1000BASE-T/100BASE-T2 Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	7:0	Always read as 0b. Write to 0b for normal operation.	R/W	0b	0b
1000BASE-T Half Duplex	8	1b = DTE is 1000BASE-T capable. 0b = DTE is not 1000BASE-T capable. This bit is used by Smart Negotiation. **Note:** For the **82563EB/82564EB**, software MUST NOT program this bit to 1b.	R/W	0b	Retain

Table 13-23. 1000BASE-T/100BASE-T2 Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
1000BASE-T Full Duplex	9	1b = DTE is 1000BASE-T full duplex capable. 0b = DTE is not 1000BASE-T full duplex capable. This bit is used by Smart Negotiation.	R/W	1b	Retain
Port Type	10	1b = Prefer multi-port device (Master)[1]. 0b = Prefer single port device (Slave)[1].	R/W	0b	Retain
Master/Slave Configuration Value	11	1b = Manual configure as Master[2]. 0b = Manual configure as Slave[2].	R/W	0b	Retain
Master/Slave Manual Configuration Enable	12	1b = Manual Master/Slave configuration. 0b = Automatic Master/Slave configuration.	R/W	0b	Retain
Test mode	15:13	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. Only when PHY register 09d, bit 12 is set to 0b.
2. Only when PHY register 09d, bit 12 is set to 1b.

13.3.8.12 1000BASE-T/100BASE-T2 Status Register

GSTATUS (10d; R)

Table 13-24. 1000BASE-T/100BASE-T2 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, LH	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

Table 13-24. 1000BASE-T/100BASE-T2 Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
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
Local Receiver Status	13	1b = Local Receiver OK. 0b = Local Receiver Not OK.	RO	0b	0b
Master/Slave Resolution	14	1b = Local PHY configuration resolved to Master. 0b = Local PHY configuration resolved to Slave. Values in bits 11:10 are not valid until the ANE Register Page Received bit equals 1b.	RO	0b	0b
Master/Slave Configuration Fault	15	1b = Master/Slave configuration fault detected. 0b = No Master/Slave configuration fault detected.	RO, LH	0b	0b

13.3.8.13 Extended Status Register

ESTATUS (15d; R)

Table 13-25. Extended Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	11:0	Reserved. Always read as 0b.	RO	0b	0b
1000BASE-T Half Duplex	12	1b = 1000BASE-T half duplex capable. 0b = not 1000BASE-T half duplex capable.	RO	1b	1b
1000BASE-T Full Duplex	13	1b = 1000BASE-T full duplex capable. 0b = Not 1000BASE-T full duplex capable.	RO	1b	1b
1000BASE-X Half Duplex	14	1b =1000BASE-X half duplex capable. 0b = Not 1000BASE-X half duplex capable.	RO	0b	0b
1000BASE-X Full Duplex	15	1b =1000BASE-X full duplex capable. 0b = Not 1000BASE-X full duplex capable.	RO	0b	0b

13.3.8.14 Port Configuration Register (82571EB/82572EI)

PCONF (16d; R/W)

Table 13-26. 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
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
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

Table 13-26. Port Configuration Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
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	15	Always read as 0b. Write 0b for normal operation.	R/W	0b	0b

13.3.8.15 PHY Specific Control Register

PCON (16d; R/W)

Table 13-27. PHY Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Disable Jabber	0	Jabber has effect only in 10BASE-T half-duplex mode. 1b = Disable jabber function. 0b = Enable jabber function.	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 Alternate Power Down (**82563EB/ 82564EB**)	2	SQE test is automatically disabled in full-duplex mode regardless of the state of bit 2. 1b = SQE test enabled. 0b = SQE test disabled. For the **82563EB/82564EB**, power down is controlled via PHY register 0 bit 11 and this bit. Both bits must be set to 0b before the PHY transitions from power down to normal operation. When the port is switched from power down to normal operation, a software reset and restart Auto-Negotiation are performed even when bits Reset (PHY register 0 bit 15) and Restart Auto-Negotiation (PHY register 0 bit 9) are not set by the programmer. 1b = Power down. 0b = Normal operation.	R/W	0b	Retain

Table 13-27. PHY Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
MAC Interface Power Down Cooper Transmitter Disable (**82563EB/ 82564EB**)	3	Changes to this bit are disruptive during the normal operation. As a result, any changes to these registers must be followed by a software reset to take effect. This bit determines whether the MAC interface powers down when PHY Control register bit 11 is used to power down the Ethernet controller or when the PHY enters the energy detect state. 1b = Always power up. 0b = Can power down. For the **82563EB/82564EB**: 1b = Transmitter disable. 0b = Transmitter enable.	R/W	1b 0b for the **82563EB/ 82564EB**	Update Retain for the **82563EB/ 82564EB**
Disable 125CLK	4	Changes to this bit are disruptive during the normal operation. As a result, any changes to these registers must be followed by a software reset to take effect. 1b = 125CLK low. 0b = 125CLK toggle. This is a reserved bit for the **82563EB/ 82564EB**.	R/W	0b	Update (Retain for the **82563EB/ 82564EB**)
MDI Crossover Mode	6:5	Changes to this bit are disruptive during the normal operation. As a result, any changes to these registers must be followed by a software reset to take effect. 00b = Manual MDI configuration. 01b = Manual MDIX configuration. 10b = Reserved. 11b = Enable automatic crossover for all modes.	R/W	03h	Update
Enable Extended Distance	7	When using a cable exceeding 100 m, the 10BASE-T receive threshold must be lowered in order to detect incoming signals. 1b = Lower 10BASE-T receive threshold. 0b = Normal 10BASE-T receive threshold.	R/W	0b	Retain
Energy Detect	9:8	0xb = Off. 10b = Sense only on Receive (Energy Detect). 1b = Sense and periodically transmit NLP (Energy Detect*).	R/W	11b	Retain

Table 13-27. PHY Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Force Link Good	10	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. 1b = Force link good 0b = Normal operation	R/W	0b	Retain
Assert CRS on Transmit Downshift Enable (**82563EB/82564EB**)	11	This bit has no effect in full duplex mode. 1b = Assert on transmit. 0b = Never assert on transmit. For the **82563EB/82564EB**, changes to this bit are disruptive during the normal operation. As a result, any changes to these registers must be followed by a software reset to take effect. 1b = Enable downshift. 0b = Disable downshift.	R/W	0b (1b for the **82563EB/ 82564EB**)	Retain (Update for the **82563EB/ 82564EB**)
Reserved	13:12	Reserved	R/W	0b	Retain
Downshift Counter[1]	14:12	Changes to these bits are disruptive during normal operation. As a result, any changes to these registers must be followed by software reset to take effect. 1x, 2x, ...8x is the number of times the PHY attempts to establish gigabit link before the PHY downshifts to the next highest speed. 000b = 1x - 100b = 5x 001b = 2x - 101b = 6x 010b = 3x - 110b = 7x 011b = 4x - 111b = 8x.	R/W	010b	Update
Transmit FIFO Depth	15:14	1000BASE-T 10/100BASE-T 00b = +/- 16 Bits 00b = +/- 8 Bits 01b = +/- 24 Bits 01b = +/- 12 Bits 10b = +/- 32 Bits 10b = +/- 16 Bits 11b = +/- 40 Bits 11b = +/- 20 Bits	R/W	0b	Retain
Disable Link Pulses[1]	15	1b = Disable link pulse. 0b = Enable link pulse.	R/W	0b	0b

1. **82563EB/82564EB** only.

13.3.8.16 External MDIO Control Register (82563EB/82564EB) (Page 192, 16d; R/W)

Table 13-28. External MDIO Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
external_mdio	0	Select access method for all of a port's registers other than page 192 registers: 0b = GLCI in-band access. 1b = External MDIO access. One is enabled at a time.	R/W	0b	0b
Reserved	15:1	Reserved, write as 0b.	R/W	0b	0b

13.3.8.17 GLCI Mode Control Register (82563EB/82564EB) (Page 193, 16d; R/W)

Table 13-29. External MDIO Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	1:0	Reserved, write as 0b.	R/W	0b	0b
Force Duplex	2	Use the Programmed Duplex value instead of the duplex indication from the PHY.	R/W	0b	0b
Programmed Duplex	3	Programmed duplex value, used only when Force Duplex is 1b. 0b = Half Duplex. 1b = Full Duplex.	R/W	0b	0b
Reserved	4	Reserved, write as 0b.	R/W	0b	0b
phy_leds_en	5	Select between PHY and MAC LEDs. 0b = GLCI inband LEDs from MAC. 1b = PHY LEDs.	R/W	0b	0b
Force Link Up	6	Indicate link up to the MAC even if the PHY is indicating link down (The PHY may indicate link down in loopback modes).	R/W	0b	0b
Reserved	7	Suppress the IEEE clause 36 Receive PCS function of placing a special End of Packet extension symbol (/R/) in the 1000Base-T data stream at the end of the packet going from the MAC to the PHY.	R/W	1b	1b
Reserved	15:11	Reserved, write as read.	R/W	01h	01h

13.3.8.18 GLCI FIFO's Control/Status Register (82563EB/82564EB) (Page 194, 16d; R/W)

Table 13-30. GLCI FIFO's Control/Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
TX FIFO Overflow	0	TX FIFO overflow occurred.	RO, LH	0b	0b
TX FIFO Underflow	1	TX FIFO underflow occurred.	RO, LH	0b	0b
Reserved	7:2	Reserved, read only.	RO	010100b	010100b
RX FIFO Overflow	8	RX FIFO overflow occurred.	RO, LH	0b	0b
RX FIFO Underflow	9	RX FIFO underflow occurred.	RO, LH	0b	0b
Reserved	15:10	Reserved, read only.	RO	000100b	000100b

13.3.8.19 Port Status 1 Register (82571EB/82572EI) PSTAT (17d; RO)

Table 13-31. Port 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 17d, bit 10 = 1b). Auto-Negotiation is disabled (PHY register 00d, bit 12 = 0b). Auto-Negotiation is restarted (PHY register 00d, 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

Table 13-31. Port Status 1 Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
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 01, bit 2 in that this bit changes anytime the link status changes. PHY register 01, 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 internal loopback, this bit reads as 0b.	RO	0b	0b
Transmit Status	13	1b = PHY currently transmitting a packet. 0b = PHY transmitter is IDLE. When in internal loopback, this bit reads as 0b.	RO	00b	00b
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

13.3.8.20 PHY Specific Status Register[1]

PSSTAT (17d; RO)

Table 13-32. PHY Specific Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Jabber (Real Time)	0	1b = Jabber. 0b = No jabber.	RO	0b	Retain (0b for the **82563EB/ 82564EB**)
Polarity (Real Time)	1	1b = Reversed. 0b = Normal. For the **82563EB/82564EB**, the detected polarity status is shown in PHY register 0, page 17, bit 1 or in 1000BASE-T mode (PHY register 5, page 21, bits 3:0).	RO	0b	0b
Receive Pause Enabled	2	This is a reflection of the MAC pause resolution. This bit is for information purposes and is not used by the Ethernet controller. This status bit is valid only after the *Speed and Duplex Resolved* bit (11) = 1b. The *Speed and Duplex Resolved* bit is set when Auto-Negotiation has completed or Auto-Negotiation is disabled. 1b = Receive pause enabled. 0b = Receive pause disabled. This is a reserved bit for the **82563EB/ 82564EB**.	RO	0b	0b
Transmit Pause Enabled Global Link Status (**82563EB/82564EB)**	3	This is a reflection of the MAC pause resolution. This bit is for information purposes and is not used by the Ethernet controller. This status bit is valid only after the *Speed and Duplex Resolved* bit (11) = 1b. The *Speed and Duplex Resolved* bit is set when Auto-Negotiation has completed or Auto-Negotiation is disabled. 1b = Transmit pause enabled. 0b = Transmit pause disabled. For the **82563EB/82564EB**: 1b = Copper link is up. 0b = Cooper link is down.	RO	0b	0b

1. PHY Specific Status Register 1 for the **82563EB/82564EB**.

Table 13-32. PHY Specific Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Copper Energy Detect Status	4	1b = Sleep. 0b = Active. For the **82563EB/82564EB**: 1b = No energy detected. 0b = Energy detected.	RO	1b (0b for the **82563EB/ 82564EB)**	1b (0b for the **82563EB/ 82564EB)**
Downshift Status	5	1b = Downshift. 0b = No downshift.			
MDI Crossover Status	6	This status bit is valid only after the *Speed and Duplex Resolved* bit (11) = 1b. The *Speed and Duplex Resolved* bit is set when Auto-Negotiation has completed or Auto-Negotiation is disabled. This bit is 0b or 1b depending on what is written to PHY register 16 bits 6:5 in manual configuration mode. PHY register 16 bits 6:5 are updated with software reset. 1b = MDIX 0b = MDI	RO	0b	Retain
Reserved[1]	7	Reserved.	RO	0b	0b
Receive Pause Enabled[1]	8	This is a reflection of the MAC pause resolution. This bit is for information purposes and is not used by the Ethernet controller. This status bit is valid only after the *Speed and Duplex Resolved* bit (11) = 1b. The *Speed and Duplex Resolved* bit is set when Auto-Negotiation has completed or Auto-Negotiation is disabled. 1b = Receive pause enabled. 0b = Receive pause disabled.			
Transmit Pause Enabled[1]	9	This is a reflection of the MAC pause resolution. This bit is for information purposes and is not used by the Ethernet controller. This status bit is valid only after the *Speed and Duplex Resolved* bit (11) = 1b. The *Speed and Duplex Resolved* bit is set when Auto-Negotiation has completed or Auto-Negotiation is disabled. 1b = Transmit pause enabled. 0b = Transmit pause disabled.			

Table 13-32. PHY Specific Status Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Cable Length 1000 mode only)	9:7	Cable length measurement is only a rough estimate. Actual value depends on the attenuation of the cable, output levels of the remote transceiver, connector impedance, etc. 000b = < 50 m. 001b = 50 - 80 m. 010b = 80 - 110 m. 011b = 110 - 140 m. 100b = greater than 140 m.	RO	0b	0b
Link (Real Time)	10	1b =Link up. 0b = Link down.	RO	0b	0b
Speed and Duplex Resolved	11	When Auto-Negotiation is not enabled, PHY register 17 bit 11 = 1b. 1b = Resolved. 0b = Not resolved.	RO	0b	0b
Page Received	12	1b = Page received. 0b = Page not received.	RO, LH	0b	0b
Duplex	13	This status bit is valid only after the *Speed and Duplex Resolved* bit (11) = 1b. The *Speed and Duplex Resolved* bit is set when Auto-Negotiation has completed or Auto-Negotiation is disabled. 1 = Full-duplex. 0 = Half-duplex.	RO	0b	Retain
Speed	15:14	This status bit is valid only after the *Speed and Duplex Resolved* bit (11) = 1b. The *Speed and Duplex Resolved* bit is set when Auto-Negotiation has completed or Auto-Negotiation is disabled. 11b = Reserved. 10b = 1000 Mb/s. 01b = 100 Mb/s. 00 = 10 Mb/s. For the **82571EB/82572EI**: 11b = 1000 Mb/s. 10b = 100 Mb/s. 01b = 10 Mb/s. 00 = Reserved.	RO	0b	Retain

1. **82563EB/82564EB** only.

13.3.8.21 Port Reset Register (82563EB/82564EB)

(Page 193, 17d; R/W)

Table 13-33. Port Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Port Reset	0	Port reset bit. The Port reset bit resets (hard) the entire port, including this bit.	R/W	0b	0b
Reserved	15:1	Reserved, write as 0b.	R/W	0b	0b

13.3.8.22 Transmit Amplitude 10/100 Register (82563EB/82564EB)

(Page 255, 17d; R/W)

Table 13-34. Transmit Amplitude 10/100 Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
10BASE-T Transmit Amplitude MDI[0]	3:0	Same encoding as 100BASE-T transmit amplitude MDI[1]. **Note:** Set these bits to 0001b.	R/W	0b	Retain
10BASE-T Transmit Amplitude MDI[1]	7:4	Same encoding as 100BASE-T transmit amplitude MDI[1].	R/W	0b	Retain
100BASE-T Transmit Amplitude MDI[0]	11:8	Same encoding as 100BASE-T transmit amplitude MDI[1].	R/W	0b	Retain
100BASE-T Transmit Amplitude MDI[1]	15:9	0000b = 0%, 0001b = -2%, 0010b = -4%, 0011b = -6% 0100b = -8%, 0101b = -10%, 0110b = -12%, 0111b = -14% 1000b = +0%, 1001b = +2%, 1010b = +4%, 1011b = +6% 1100b = +8%,1101b = +10%, 1110b = +12%,1111b = +14%	R/W	0b	Retain

13.3.8.23 Port Control Register (82571EB/82572EI)

PCONT (18d; R/W)

Table 13-35. 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
Extend SPD Delay	5	When set, extends the delay of a power down if the Ethernet cable is disconnected. 0b = Wait four seconds before beginning power down. 1b = Wait 6.3 seconds before beginning power down.	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
TEN_CRS_Select	10	1b = Extend CRS to cover 1000Base-T latency and RX_DV. 0b = Do not extend CRS (RX_DV can continue past CRS).	R/W	1b	1b
Flip_Chip	11	Used for applications where the core or application is mirror-imaged. Channel D acts like channel A with t10pol_inv set and vice-versa. Channel C acts like channel B with t10pol_inv set and vice-versa. This forces the correctness of all MDI/MDIX and polarity issues.	R/W	0b	0b
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 18d, bit 13 according to the required MDI-X mode.	R/W	1b	1b
MDI-X Mode	13	Force MDI-X mode. Valid only when operating in manual mode. (PHY register 18d, 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

Table 13-35. Port Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
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/external PHY register 4011h, bit 15. In order to have the clock probed out, it is required to perform the following write sequence: PHY register 18d, bit 15 = 1b PHY register 31d = 4010h (page select) PHY register 17d = 0080h PHY register 31d = 0000h (page select)	R/W	0b	0b

13.3.8.24 Interrupt Enable Register (82573E/82573V/82573L) (18d; R/W)

Table 13-36. Interrupt Enable Register 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	Reserved. Always write 00b.	R/W	0b	Retain
Energy Detect Interrupt Enable	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. 0b = Interrupt disable.	R/W	0b	Retain
Symbol Error Interrupt Enable	9	1b = Interrupt enable. 0b = Interrupt disable.	R/W	0b	Retain

Table 13-36. Interrupt Enable Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
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
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

13.3.8.25 Revision ID Register (82563EB/82564EB)

(Page 193, 18d; R/W)

Table 13-37. Revision ID Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
revi_id	15:0	Revision ID.	RO	00021h	

13.3.8.26 Inband Control Register (82563EB/82564EB)

(Page 194, 18d; R/W)

Table 13-38. Inband Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	7:0	Reserved, write as read. **Note:** Bit 4 must be set to 1b.	R/W	0b	0b
Link Status Retransmission Period	15:8	Link status transmission period in tens of μs.	R/W	5h	5h

13.3.8.27 Transmit Amplitude 1000 Register (82563EB/82564EB) (Page 255, 18d; R/W)

Table 13-39. Transmit Amplitude 1000 Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
1000BASE-T Transmit Amplitude MDI[3]	3:0	Same encoding as 1000BASE-T transmit amplitude MDI[1].	R/W	0b	Retain
1000BASE-T Transmit Amplitude MDI[2]	7:4	Same encoding as 1000BASE-T transmit amplitude MDI[1].	R/W	0b	Retain
1000BASE-T Transmit Amplitude MDI[1]	11:8	Same encoding as 1000BASE-T transmit amplitude MDI[1].	R/W	0b	Retain
1000BASE-T Transmit Amplitude MDI[0]	15:12	0000b = 0%, 0001b = -2%, 0010b = -4%, 0011b = -6% 0100b = -8%, 0101b = -10%, 0110b = -12%, 0111b = -14% 1000b = +0%, 1001b = +2%, 1010b = +4%, 1011b = +6% 1100b = +8%,1101b = +10%, 1110b = +12%,1111b = +14%	R/W	0b	Retain

13.3.8.28 Link Health Register (82571EB/82572EI) LINK (19d; RO)

Table 13-40. 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. If An_Enable is true, valid_chan_A = dsplockA latched on the rising edge of link_fail_inhibit_timer_done and link = 0b. If An_enable is false, valid_chan_A = dsplockA.	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 13-40. 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 01d, bit 4, PHY register 06d, bit 4, and PHY register 10d, 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: N/A. 100: Scrambler unlocked. 1000: Local receiver not OK.	RO/ LH	0b	0b
Count Overflow	10	32 idle error events were counted in less than 1 ms.	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 10d, bit 12.	RO/ LH	0b	0b
Gigabit Master Resolution	12	Gig has resolved to master. This is a duplicate of PHY register 10d, bit 14. Programmers must read PHY register 10d, 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

13.3.8.29 Interrupt Status Register[1] (19d; RO)

Table 13-41. Interrupt Status Register 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. Always write 0b.	RO, LH	0b	0b
Energy Detect Changed	4	1b = Energy detect state changed. 0b = No energy detect state change detected.	RO, LH	0b	0b
Downshift Interrupt	5	1b = Downshift detected. 0b = No downshift.	RO, LH	0b	0b
MDI Crossover Changed	6	1b = Crossover changed. 0b = No crossover change.	RO, LH	0b	0b
FIFO Over/Underflow	7	1b = Over/underflow error. 0b = No FIFO error. This is a reserved bit for the **82563EB/ 82564EB**.	RO, LH	0b	0b
False Carrier	8	Does not apply to 1000 Mb/s. 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 = No link state change.	RO/ LH	0b	0b
Auto-Negotiation Completed	11	1b = Auto-negotiation completed. 0b = Auto-negotiation did not complete.	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 = No duplex change.	RO, LH	0b	0b
Speed Changed	14	1b = Speed changed. 0b = No speed change.	RO, LH	0b	0b
Auto-Negotiation Error	15	An error is said to occur if MASTER/ SLAVE does not resolve, parallel detect fault, no common HCD, or link does not come up after negotiation completes. 1b = Auto-negotiation error 0b = No Auto-negotiation error	RO, LH	0b	0b

1. Interrupt Status Register 2 for the **82563EB/82564EB**.

13.3.8.30 Device ID Register (82563EB/82564EB)

(Page 193, 19d; R/W)

Table 13-42. Device ID Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
device_id	15:0	Device ID	RO	02AAh	

13.3.8.31 GLCI Diagnostic Register (82563EB/82564EB)

(Page 194, 19d; R/W)

Table 13-43. GLCI Diagnostic Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Comma Align Lock Loss	7:0	SerDes RX comma alignment lost.	RO, LH	0b	0b
RX PCS Lock	1	Serial RX PCS not synchronized.	RO, LH	0b	0b
Reserved	3:2	Reserved, write as read.	R/W	11b	11b
Reserved	6:4	Reserved.	RO, LH	0b	0b
CRC Error	7	A CRC error occurred in an inband message.	RO, LH	0b	0b
Reserved	15:8	Reserved, write as read.	R/W	00000001b	0000 0001b

13.3.8.32 1000Base-T FIFO Register (82571EB/82572EI)

PFIFO (20d; R/W)

Table 13-44. 1000Base-T 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 1000Base-T'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
Reserved	7:4	Always read as 0b. Write to 0b for normal operation.	R/W	00000b	0000b

Table 13-44. 1000Base-T FIFO Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
FIFO Out Steering	9:8	00b, 01b: Enable the output data bus from 1000Base-T FIFO to transmitters, drives zeros on the output loop-back bus from 1000Base-T FIFO to external application and to DSP RX-FIFOs in test mode. 10b: Drive zeros on output bus from 1000Base-T FIFO to transmitters, enable data on the output loop-back bus from 1000Base-T FIFO to external application and to DSP RX-FIFOs in test mode. 11b: Enable the output data bus from 1000Base-T 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 1000Base-T'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 slower than internal 1000Base-T's gtx_clk has allowed the FIFO to fill to capacity mid packet. Decrease buffer size.	RO/ LH	0b	0b
FIFO Underflow	15	Status bit set when read clock that is faster than internal 1000Base-T's gtx_clk empties the FIFO mid packet. Increase the buffer size.	RO/ LH	0b	0b

13.3.8.33 Extended PHY Specific Control Register (82573E/82573V/ 82573L)

(20d; R/W)

Table 13-45. Extended PHY Specific Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	0	Reserved.	R/W	0b	Retain

Table 13-45. Extended PHY Specific Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reverse Autoneg	1	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (PHY register 0 bit 15). Restart Auto-Negotiation is asserted (PHY register 0 bit 9) Power down (PHY register 0 bit 11) transitions from power down to normal operation. Copper link goes down. Upon hardware reset this bit defaults as follows: REV_ANEG / Page 26, PHY register 0 bit 13 0b / 0b 1b / 1b When REV_ANEG transitions from 1b to 0b this bit is set to 0b. When REV_ANEG transitions from 0b to 1b this bit is set to 1b. 1b = Reverse Auto-negotiation 0b = Normal Auto-negotiation	R/W		Retain
Disable 1000BASE-T	2	When set to disabled, 1000BASE-T is not advertised even if PHY register 9 bits 9:8 are set to 1b. A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (PHY register 0 bit 15). Restart Auto-negotiation is asserted (PHY register 0 bit 9). Power down (PHY register 0 bit 11) transitions from power down to normal operation. Copper link goes down. Upon hardware reset this bit defaults as follows: A1000_DIS / Bit 20.2 0b / 0b 1b /1b When A1000_DIS transitions from 1b to 0b this bit is set to 0b. When A1000_DIS transitions from 0b to 1b this bit is set to 1b. 1b = Disable 1000BASE-T Advertisement 0b = Enable 1000BASE-T Advertisement	R/W		Retain
Reserved	3	Always write 0b.	R/W	0b	Update

Table 13-45. Extended PHY Specific Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Default MAC Interface Speed	6:4	Changes to these bits are disruptive during normal operation. As a result, any changes to these registers must be followed by software reset to take effect. MAC Interface Speed during Link down while Auto-Negotiation is enabled. Bit Speed TX_CLK speed / link down TX_CLK speed - 1000BASE-T link: 000b = 10 Mbps 2.5 MHz / 0 MHz 001b = 100 Mbps 25 MHz / MHz 01Xb = 1000 Mbps 0 MHz /0 MHz 100b = 10 Mbps 2.5 MHz / 2.5 MHz 101b = 100 Mbps 25 MHz / 25 MHz 110b = 1000 Mbps 2.5 MHz / 2.5 MHz 111b = 1000 Mbps 25 MHz /25 MHz	R/W	06h	Update
Transmitter Type	7	0b = Class B. 1b = Class A.	R/W	0b	Retain
Downshift Enable	8	Changes to these bits are disruptive during normal operation. As a result, any changes to these registers must be followed by software reset to take effect. 1b = Enable downshift. 0b = Disable downshift.	R/W	01h	Update
Downshift Counter	11:9	Changes to these bits are disruptive during normal operation. As a result, any changes to these registers must be followed by software reset to take effect. 1x, 2x, ...8x is the number of times the PHY attempts to establish gigabit link before the PHY downshifts to the next highest speed. 000b = 1x - 100b = 5x 001b = 2x - 101b = 6x 010b = 3x - 110b = 7x 011b = 4x - 111b = 8x.	R/W	06h	Update
Disable Link Pulses	12	1b = Disable link pulse. 0b = Enable link pulse.	R/W	0b	0b

Table 13-45. Extended PHY Specific Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Power Down	13	Power down is controlled via PHY register 0 bits 13 and 11. Both bits must be set to 0b before the PHY 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 when PHY register bit 15 and PHY register 0 bit 9) are not set by the programmer. IEEE power down shuts down the Ethernet controller except for the 1000Base-T interface if PHY register 16 bit 3 is set to 1. If set to 0b, then the 1000Base-T interface also shuts down. 1b = Power down. 0b = Normal operation.	R/W	0b	Retain
Line Loopback	14	1b = Enable line loopback. 0b = Normal operation.	R/W	0b	0b
Block Carrier Extension Bit	15	1b = Enable Block Carrier Extension. 0b = Disable Block Carrier Extension. In 1000Base-T mode, PHY register 20 bit 15 is the register bit used to block carrier extension.	R/W	0b	0b

13.3.8.34 Power Management Control Register (82563EB/82564EB) (Page 193, 20d; R/W)

Table 13-46. Power Management Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Enable Electrical Idle	0	Enable SerDes electrical idle on link down and either: • Enable reverse AN (9h) • Disable 1000 in non-D0 (11h)	R/W	1b	1b
Disable Port	1	Disable SerDes and GLCI logic and powers down the PHY. Register logic is not disabled. The PHY should be reset with either a soft or hard reset after writing this bit to a 0b.	R/W	0b	0b
Disable SerDes	2	Disable SerDes.	R/W	0b	0b
Reverse Auto-Negotiation	3	Enable Auto-Negotiation.	R/W	1b	1b
Disable 1000 in non-D0	4	Disable 1000 Mb/s Auto-Negotiation.	R/W	0b	0b
Always Disable 1000	5	Disable 1000 Mb/s Auto-Negotiation.	R/W	0b	0b

Table 13-46. Power Management Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
D0a Reverse Auto-Negotiation	6	Enable D0a Auto-Negotiation.	R/W	0b	0b
Force Power State	7	If set to 1b, use the programmed power bits (9:8) instead of the power state communicated inband.	R/W	0b	0b
Programmed Power State	9:8	Programmed power state. 00b = Dr 01b = D0u 10b = D0a 11b = D3	R/W	0b	0b
Reserved	15:10	Reserved, write as 0b.	R/W	0b	0b

13.3.8.35 Channel Quality Register (82571EB/82572EI) CHAN (21d; RO)

Table 13-47. Channel Quality Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
MSE_A	3:0	The converged mean square error for Channel A.	RO	0b	0b
MSE_B	7:4	The converged mean square error for Channel B.	RO	0b	0b
MSE_C	11:8	The converged mean square error for Channel C.	RO	0b	0b
MSE D	15:12	The converged mean square error for Channel D. 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

13.3.8.36 Receive Error Counter Register (21d; RO)

Table 13-48. Receive Error Counter Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Receive Error Count	15:0	Counter pegs at FFFFh and does not roll over. Both false carrier and symbol errors are reported.	RO, LH	0000h	Retain

13.3.8.37 MAC Control Register (82563EB/82564EB) (Page 2, 21d; R/W)

Table 13-49. MAC Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
PCS Loopback Speed	2:0	These bits control the speed used for PCS loopback. Changing this value is disruptive to normal operations and must be followed by a software reset. 100b = 10 Mb/s 101b = 100 Mb/s 110b = 1000 Mb/s Others = Reserved	R/W	110b	Update
Reserved	3	Reserved. Write as read	R/WP	0b	Retain
Reserved	13:4	Reserved. Write as read.	R/WP	104h	See Note[1]
Line Loopback	14	1b = Enable line loopback 0b = Disable line loopback	R/W	0b	0b
Reserved	15	Reserved. Write as read.	R/WP	0b	0b

1. Some reserved bits are preserved across a software reset, while others are zeroed. However, as long as the value read is written back in this field, a software reset has no effect on any of the bits.

13.3.8.38 Extended Address Register (82573E/82573V/82573L) (22d; R/W)

Table 13-50. Extended Address Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Page Select for PHY Register 28	7:0	PHY register 22 bits 7:0 are used for selecting the VCT* pair. Results are read from PHY register 28. 03h = MDI pair 3 02h = MDI pair 2 01h = MDI pair 1 00h = MDI pair 0 **Note:** For all other register accesses, PHY register 22 bits 7:0 must be 00h.	R/W	00h	Retain
Reserved	15:8	Reserved.	RO, LH	00h	00h

13.3.8.39 Page Register (82563EB/82564EB) (22d; R/W)

Table 13-51. Page Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Page	7:0	Subsequent reads or writes to registers in the address range of 16-21 and 23-28 go to this page.	R/W	0b	See note
Reserved	15:8	Reserved. Write as 0, ignore on read.	R/W	0b	0b

NOTE: After a PHY soft reset, if the value of the PHY register 22 was 0, it remains 0b. If it was any value other than 0b, the register goes into an indeterminate state and must be written before reading or writing any registers in the address rage of 16-21 or 23-28.

13.3.8.40 Indirect Access Control Register (82563EB/82564EB) (Page 255, 23d; R/W)

Table 13-52. Indirect Access Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Address	7:0	Address of indirect register (17h or 18h are valid).	R/W	0b	X
Reserved	11:8	Reserved, write as 0b.	R/W	0b	X
Read	12	Writing a 1b to this bit initiates a read to the register simultaneously programmed into the Address field.	R/W	0b	X
Write	13	Writing a 1b to this bit initiates a write to the register simultaneously programmed into the Address field. The write data should have already been written to the Indirect Access Write Data Register.	R/W	0b	X
Reserved	14	Reserved, write as 0b.	R/W	0b	X
Read Complete	15	1b = Read complete, and Indirect Access Read Data Register is valid. 0b = Read not complete. Always write this bit to a 0b.	R/W	0b	X

13.3.8.41 LED Control Register (82573E/82573V/82573L) (24d; R/W)

Table 13-53. LED Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
LED_TX Control (MSB)	0	See bit 6.	R/W	0b	Retain
LED_RX Control	1	1b = Receive activity/link. 0b = Receive activity.	R/W	0b	Retain
LED_DUPLEX Control	2	Bit 2 and bit 7: LED duplex behavior: 00b -- Low = Full Duplex, High = Half Duplex, Blink = Collision. 01b -- Low = Full Duplex, High = Half Duplex. 10b -- Reserved. 11b -- Reserved.	R/W	0b	Retain
LED_LINK Control	5:3	000b = Direct LED mode. 011b = Master/Slave LED mode.	R/W	0b	Retain
LED_TX Control (LSB)	6	Bit 6 and bit 0: 00b = Transmit activity - solid on 01b = Transmit or receive activity - solid on. 10b = Link and no activity - solid on. Transmit or receive activity - blink. 11b = Transmit or receive activity - blink.	R/W	0b	Retain
LED Duplex Control	7	See bit 2.	R/W	0b	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	1b	Retain
Force Interrupt	11	1b = Force INT pin to assert. 0b = Normal operation.	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 011b = 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 = 1.3 s to 2.7 s	R/W	04h	Retain
Disable LED	15	1b = Disable. 0b = Enable.	R/W	0b	Retain

13.3.8.42 Indirect Access Write Data Register (82563EB/82564EB) (Page 255, 24d; WO)

Table 13-54. Indirect Access Write Data Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Write Data	15:0	Data written to indirectly access register.	WO	0b	X

13.3.8.43 Manual LED Override Register (82573E/82573V/82573L) (25d; R/W)

Table 13-55. Manual LED Override Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
LED_TX	0	LED Off means LED pin output = high. LED On means LED pin output = low. 00b = Normal. 01b = Blink. 10b = LED Off. 11b = LED On.	R/W	0b	Retain
LED_RX	1	LED Off means LED pin output = high. LED On means LED pin output = low. 00b = Normal. 01b = Blink. 10b = LED Off. 11b = LED On.	R/W	0b	Retain
LED_LINK1000	2	LED Off means LED pin output = high. LED On means LED pin output = low. 00b = Normal. 01b = Blink. 10b = LED Off. 11b = LED On.	R/W	0b	Retain
LED_LINK100	3	LED Off means LED pin output = high. LED On means LED pin output = low. 00b = Normal. 01b = Blink. 10b = LED Off. 11b = LED On.	R/W	0b	Retain

Table 13-55. Manual LED Override Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
LED_LINK10	9:8	LED Off means LED pin output = high. LED On means LED pin output = low. 00b = Normal. 01b = Blink. 10b = LED Off. 11b = LED On.	R/W	0b	Retain
LED_DUPLEX	11:10	LED Off means LED pin output = high. LED On means LED pin output = low. 00b = Normal. 01b = Blink. 10b = LED Off. 11b = LED On.	R/W	0b	Retain
Reserved	13:12	Must be set to 00b.	R/W	0b	Retain
Reserved	15:14	Always write 00b.	R/W	0b	Retain

13.3.8.44 PHY Power Management (82571EB/82572EI) (25d; R/W)

Table 13-56. PHY Power Management Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
SPD_EN	0	Smart Power Down When set, enables PHY Smart Power Down mode. Note that bit can be loaded from EEPROM.	R/W	1b	1b
D0LPLU	1	D0 Low Power Link Up When set, configures the PHY to negotiate for a low speed link while in D0a state.	R/W	0b	0b
LPLU	2	Low Power on Link Up When set, enables the decrease in link speed while in non-D0a states when the power policy and power management state specify it. Note that bit can be loaded from EEPROM.	R/W	1b	1b
Disable 1000 nD0a	3	Disables 1000 Mb/s operation in non-D0a states. Note that this bit can be loaded from EEPROM.	R/W	0b	0b
Link Energy Detect	4	This bit is set when the PHY detects energy on the link. Note that this bit is valid only if AN enabled (PHY register 00b, bit 12) and SPD_EN is enabled (PHY register 25d, bit 0).	R/W	0b	0b
Reserved	5	Reserved.	R/W	0b	0b

Table 13-56. PHY Power Management Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Disable 1000	6	When set, disables 1000 Mb/s in all power modes.	R/W	0b	0b
Reserved	7	Always read as 0b. Write to 0b for normal operation.	R/W	0b	0b
rst_compl	8	Indicates PHY internal reset cleared.	LH	0b	0b
Reserved	15:9	Always read as 0b. Write to 0b for normal operation.	R/W	0b	0b

13.3.8.45 Indirect Access Read Data Register (82563EB/82564EB) (Page 255, 25d; RO)

Table 13-57. Indirect Access Read Data Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Read Data	15:0	Read written to indirectly access register.	RO	X	X

13.3.8.46 Copper Specific Control Register (82563EB/82564EB) (26d; R/W)

Table 13-58. Special Gigabit Disable Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	0	Reserved. Write as 1b.	R/W	0b	Retain.
10 BT Polarity Force	1	1b = Force negative polarity. 0b = Normal operation.	R/W	0b	Retain
1000 Mb Test Select	3:2	00b = Normal Operation. 10b = Select 112 ns sequence. 11b = Select 16 ns sequence.	R/W	00b	Retain
Reserved	5:4	Reserved. Write as 0b.	R/W	0b	Retain
Reserved	6	Reserved. Write as 1b.	R/W	1b	Retain
Reserved	11:7	Reserved. Write as 0b.	R/W	0b	Retain
100BASE-T Transmitter Type	12	Transmitter type for 100 Mb/s operation. 1b = Class A. 0b = Class B. Note that 10 Mb/s operation always uses class B.	R/W	0b	Retain

Table 13-58. Special Gigabit Disable Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reverse Autoneg	13	A write to this register bit does not take effect until any one of the following occurs: Software reset is asserted (PHY register 0 bit 15). Restart Auto-Negotiation is asserted (PHY register 0 bit 9) Power down (PHY register 0 bit 11 and PHY register 0, page 16, bit 2) transitions from power down to normal operation. Copper link goes down. A change in priority speed changes this bit. A change in priority speed from 10 Mb/s to 1000 Mb/s sets this bit to 0b. A change in priority speed from 1000 Mb/s to 10 Mb/s sets this bit to 1b. The value of this bit, after a hard reset, depends on the MODE_SEL pin at reset deassertion[1]. • MODE_SEL = 1b - 0b. • MODE_SEL = 0b - 1b. It is recommended that software not use this bit, but instead use PHY register 20, page 193, bit 3 and PHY register 20, page 193, bit 6.	R/W		Retain

Table 13-58. Special Gigabit Disable Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Disable 1000BASE-T	14	When set to disabled, 1000BASE-T is not advertised even if PHY register 9 bits 9:8 are set to 1b. A write to this register bit does not take effect until any one of the following occurs: • Software reset is asserted (PHY register 0 bit 15). • Restart Auto-Negotiation is asserted (PHY register 0 bit 9). • Power down (PHY register 0 bit 11 or PHY register 0, page 16, bit 2) transitions from power down to normal operation. • Copper link goes down. Changes to PHY register 20, page 193, bit 4 and PHY register 20, page 193, bit 5 might affect this bit. Also, if either PHY register 20, page 193, bit 4 or PHY register 20, page 193, bit 5 are 1b, changes in power state might affect this bit. Any transition of either bit or the current power state that would disable 1000 Mb/s auto-negotiation sets this bit and any transition of PHY register 20, page 193, bit 4 or PHY register 20, page 193, bit 5, or the current power state that would enable 1000 Mb/s auto-negotiation clears this bit. It is highly recommended that software either use PHY register 20, page 193, bit 4 or PHY register 20, page 193, bit 5, or this bit to disable 1000 Mb/s auto-negotiation, but not both.	R/W	0b	Retain
1000BASE-T Transmitter Type	15	Transmitter type for 1000 Mb/s operation. 1b = Class A. 0b = Class B.	R/W	1b	Retain

1. MODE_SEL is latched at the rising edge of PHY_PWR_GOOD, the rising edge of PHY_RESET_N, or the falling edge of PHY_SLEEP, whichever occurs last. All three events generate hard resets that sets this bit to the value determined by MODE_SEL. Writing to the Port Reset bit (PHY register 17, page 193, bit 0) or receiving an inband reset also generates a hard reset to the PHY, but won't cause MODE_SEL to be re-latched. In that case, the previously latched version of MODE_SEL is relevant.

13.3.8.47 VTC* DSP Register (82563EB/82564EB) (Page 5, 26d; RO)

Table 13-59. VCT* DSP Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Cable Length	2:0	A rough estimate of cable length. The actual value depends on the attenuation of the cable, output levels of the remote transceiver, connector impedance, etc. 000b = < 50 m 001b = 50 - 80 m 010b = 80 - 110 m 011b = 110 - 140 m 100b = > 140 m	RO	0b	0b
Reserved	15:3	Reserved	RO	0b	0b

13.3.8.48 Special Gigabit Disable Register (82571EB/82572EI) (26d; R/W)

Table 13-60. Special Gigabit Disable Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	15:0	Always read as 0b. Write to 0b for normal operation.	R/W	0b	0b

13.3.8.49 Misc Cntrl Register 1 (82571EB/82572EI) (27d; R/W)

Table 13-61. Misc Cntrl Register 1 Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	4:0	Reserved.	R/W	0b	0b
T10_auto_pol_dis	5	When set, disables the auto-polarity mechanism in the 10 block.	R/W	0b	0b
ss_cfg_cntr	8:6	Smart speed counter configuration: 1-5 (001b:101b).	R/W	010b	010b
Reserved	15:9	Reserved.	R/W	0b	0b

13.3.8.50 Misc Cntrl Register 2 (82571EB/82572EI) (28d; RO)

Table 13-62. Misc Cntrl Register 2 Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	7:0	Reserved.	R/W	0b	0b
Act_an_adv_10hdx	8	Indicates the actual AN advertisement of the PHY for 10 Half-Duplex Capability. 0b = Not 10 Half Duplex Capable. 1b = 10 Half Duplex Capable.	RO	0b	0b
Act_an_adv_10fdx	9	Indicates the actual AN advertisement of the PHY for 10 Full-Duplex Capability. 0b = Not 10 Full Duplex Capable. 1b = 10 Full Duplex Capable.	RO	0b	0b
Act_an_adv_100hd	10	Indicates the actual AN advertisement of the PHY for 100 half-Duplex Capability. 0b = Not 100 Half Duplex Capable. 1b = 100 Half Duplex Capable.	RO	0b	0b
Act_an_adv_100fd	11	Indicates the actual AN advertisement of the PHY for 100 Full-Duplex Capability. 0b = Not 100 Full Duplex Capable. 1b = 100 Full Duplex Capable.	RO	0b	0b
Act_an_adv_gighdx	12	Indicates the actual AN advertisement of the PHY for 1000 Half-Duplex Capability. 0b = Not 1000 Half Duplex Capable. 1b = 1000 Half Duplex Capable.	RO	0b	0b
Act_an_adv_gigfdx	13	Indicates the actual AN advertisement of the PHY for 1000 Full-Duplex Capability. 0b = Not 1000 Full Duplex Capable. 1b = 1000 Full Duplex Capable.	RO	0b	0b
Reserved	15:14	Reserved.	R/W	0b	0b

NOTE: Bits 13:8 might differ from the corresponding bits in PHY register 04d and 09d due to non-IEEE PHY features (lplu, an1000_dis, and smart-speed).

13.3.8.51 MDI[0] Virtual Cable Tester* Register (28d; RO)[1]

Table 13-63. MDI[0] VTC* Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Distance	7:0	Approximate distance (+/- 1 m) to the open/short location on Pair MDI[0]+/-, measured at nominal conditions (room temperature and typical VDDs).	RO	0b	Retain
Amplitude	12:8	Amplitude of reflection on pair MDI[0]+/-. These amplitude bits range from 07h to 1Fh. 1Fh = Maximum positive amplitude (+1). 10h = Zero amplitude. 00h = Maximum negative amplitude (-1).	RO	00h	Retain
Status	14:13	MDI[0] +/- VCT test result: 00b = Valid test, normal cable (no short or open in cable). 01b = Valid test, short in cable (Impedance < 33 ohms). 10b = Valid test, open in cable (Impedance greater than 330 ohms). 11b = Test Fail.	RO	0b	Retain
Run VCT* Test	15	1b = Run VCT* test. 0b = VCT* test completed.	R/W, SC	0b	Retain

1. Page 5, 16d; RO for the **82563EB/82564EB**.

13.3.8.52 MDI[1] Virtual Cable Tester* Register (Page 1, 28d; RO)[1]

Table 13-64. MDI[1] VTC* Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Distance	7:0	Approximate distance (+/- 1 m) to the open/short location on Pair MDI[1]+/-, measured at nominal conditions (room temperature and typical VDDs).	RO	0b	Retain
Amplitude	12:8	Amplitude of reflection on pair MDI[1]+/-. These amplitude bits range from 07h to 1Fh. 1Fh = Maximum positive amplitude (+1). 10h = Zero amplitude. 00h = Maximum negative amplitude (-1).	RO	00h	Retain
Status	14:13	MDI[1] +/- VCT test result: 00b = Valid test, normal cable (no short or open in cable). 01b = Valid test, short in cable (Impedance < 33 ohms). 10b = Valid test, open in cable (Impedance greater than 330 ohms). 11b = Test Fail.	RO	0b	Retain
Disable Waiting Period	15	1b = Start VCT* without waiting to bring link down. 0b = Wait for link down before starting VCT*.	R/W	0b	Retain

1. Page 5, 17d; RO for the **82563EB/82564EB**.

13.3.8.53 MDI[2] Virtual Cable Tester* Register (82573E/82573V/82573L) (Page 2, 28d; RO)[1]

Table 13-65. MDI[2] VTC* Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Distance	7:0	Approximate distance (+/- 1 m) to the open/short location on Pair MDI[2]+/-, measured at nominal conditions (room temperature and typical VDDs).	RO	0b	Retain
Amplitude	12:8	Amplitude of reflection on pair MDI[2]+/-. These amplitude bits range from 07h to 1Fh. 1Fh = Maximum positive amplitude (+1). 10h = Zero amplitude. 00h = Maximum negative amplitude (-1).	RO	00h	Retain
Status	14:13	MDI[2] +/- VCT test result: 00b = Valid test, normal cable (no short or open in cable). 01b = Valid test, short in cable (Impedance < 33 ohms). 10b = Valid test, open in cable (Impedance greater than 330 ohms). 11b = Test Fail.	RO	0b	Retain
Reserved	15	Reserved.	R/W	0b	Retain

1. Page 5, 18d; RO for the **82563EB/82564EB**.

13.3.8.54 MDI[3] Virtual Cable Tester* Register (82573E/82573V/82573L) (Page 3, 28d; RO)[1]

Table 13-66. MDI[3] VTC* Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Distance	7:0	Approximate distance (+/- 1 m) to the open/short location on Pair MDI[3]+/-, measured at nominal conditions (room temperature and typical VDDs).	RO	0b	Retain
Amplitude	12:8	Amplitude of reflection on pair MDI[3]+/-. These amplitude bits range from 07h to 1Fh. 1Fh = Maximum positive amplitude (+1). 10h = Zero amplitude. 00h = Maximum negative amplitude (-1).	RO	00h	Retain
Status	14:13	MDI[3] +/- VCT test result: 00b = Valid test, normal cable (no short or open in cable). 01b = Valid test, short in cable (Impedance < 33 ohms). 10b = Valid test, open in cable (Impedance greater than 330 ohms). 11b = Test Fail.	RO	0b	Retain
Reserved	15	Reserved.	R/W	0b	Retain

1. Page 5, 19d; RO for the **82563EB/82564EB**.

13.3.8.55 100BASE-T Pair Skew Register (Page 4, 28d; RO)[1]

Table 13-67. 100BASE-T Pair Skew Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Pair 1, 2 (MDI[0]+/-)	3:0	Skew = bit value x 8 ns. Value is correct to within +/- 8 ns. The contents of PHY register 28, page 4, bits 15:0 are valid only if PHY register 28, page 5, bit 6 = 1b.	RO	0b	0b
Pair 3, 6 (MDI[1]+/-)	7:4	Skew = bit value x 8 ns. Value is correct to within +/- 8 ns.	RO	0b	0b
Pair 4, 5 (MDI[3]+/-)	11:8	Skew = bit value x 8 ns. Value is correct to within +/- 8 ns.	RO	0b	0b
Pair 7, 8 (MDI[3]+/-)	15:12	Skew = bit value x 8 ns. Value is correct to within +/- 8 ns.	RO	0b	0b

1. Page 5, 20d; RO for the **82563EB/82564EB**.

13.3.8.56 100BASE-T Pair Swap and Polarity Register (Page 5, 28d; RO)[1]

Table 13-68. 100BASE-T Pair Swap and Polarity Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Pair 1, 2 (MDI[0]+/-)	0[1]	Skew = bit value x 8 ns. Value is correct to within +/- 8 ns. The contents of PHY register 28, page 4, bits 15:0 are valid only if PHY register 28, page 5, bit 6 = 1b. For the **82563EB/82564EB**: 1b = Negative. 0b = Positive.	RO	0b	0b
Pair 3, 6 (MDI[1]+/-)	1	Skew = bit value x 8 ns. Value is correct to within +/- 8 ns. For the **82563EB/82564EB**: 1b = Negative. 0b = Positive.	RO	0b	0b
Pair 4, 5 (MDI[3]+/-)	2	Skew = bit value x 8 ns. Value is correct to within +/- 8 ns. For the **82563EB/82564EB**: 1b = Negative. 0b = Positive.	RO	0b	0b
Pair 7, 8 (MDI[3]+/-)	3	Skew = bit value x 8 ns. Value is correct to within +/- 8 ns. For the **82563EB/82564EB**: 1b = Negative. 0b = Positive.	RO	0b	0b
A, B Crossover	4	1b = Channel A received on MDI[0]+/- 1b = Channel B received on MDI[1]+/- 0b = Channel B received on MDI[0]+/- 0b = Channel A received on MDI[1]+/-	RO	0b	0b
C, D Crossover	5	1b = Channel C received on MDI[2]+/- 1b = Channel D received on MDI[3]+/- 0b = Channel D received on MDI[2]+/- 0b = Channel C received on MDI[3]+/-	RO	0b	0b
PHY Register 28 Page 4 and Page 5 are Valid Valid (**82563EB/ 82564EB**)	6	1b = Valid 0b = Invalid For the **82563EB/82564EB**, valid bit. The rest of the bits of this register, as well as register 20, are only valid when this bit is set to 1b.	RO	0b	0b
Reserved	15:7	Reserved.	RO	0b	0b

1. For the **82563EB/82564EB**, bits 5:0 are valid only if bit 6 = 1b.

1. Page 5, 21d; RO for the **82563EB/82564EB**.

13.3.8.57 Extended Address Register (82573E/82573V/82573L) (29d; R/W)

Table 13-69. Extended Address Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Page Select	4:0	00000b = No register 30 page access or page 0 of register 30. 00010b = Select page 2 of register 30. 01011b = Select page 11 of register 30. 01100b = Select page 12 of register 30. 10000b = Select page 16 of register 30. 10010b = Select page 18 of register 30.	R/W	00h	Retain
Reserved	15:5	Reserved	R/W	000h	000h

13.3.8.58 Alternate Page Register (82563EB/82564EB) (29d; R/W)

Table 13-70. Alternate Page Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Page	7:0	Subsequent reads or writes to registers in the address range of 30-31 go to this page.	R/W	0b	See note
Reserved	15:8	Reserved. Write as 0, ignore on read.	R/W	0b	Retain

NOTE: Always write to this register before reading or writing PHY registers 30 or 31.

13.3.8.59 Test Clock Control Register (82563EB/82564EB) (30d; R/W)

Table 13-71. Test Clock Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	2:0	Reserved. Write as 0b.	R/W	0b	Retain
LED Output	6:3	0000b = Normal operation. 1000b = IEEE 1000 Mb/s Test Mode including: LEDA/B_RX_ACTIVITY = TXCLK. LEDA/B_TX_ACTIVITY = TXCLK / 6. Others: Reserved.	R/W	0b	Retain
Reserved	15:7	Reserved.	R/W	0b	Retain

NOTE: Before reading or writing this register, PHY register 29 must first be set to 0b.

13.3.8.60 IEEE Test Mode Selector Register (82573E/82573V/82573L) (30d; R/W)

Table 13-72. IEEE Test Mode Selector Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	2:0	Reserved	R/W	0b	0b
LED Output	5:3	000b = Normal. 110b =: LINK10 - Negative Linkpulse. LINK100 - Positive Linkpulse. LINK1000 - TX_TCLK. DUPLEX - TX_TCLK / 6. RX - TX_TCLK / 4. TX - TX_TCLK LFSR Strobe.	R/W	0b	Retain
Reserved	15:6	Reserved	R/W	000h	000h

13.3.8.61 10/100 Test Features Register (82573E/82573V/82573L) (Page 2, 30d; R/W)

Table 13-73. 10/100 Test Features Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	0	Reserved	R/W	0b	0b
10 BT Polarity Force	1	1b = Force negative polarity. 0b = Do not force negative polarity.	R/W	0b	Retain
16/112 Pattern Select	2	1b = Select 16 ns sequence. 0b = Select 112 ns sequence.	R/W	0b	Retain
100 Mb Test Select	3	1b = Select 100 MB transmit test mode.	R/W	0b	Retain
Reserved	15:4	Reserved	R/W	000h	000h

13.3.8.62 Analog 3 Register (82573E/82573V/82573L) (Page 11, 30d; R/W)

Table 13-74. Analog 3 Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Reserved	14:0	Reserved.	R/W	0000h	0000h
Transmitter Type	15	1b = Class A. 0b = Class B.	R/W	0b	Retain

13.3.8.63 CRC Checker Result Register (82573E/82573V/82573L) (Page 12, 30d; RO)

Table 13-75. CRC Checker Result Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
CRC Error Count	7:0	Error counter is stored in these bits. 00000000b = no frame counted. 11111111b = maximum number of frame counted. The counter does not clear on a read command. To clear the CRC error counter, disable the crc_checker (PHY register 30, page 16, bit 0 = 0b).	RO	00h	Retain
Frame Count	15:8	Frame count is stored in these bits. 00000000b = no frame counted. 11111111b = maximum number of frames counted. The counter does not clear on a read command. To clear the frame counter, disable the crc_checker (PHY register 30, page 16, bit 0 = 0b).	RO	00h	Retain

13.3.8.64 Test Enable Control Register (82573E/82573V/82573L) (Page 16, 30d; R/W)

Table 13-76. Test Enable Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Enable CRC Checker	0	1b = Enable CRC checker. 0b = Disable CRC checker, clear frame counter, and CRC error counter (PHY register 30, page 12, bits 15:0 = 0000h).	R/W	0b	Retain
Enable gigabit Stub Loopback	1	**Note:** PHY register 30, page 18, bit 0 must be set to 1b in addition to this bit in order for the Stub loopback to operate correctly. 1b = Enable Gigabit Stub loopback when PHY register 30, page 18, bit 0 = 1b. 0b = Normal operation.	R/W	0b	Retain
Reserved	15:2	Reserved. These bits must be read and maintained as read when performing a write to bit 0 or bit 1.	RO	0000h	0000h

13.3.8.65 Miscellaneous Control Register (82573E/82573V/82573L) (Page 18, 30d; R/W)

Table 13-77. Miscellaneous Control Register Bit Description

Field	Bit(s)	Description	Mode	HW Rst	SW Rst
Turn off NEXT Canceller	0	1b = Disable NEXT. 0b = Enable NEXT.	R/W	0b	Retain
Reserved	1	Reserved. These bits must be read and maintained as read when performing a write to the other bits.	R/W	0b	Retain
Packet Generator	5:2	Bit 5: 1b = Packet generation enable, 0b = disable. Bit 4: 0b = Random, 1b = 5Ah (packet pattern select). Bit 3: 0b = 64 byte, 1b = 1518 byte (packet length select). Bit 2: 1b = generate error packet, 0b = no error.	R/W	0b	Retain
Reserved	11:6	Reserved. Always write 001h.	R/W	1b	Retain
Reserved	15:12	Reserved. These bits must be read and maintained as read when performing a write to the other bits.	R/W	0b	Retain

13.3.8.66 Page Select Core Register[1]

(31d; WO)

Table 13-78. Page Select Core 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

13.3.8.67 82563EB/82564EB Copper Specific Control Register 1 (Page 0, Register 16

(3B60h; R/W)

Table 13-79. Copper Specific Control Register 1 Description

Function	Bit(s)	Description	HW Rst	SW Rst
Disable Jabber	0	Jabber has effect only in 10BASE-T half-duplex mode. 1b = Disable jabber function. 0b = Enable jabber function.	0b	Retain
Polarity Reversal Disable	1	If polarity is disabled, then the polarity is forced to be normal in 10BASE-T. 1b = Polarity reversal disabled 0b = Polarity reversal enabled	0b	Retain
Alternate Power Down	2	Power down is controlled via register bits 0.11 and 0`16.2. Both bits must be set to 0 before the PHY 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 when bits Reset (0.15) and Restart Auto-Negotiation (0.9) are not set by the programmer. 1b = Power down. 0b = Normal operation.	0b	Retain
Copper Transmitter Disable	3	1b = Transmitter disable. 0b = Transmitter enable.	0b	Retain
Reserved	4	Reserved, write as 0b.	0b	Retain

1. This is a reserved register for the **82573E/82573V/82573L**. Always set to 0b.

Function	Bit(s)	Description	HW Rst	SW Rst
MDI Crossover Mode	6:5	Changes to these bits are disruptive to the normal operation; therefore, any changes to these registers must be followed by a software reset to take effect. 00b = Manual MDI configuration. 01b = Manual MDIX configuration. 10b = Reserved. 11b = Enable automatic crossover for all modes.	11b	Update
Enable Extended Distance	7	When using cable exceeding 100m, the 10BASE-T receive threshold must be lowered in order to detect incoming signals. 1b = Lower 10BASE-T receive threshold. 0b = Normal 10BASE-T receive threshold.	0b	Retain
Energy Detect	9:8	0xb = Off 10b = Sense only on receive (Basic Energy Detect). 11b = Sense and periodically transmit NLP (Energy Detect*).	11b	Update
Force Copper Link Good	10	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. 1b = Force link good. 0b = Normal operation.	0b	Retain
Downshift Enable	11	Changes to these bits are disruptive to the normal operation; therefore, any changes to these registers must be followed by software reset to take effect. 1b = Enable downshift. 0b = Disable downshift.	1b	Update
Downshift Counter	14:12	Changes to these bits are disruptive to the normal operation; therefore, any changes to these registers must be followed by software reset to take effect. 1x, 2x, ...8x is the number of times the PHY attempts to establish 1000BASE-T link before the PHY downshifts to the next highest speed. 000b = 1x 100 = 5x. 001b = 2x 101 = 6x. 010b = 3x 110 = 7x. 011b = 4x 111 = 8x.	011b	Update
Disable Link Pulses	15	1b = Disable link pulse 0b = Enable link pulse	0b	0b

13.3.8.68 82563EB/82564EB Page Register (Any Page, Register 22)

(0000h; R/W)

Table 13-80. Page Register Description

Function	Bit(s)	Description	HW Rst	SW Rst
Page	7:0	Subsequent reads or writes to registers in the address range of 16-21 and 23-28 go to this page.	0h	See below
Reserved	15:8	Write as 0b, ignore on read.	0h	0h

Note: After a **82563EB/82564EB** soft reset, if the value of the page register was 0h, it remains 0h. If it was any value other than 0h, the register stays in an indeterminate state and must be written before reading or writing any registers in the address rage of 16-21 or 23-28.

13.3.8.69 82563EB/82564EB Alternate Page Register (Any Page, Register 29)

(0000h; R/W)

Table 13-81. Alternate Page Register Description

Function	Bit(s)	Description	HW Rst	SW Rst
Page	5:0	Subsequent reads or writes to registers in the address range of 30-31 go to this page.	0h	Retain
Reserved	15:6	Write as 0b, ignore on read.	0h	Retain

Note: Always write to this register before reading or writing registers 30 or 31.

13.3.8.70 82563EB/82564EB GLCI Mode Control Register (Page 193, Register 16)

(0180h; R/W)

Table 13-82. GLCI Mode Control Register Description

Function	Bit(s)	Description	Rst
Reserved	1:0	Reserved, write as 00b.	00b
Force Duplex	2	Use the programmed duplex value instead of the duplex indication from the **82563EB/82564EB**.	0b
Programmed Duplex	3	Programmed duplex value, used only when force duplex is 1b. 0b = Half duplex. 1b = Full duplex.	0b
Reserved	4	Reserved, write as 0b.	0b
phy_leds_en	5	Select between **82563EB/82564EB** and MAC LEDs 0b = GLCI inband LEDs from the **631xESB/ 632xESB**. 1b = **82563EB/82564EB** LEDs	0b
Force Link Up	6	Indicate link up to the **631xESB/632xESB** even if the **82563EB/82564EB** is indicating link down. The **82563EB/82564EB** might indicate link down in loopback modes.	0b
Reserved	10:7	Reserved, write as read.	0011b
Pass False Carrier	11	Pass false carrier (RXDV=0b, RXER=1b) from the **82563EB/82564EB** to the **631xESB/632xESB**.	0b
Reserved	15:12	Reserved, write as read.	0h

13.3.9 PHY Address and Page Register (82563EB/82564EB)

The IEEE specification allows five bits for the register access. Registers 0 to 15 are defined by the specification, while registers 16 to 31 are left available to the OEM.

The **82563EB/82564EB** implements many registers for diagnostic purposes. In addition, the **82563EB/82564EB** contains registers controlling the serial/GLCI interface as well as top-level functions. The total number of registers implemented far exceeds the 16 registers available to the OEM. When this occurs, a common technique is to use paging.

The **82563EB/82564EB**'s registers are divided into pages. Each page has 32 registers. Registers 0-15 are identical in all the pages and are the IEEE defined registers. Register 22 is the page register in all pages. Registers 16-21 and 23-28 are page specific. Register 29 serves as an alternative page register. It is present in all pages and controls access to registers 30-31.

In order to read or write registers 16-21 and 23-28, software should first set the page register (register 22) to map the appropriate page. Software can then read or write any register in that page. Likewise, to read or write registers 30-31, software should set the alternate page register (register 29) to map the appropriate page, then read or write to the register.

82563EB/82564EB registers are mapped to pages 0-255. GLCI and other non-PHY registers are mapped to pages 192-195.

13.3.10 SERDES ANA (82571EB/82572EI)

SERDESCTL (00024h; R/W)

Table 13-83. SERDES ANA Bit Description

Field	Bit(s)	Initial Value	Description
Data	7:0	0b	Data to SerDes.
Address	15:8	0b	Address to SerDes.
Reserved	30:16	0b	Reserved.
Done Indication	31	1b	When a write operation completes, this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.

13.3.11 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 PCIe* 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 13-84. FCAL Register Bit Description

Field	Bit(s)	Initial Value	Description
FCAL	31:0	X	Flow Control Address Low Should be programmed with 00_C2_80_01h

13.3.12 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 13-85. FCAH Register Bit Description

Field	Bit(s)	Initial Value	Description
FCAH	15:0	X	Flow Control Address High Should be programmed with 01_00h.
Reserved	31:16	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

13.3.13 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 13-86. FCT Register Bit Description

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.

13.3.14 GLCI Control and Status Registers (631xESB/632xESB)

KUMCTRLSTA (00034h; R/W)

The GLCI MAC-to-PHY registers are controlled through a mechanism at address 00034h.

To write to one of these registers, software writes the offset of the register in the *Offset* field, bits 20:16. REN bit (bit 21) is reset to 0b. The data that is written resides in the lower bits of the register (bits 15:0).

To read from one of these registers, software first writes to address 00034h with the *Offset* field including the offset of the register and bit REN set to 1b. This causes the offset to be latched in the internal register. Software then performs a read operation from the same address with the requested data residing on bits 15:0.

Note: These registers can be accessed by both software and firmware. As a result, programmers have to implement a careful arbitration, in order to prevent collisions between accesses from both sides.

Note: For a read operation, the *REN* bit should be written to 1b only once before performing the read.

Note: GLCI registers cannot be read through I/O space. They can only be read through memory space.

Field	Bit(s)	Initial Value	Description
Control and Status	15:0	X	Data written into the GLCI register or read from it. **Note:** Bit 4 must be set to 0b.
Offset	20:16	0b	Offset of the internal register in the MAC/PHY interface.
REN	21	0b	When '0', the Control and Status bits are written into the selected register. 1b = Data is not written, but address is latched, so an upcoming read access to the same register provides status bits.
Reserved	31:22		Reserved.

Notes: During initialization and after a system reset, the following bit values must be set:

- At base address 34h, offset 10, bit 2 must be set to 0b for link at 1000 Mb/s and 1b for link at 10/100 Mb/s.
- At base address 34h, offset 0, bits 3 and 11 must be set to a value of 808h.
- At base address 34h, offset 2, bit 4 must be set to a value of 510h.

13.3.15 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 13-87. VET Register Bit Description

Field	Bit(s)	Initial Value	Description
VET	15:0	8100h	VLAN EtherType Should be programmed with 8100h.
Reserved	31:16	0b	Reserved Reads as 0b.

13.3.16 MDC/MDIO PHY Address Register (631xESB/632xESB)

MDPHYA (0003Ch; R/W)

This register holds the PHY address. It can be used in case the external PHY base address is not zero. It is strongly recommended to keep the register at its default value.

Table 13-88. MDC/MDIO PHY Address Bit Description

Field	Bit(s)	Initial Value	Description
PHYA	4:0	Port0 = 0b Port1 = 1b	PHY address.
Reserved	31:5	0b	Reserved.

13.3.17 ULT Fuse Register 3 (82573E/82573V/82573L)

UFUSE3 (000F0h; RO)

Table 13-89. ULT Fuse Register 3 Bit Description

Field	Bit(s)	Initial Value	Description
DRRED	14:0	X	Data RAM Redundancy Fuses.
CRRED	27:15	X	Code RAM Redundancy Fuses.
ENAD	28	0b	Enable Data RAM Redundancy Fuses This fuse has a default value of 0b (Data RAM redundancy disabled).
ENAC	29	0b	Enable Code RAM Redundancy Fuses This fuse has a default value of 0b (Code RAM redundancy disabled).
Reserved	31:30	0b	Reserved, reads as 0b (no fuses).

13.3.18 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 13-90. FCTTV Register Bit Description

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 0h prior to initializing the Pause frame. For the **631xESB/632xESB**, the recommended value is FFFFh.
Reserved	31:16	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

13.3.19 Transmit Configuration Word Register[1]

TXCW (00178h; R/W)

This register is applicable while operating in internal SerDes mode. For internal PHY mode, program the register to 0000h.

This register has two meanings depending on the state of ANE: 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. The value of lower 16 bits of this register is encoded as two 10-bit symbols and sent as the *config_word* field for the /C/ ordered set during Auto-Negotiation.

If ANE is cleared, then the Ethernet controller is performing 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 (in general) read back the contents of this register. If hardware loses receive synchronization, the contents of this register changes and during the time of the change, the value read back might be inconsistent. In the absence of loss of synchronization, the value read back is stable and equal to the last value written.

Note: Careful attention to the IEEE 802.3z standard is required in order to meet the specified requirements for timing during a software negotiated link.

1. Not applicable to the **82573E/82573V/82573L**.

Table 13-91. TXCW Register Bit Description

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.
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.

13.3.20 Receive Configuration Word Register[1]

RXCW (00180h; R)

This register is applicable while operating in internal SerDes mode. The RXCW register records the partner abilities and provides indications about its Auto-Negotiation status.

Note: While in internal SerDes mode, software might be required to inspect or monitor the results of RXCW to generate a link up/down indication.

The contents of this register depends on the state of TXCW.ANE. If ANE is set, then this register records the 16 bits defined in IEEE 802.3z and is called the AN link partner ability base page register.

Note: RXCW.ANC should be ignored if software Auto-Negotiation is used. The ANC bit is not indicative of link status in this mode.

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. In addition to the 16-bit configuration word, five other bits qualify the exchange of data. The bit RxSynchronize qualifies all other bits in the register (when in software Auto-Negotiation). Each time the receive MAC loses bit synchronization, this bit becomes 0b and stays 0b (sticky) until software reads this register. RxConfig further indicates whether the interface is receiving the /C/ ordered set, or the normal idle/data stream. Once the interface has seen a change to RxConfigWord, RxConfigChange is set. This bit is sticky high until software reads this register.

If hardware sees an invalid symbol at any time, it sets RxConfigInvalid. This bit is sticky high until software reads this register. Software is supposed to restart the configuration process if it receives an invalid symbol at any time during the link start-up negotiation.

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.

1. Not applicable to the **82573E/82573V/82573L**.

Table 13-92. 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.
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.
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.

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.
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.
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.

13.3.21 LED Control

LEDCTL (00E00h; RW)

Table 13-93. LED Control Bit Description

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	4	0b	Reserved. Read-only as 0b. Write as 0b for future compatibility.
GLOBAL_ BLINK_MODE	5	0b[1]	Global Blink Mode. This field specifies the blink mode of all the LEDs. 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off.
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[1]	LED1/ACTIVITY# Mode. This field specifies the control source for the LED1 output. An initial value of 0011b selects ACTIVITY# indication.
Reserved	12	0b	Reserved. Read-only as 0b. Write as 0b for future compatibility.
LED1_ BLINK_MODE	13	0b[1]	LED1 Blink Mode. This field needs to be configured with the same value as GLOBAL_BLINK_MODE as it specifies the blink mode of the LED. 0b = Blink at 200 ms on and 200ms off. 1b = Blink at 83 ms on and 83 ms off.
LED1_IVRT	14	0b[1]	LED1/ACTIVITY# Invert.
LED1_BLINK	15	1b[1]	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	20	00b	Reserved. Read-only as 0b. Write as 0b for future compatibility.
LED2_ BLINK_MODE	21	0b[1]	LED2/LINK100# Blink Mode. This field needs to be configured with the same value as GLOBAL_BLINK_MODE as it specifies the blink mode of the LED. 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off.

Field	Bit	Initial Value	Description
LED2_IVRT	22	0b[1]	LED2/LINK100# Invert.
LED2_BLINK	23	0b[1]	LED2/LINK100# Blink.
LED3_MODE	27:24	0111b[1]	LED3/LINK1000# Mode. This field specifies the control source for the LED3 output. An initial value of 0111b selects LINK1000# indication. This is a reserved bit for the **82573E/82573V/82573L**.
Reserved	28	0b	Reserved. Read-only as 0b. Write as 0b for future compatibility.
LED3_ BLINK_MODE	29	0b[1]	LED3/LINK1000# Blink Mode. This field needs to be configured with the same value as GLOBAL_BLINK_MODE as it specifies the blink mode of the LED. 0b = Blink at 200 ms on and 200ms off. 1b = Blink at 83 ms on and 83 ms off. This is a reserved bit for the **82573E/82573V/82573L**.
LED3_IVRT	30	0b	LED3/LINK1000# Invert. This is a reserved bit for the **82573E/82573V/82573L**.
LED3_BLINK	31	0b	LED3/LINK1000# Blink. This is a reserved bit for the **82573E/82573V/82573L**.

1. These bits are read from the EEPROM/NVM.

13.3.21.1 MODE Encodings for LED Outputs

Table 13-94 lists the MODE encodings used to select the desired LED signal source for each LED output.

Note: When LED Blink mode is enabled the appropriate LED Invert bit should be set to zero.

The dynamic LED modes (FILTER_ACTIVITY, LINK/ACTIVITY, COLLISION, ACTIVITY, PAUSED) should be used with LED Blink mode enabled.

When LED blink mode is enabled, the blinking frequencies are 1/5 of the rates listed in the Table 13-94.

For the **631xESB/632xESB**, when operating in SerDes mode, the **631xESB/632xESB** speed and duplex LEDs should not be used. Since 1000 Mb/s full duplex is the only mode supported, the speed and duplex LEDs can be hardwired on the system board to reflect this state.

Note: All 15 modes listed are functional.

Table 13-94. Mode Encodings for LED Outputs

Mode	Pneumonic	State / Event Indicated
0000b	LINK_10/1000	Asserted when either 10 or 1000 Mb/s link is established and maintained.
0001b	LINK_100/1000	Asserted when either 100 or 1000 Mb/s link is established and maintained.
0010b	LINK_UP	Asserted when any speed link is established and maintained.

Mode	Pneumonic	State / Event Indicated
0011b	FILTER_ACTIVITY	Asserted when link is established and packets are being transmitted or received that passed MAC filtering.
0100b	LINK/ACTIVITY	Asserted when link is established and when there is no transmit or receive activity.
0101b	LINK_10	Asserted when a 10 Mb/s link is established and maintained.
0110b	LINK_100	Asserted when a 100 Mb/s link is established and maintained.
0111b	LINK_1000	Asserted when a 1000 Mb/s link is established and maintained.
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	ACTIVITY	Asserted when link is established and packets are being transmitted or received.
1100b	BUS_SIZE	Asserted when the Ethernet controller detects a 1 Lane PCIe* connection.
1101b	PAUSED	Asserted when the Ethernet controller's transmitter is flow controlled.
1110b	LED_ON	Always high. Assuming no optional inversion selected, causes output pin high / LED ON for typical LED circuit.
1111b	LED_OFF	Always low. Assuming no optional inversion selected, causes output pin low / LED OFF for typical LED circuit.

13.3.22 Extended Configuration Control (82573E/82573V/82573L) EXTCNF_CTRL (00F00h; R/W)

Table 13-95. Extended Configuration Control Bit Description

Field	Bit(s)	Initial Value	Description
Reserved	0	0b[1]	Should be set to 0b.
PHY Write Enable	1	0b[1]	When set, enables loading the Extended PHY Configuration area in the **82573E/82573V/82573L**. When disabled, the Extended PHY Configuration area is ignored.
D/UD EN	2	0b[1]	When set, enables loading the Extended Dock or Undock Configuration area in the **82573E/82573V/82573L**. When disabled, the Extended Dock and Undock Configuration areas are ignored.
Reserved	3	1b[1]	Reserved.
Dock Config Owner	4	0b	Determines the owner to load the dock/undock configuration into the **82573E/82573V/82573L**. When cleared, hardware is responsible for loading the dock/undock configuration into the **82573E/82573V/82573L**. When set, software is responsible for loading the dock/undock configuration. **Note:** This bit is reset to its default state each time the PHY is reset.
MDIO SW Ownership	5	0b	Software request for access to MDIO. Part of the arbitration scheme for MDIO access.
MDIO HW Ownership	6	0b	Hardware request for access to MDIO. Part of the arbitration scheme for MDIO access
Reserved	15:7	0b	Reserved.
Extended Configuration Pointer	27:16	0b[1]	Defines the base address (in DW) of the Extended Configuration area in the NVM. A value of 000h means that the Extended Configuration area is disabled.
Reserved	31:28	0b	Reserved.

1. These bits are read from the NVM.

13.3.23 Extended Configuration Size (82573E/82573V/82573L) EXTCNF_SIZE (00F08h; R/W)

Table 13-96. Extended Configuration Size Bit Description

Field	Bit(s)	Initial Value	Description
Extended PHY Length	7:0	0b[1]	Size (in DW) of the Extended PHY Configuration area.
Extended Dock Length	15:8	0b[1]	Size (in DW) of the Extended Dock Configuration area and the Extended Undock Configuration area.
Reserved	23:16	0b[1]	Reserved. Should be set to 0b.
Reserved	31:24	0b	Reserved.

1. These bits are read from the NVM.

13.3.24 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 six bits. The transmit allocation is read-only and is calculated based on PBA.RXA. The partitioning size is 1 KB.

Note: Programming this register does not automatically re-load or initialize internal packet-buffer RAM pointers. Software must reset both transmit and receive operation (using the global device reset **CTRL.RST** bit) after changing this register in order for it to take effect. The PBA register itself is not reset by assertion of the global reset, but is only reset upon initial hardware power-on.

Note: For best performance, the transmit buffer allocation should be set to accept two full-sized packets (For good 9 KB jumbo frame performance (not applicable to the **82573E/82573V**), the transmit allocation should be a minimum of 18 KB).

Note: Transmit packet buffer size should be configured to be more than 8 KB.

Table 13-97. PBA Register Bit Description

Field	Bit(s)	Initial Value	Description
RXA	15:0	0020h	Receive Packet Buffer Allocation in KB The upper 10 bits are read only as 0b. Default is 32 KB (12 KB for the **82573E/82573V/82573L**).
TXA	31:16	0010h 0014h (**82573E/82573V/82573L**)	Transmit Packet Buffer Allocation in KB These bits read only. Default is 16 KB (20 KB for the **82573E/82573V/82573L**).

13.3.25 MNG EEPROM Control Register (82571EB and 82573E/82573V/82573L)

EEMNGCTL (01010h; R/W)

Field	Bit(s)	Initial Value	Description
CFG_DONE	18	0b	MNG Configuration Cycle Done This bit indicates that the MNG configuration cycle (configuration of SerDes, PHY, GIO and PLLs) is done. This bit is set to 1b by MNG firmware to indicate configuration done and cleared by hardware on any of the reset sources that cause the firmware to initialize the PHY. Writing a 0b by firmware does not affect the the state of this bit. **Note:** Software should not try to access the PHY for configuration unless this bit is set first.

13.3.26 Software/Firmware Synchronization (631xESB/632xESB)

SW_FW_SYNC (05B5Ch; R/W)

This register is used to synchronize between software and firmware. This register is common to both ports 0 and 1.

Field	Bit	Attr	Default	Description
SW_EEP_SM	0	RW	0b	When set to 1b, EEPROM access is owned by software.
SW_PHY_SM0	1	RW	0b	When set to 1b, PHY 0 access is owned by software.
SW_PHY_SM1	2	RW	0b	When set to 1b, PHY 1 access is owned by software.
SW_MAC_CSR_SM	3	RW	0b	When set to 1b, software owns access to shared CSRs.
Reserved	15:4	RW	0h	Reserved for future use.
FW_EEP_SM	16	RW	0b	When set to 1b, EEPROM access is owned by firmware.
FW_PHY_SM0	17	RW	0b	When set to 1b, PHY 0 access is owned by firmware.
FW_PHY_SM1	18	RW	0b	When set to 1b, PHY 1 access is owned by firmware.
FW_MAC_CSR_SM	19	RW	0b	When set to 1b, firmware owns access to shared CSRs.
Reserved	31:20	RW	0b	Reserved for future use.

Reset conditions:

- The software-controlled bits 15:0 are reset as any other CSR (on global resets, D3hot exit, software reset, and Forced TCO). Software is expected to clear these bits on entry to D3 state.
- The firmware-controlled bits 31:16 are reset on LAN_POWER_GOOD and firmware reset.

Software and firmware synchronize accesses to shared resources in the GbE controller through a semaphore mechanism and a shared configuration register. The SWESMBI bit in the Software Semaphore register (SWSM) and the EEP_FW_semaphore bit in the Firmware Semaphore register (FWSM) serve as a semaphore mechanism between software and firmware.

Once software or firmware takes control over the semaphore, it might access the Software-Firmware Synchronization register (SW_FW_SYNC) and claim ownership of a specific resource. The Software-Firmware Synchronization register includes pairs of bits (one owned by software and the other by firmware), where each pair of bits control a different resource. A resource is owned by software or firmware when the respective bit is set. It is illegal to have both bits in a pair set at the same time.

When software or firmware gains control over the Software-Firmware Synchronization register, it checks if a certain resource is owned by the other (the bit is set). If not, it might set its bits for that resource, taking ownership of the resource. The same process (claiming the semaphore and accessing the Software-Firmware Synchronization register) is done when a resource is being freed.

The following example shows how software might use this mechanism to own a resource (firmware accesses are done in an analogous manner):

1. Software takes control over the software/firmware semaphore:
   a. Software writes a 1b to the SWESMBI bit in the Software Semaphore register (SWSM).
   b. Software then reads the SWESMBI. If set, software owns the semaphore. If cleared, this is an indication that firmware currently owns the semaphore. Software should retry the previous step after some delay.
2. Software reads the Software-Firmware Synchronization register (SW_FW_SYNC) and checks the firmware bit in the pair of bits that control the resource is wants to own.
   a. If the bit is cleared (firmware does not own the resource), software sets the software bit in the pair of bits that control the resource it wants to own.
   b. If the bit is set (firmware owns the resource), see step 4.
3. Software releases the software/firmware semaphore by clearing the SWESMBI bit in the Software Semaphore register (SWSM).
4. If software did not succeed to own the resource (step 2b), software repeats the process after some delay.

The following example shows how software might use this mechanism to release a resource (firmware accesses are done in an analogous manner):

1. Software takes control over the software/firmware semaphore.
   a. Software writes a 1b to the SWESMBI bit in the Software Semaphore register (SWSM).
   b. Software then reads the SWESMBI. If set, software owns the semaphore. If cleared, this is an indication that firmware currently owns the semaphore. Software should retry the previous step after some delay.
2. Software writes a 0b to the software bit in the pair of bits that control the resource is wants to release.
3. Software releases the software/firmware semaphore by clearing the SWESMBI bit in the Software Semaphore register (SWSM).
4. Software waits some delay before trying to gain the semaphore again.

Note: There are distinct bits for each PHY port.

13.3.27 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 (0-19) is set in this register. A PCIe* interrupt is generated each time one of the bits (0-19) in this register is set, the corresponding interrupt is enabled through the Interrupt Mask Set/Read Register (see Section 13.3.30), and the Interrupt Throttle interval (see Section 13.3.28) has expired.

Each time an interrupt causing event occurs, all timers of delayed interrupts are cleared and their cause event is set in the ICR register.

How the ICR register is read can differ as explained in the cases that follow:

- Case 1: Interrupt Mask Register = 0000h (mask all) - ICR content is cleared.
- Case 2: Interrupt asserted (ICR.INT_ASSERTED = 1b) - ICR content is cleared and Auto Mask is active (the IAM register is written to the IMC register).
- Case 3: Interrupt not asserted (ICR.INT_ASSERTED = 0b) - ICR content is **NOT CLEARED**.

Writing a 1b to any bit in this register also clears that bit. Writing a 0b to any bit has no effect on that bit. The INT_ASSERTED bit is a special case. Writing a 1b or 0b to this bit has no effect. It is cleared only when all interrupt sources are cleared. Bits 7, 16, and 17 are also cleared when the CPU vector is cleared.

Note: If IMS = 0b, then the ICR register is always clear-on-read. If IMS is not 0b, but some ICR bit is set where the corresponding IMS bit is not set, then a read does not clear the ICR register. For example, if IMS = 10101010b and ICR = 01010101b, then a read to the ICR register does not clear it. If IMS = 10101010b and ICR = 0101011b, then a read to the ICR register clears it entirely (ICR.INT_ASSERTED = 1b).

Table 13-98. ICR Register Bit Description

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 one of the delayed-timers (TIDV or TADV) expires.
TXQE	1	0b	Transmit Queue Empty Set when the last descriptor block for a transmit queue has been used. When configured to use more than one transmit queue, this interrupt indication is issued if one of the queues is empty and is not cleared until all the queues have valid descriptors.
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/external PHY mode.
RXSEQ	3	0b	Receive Sequence Error While in internal SerDes mode, 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 reserved bit for the **82573E/82573V/82573L**.
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 a result caused by no available receive buffers or because PCIe* receive bandwidth is inadequate.
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. This bit is also cleared when CPU Vector is cleared (either by reading or writing). **Note:** For the **631xESB/632xESB** and the **82571EB/82572EI**, the timer might incorrectly reset after receiving an ACK packet. As a result, the software device driver should also look for receive packets after receiving an ACK frame indication.
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.

Field	Bit(s)	Initial Value	Description
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 13.3.20 for details. Only valid in internal SerDes mode. This is reserved bit for the **82573E/82573V/82573L**.
Reserved	12:11	0b	Reserved. Set these bits to 0b.
GPI_SDP2	13	0b	General Purpose Interrupt on SDP2. If GPI interrupt detection is enabled on this pin (via CTRL_EXT), this interrupt cause is set when the SDP2 is sampled high. This is reserved bit for the **82573E/82573V/82573L**.
GPI_SDP3	14	0b	General Purpose Interrupt on SDP3. If GPI interrupt detection is enabled on this pin (via CTRL_EXT), this interrupt cause is set when the SDP3 is sampled high. This is a reserved bit for the **82573E/82573V/82573L** and the **631xESB/632xESB**.
TXD_LOW	15	0b	Transmit Descriptor Low Threshold hit. Indicates that the descriptor ring has reached the threshold specified in the Transmit Descriptor Control register.
SRPD	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. This bit is also cleared when CPU Vector is cleared (either by reading or writing). **Note:** For the **631xESB/632xESB** and the **82571EB/82572EI**, when working in store bad frames mode, this interrupt might not be asserted when a small packet with bad CRC is received. Instead, the packet interrupt is asserted.
ACK	17	0b	Receive ACK Frame Detected Indicates that an ACK frame has been received and the timer in RAID.ACK_DELAY has expired. This bit is also cleared when CPU Vector is cleared (either by reading or writing). **Note:** For the **631xESB/632xESB** and the **82571EB/82572EI**, when this bit is set, the software device driver should also look for receive packets that are not ACK packets.
Reserved	19:18	0b	Reserved. Should be set to 00b.
RX_DESC_ FIFO_Par0	20	0b	Rx Descriptor FIFO Parity Error (**631xESB/632xESB**) This bit is set when a parity error is detected in the Rx Descriptor FIFO, queue 0.
TX_DESC_ FIFO_Par0	21	X	Tx Descriptor FIFO Parity Error (**631xESB/632xESB**) This bit is set when a parity error is detected in the Tx Descriptor FIFO, queue 0.

Field	Bit(s)	Initial Value	Description
PCIe* Master Data FIFO	22	0b	PCIe* Master Data FIFO (**631xESB/632xESB**) This bit is set when a parity error is detected in the host arbitration read buffer memory.
PBPar	23	0b	Packet Buffer Parity Error (**631xESB/632xESB**) Packet Buffer parity error This bit is set when a parity error is detected in the packet buffer memory, either Rx or Tx packet.
RX_DESC_ FIFO_Par1	24	0b	Rx Descriptor FIFO Parity Error (**631xESB/632xESB**) This bit is set when a parity error is detected in the Rx descriptor FIFO, queue 1.
TX_DESC_ FIFO_Par1	25	0b	Tx Descriptor FIFO Parity Error (**631xESB/632xESB**) This bit is set when a parity error is detected in the Tx descriptor FIFO, queue 1.
Reserved	29:26	0b	Reserved bits for the **631xESB/632xESB**. Read as 0b.
Reserved	30:20	0b	Reserved Reads as 0b.
INT_ASSERTED	31	0b	Interrupt Asserted This bit is set when the LAN port has a pending interrupt. If the Interrupt is enabled in the PCIe* configuration space, an Interrupt is asserted.

13.3.28 Interrupt Throttling Rate

ITR (000C4h; R/W)

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	0b	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:

$$interrupts/\mathrm{second} = (256\times10^{-9}\times interval)^{-1}$$

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:

$$inter\text{-}interrupt\ interval = (256\times10^{-9} \times interrupts/\text{sec})^{-1}$$

The optimal performance setting for this register is very system and configuration specific. A initial suggested range is 651-5580 (28Bh - 15CCh).

Note: When working at 10/100 Mb/s and running at ¼ clock, the interval time is doubled by four.

Setting the Interrupt Throttling Rate (ITR) to a non-zero value might cause an Interrupt Rx/Tx statistics miscount.

13.3.29 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 13.3.27). A PCIe* 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 13.3.30).

Bits written with 0b are unchanged.

Table 13-99. ICS Register Bit Description

Field	Bit(s)	Initial Value	Description
TXDW	0	X	Sets the Transmit Descriptor Written Back Interrupt.
TXQE	1	X	Sets the Transmit Queue Empty Interrupt.
LSC	2	X	Sets the Link Status Change Interrupt.
RXSEQ	3	X	Sets the Receive Sequence Error Interrupt. This is reserved bit for the **82573E/82573V/82573L**.
RXDMT0	4	X	Sets the Receive Descriptor Minimum Threshold Hit Interrupt.
Reserved	5	X	Reserved Should be written with 0b to ensure future compatibility.
RXO	6	X	Sets the Receiver Overrun Interrupt. Sets on Receive Data FIFO Overrun.
RXT0	7	X	Sets the Receiver Timer Interrupt.
Reserved	8	X	Reserved Should be written with 0b to ensure future compatibility.
MDAC	9	X	Sets the MDI/O Access Complete Interrupt.
RXCFG	10	X	Sets the Receiving /C/ Ordered Sets Interrupt. This is reserved bit for the **82573E/82573V/82573L**.

Field	Bit(s)	Initial Value	Description
Reserved	12:11	X	Reserved Should be written with 0b to ensure future compatibility.
GPI	14:13	X	Sets the General Purpose Interrupts. This is reserved bit for the **82573E/82573V/82573L**.
TXD_LOW	15	X	Transmit Descriptor Low Threshold Hit.
SRPD	16	X	Clears the mask for the Small Receive Packet Detect Interrupt.
ACK	17	X	Sets the Receive ACK Frame Detect Interrupt.
Reserved	19:18	X	Reserved.
RxDesFifoPar0[1]	20	0b	Set Rx Descriptor Parity error mask, queue 0.
TxDesFifoPar0[1]	21	0b	Set Tx Descriptor Parity error mask, queue 0.
HostarbPar[1]	22	0b	Set Host arb Parity error mask.
PBPar[1]	23	0b	Set Packet Buffer Parity error mask.
RxDesFifoPar1[1]	24	0b	Set Rx Descriptor Parity error mask, queue 1.
TxDesFifoPar1[1]	25	0b	Set Tx Descriptor Parity error mask, queue 1.
Reserved	31:20	X	Reserved for the **82571EB/82572EI** and the **82573E/82573V/82573L**) Should be written with 0b to ensure future compatibility. For the **631xESB/632xESB**, bits 31:26.

1. **631xESB/632xESB** only.

13.3.30 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 PCIe* 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 13.3.27).

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 13.3.31) rather than writing a 0b to a bit in this register.

Reading this register returns bits that have an interrupt mask set.

Table 13-100. IMS Register Bit Description

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Sets the mask for the Transmit Descriptor Written Back.
TXQE	1	0b	Sets the mask for the Transmit Queue Empty.
LSC	2	0b	Sets the mask for the Link Status Change.
RXSEQ	3	0b	Sets the mask for the Receive Sequence Error. This is reserved bit for the **82573E/82573V/82573L**.
RXDMT0	4	0b	Sets the mask for the Receive Descriptor Minimum Threshold hit.
Reserved	5	0b	Reserved Should be written with 0b to ensure future compatibility.
RXO	6	0b	Sets the mask for Receiver Overrun. Sets on Receive Data FIFO Overrun.
RXT0	7	0b	Sets the mask for the Receiver Timer Interrupt.
Reserved	8	0b	Reserved Should be written with 0b to ensure future compatibility.
MDAC	9	0b	Sets the mask for the MDI/O Access Complete Interrupt.
RXCFG	10	0b	Sets the mask for the Receiving /C/ Ordered Sets. This is reserved bit for the **82573E/82573V/82573L**.
Reserved	12:11	0b	Reserved Should be written with 0b to ensure future compatibility.
GPI	14:13	0b	Sets the mask for the General Purpose Interrupts. This is reserved bit for the **82573E/82573V/82573L**.
TXD_LOW	15	0b	Sets the mask for the Transmit Descriptor Low Threshold hit.
SRPD	16	0b	Sets the mask for the Small Receive Packet Detection.
ACK	17	0b	Sets the mask for the Receive ACK Frame Detection.
Reserved	19:18	X	Reserved.
RxDesFifoPar0[1]	20	0b	Set Rx Descriptor Parity error mask, queue 0.

Field	Bit(s)	Initial Value	Description
TxDesFifoPar0[1]	21	0b	Set Tx Descriptor Parity error mask, queue 0.
HostarbPar[1]	22	0b	Set Host arb Parity error mask.
PBPar[1]	23	0b	Set Packet Buffer Parity error mask.
RxDesFifoPar1[1]	24	0b	Set Rx Descriptor Parity error mask, queue 1.
TxDesFifoPar1[1]	25	0b	Set Tx Descriptor Parity error mask, queue 1.
Reserved	31:20	X	Reserved for the **82571EB/82572EI** and the **82573E/82573V/82573L**) Should be written with 0b to ensure future compatibility. For the **631xESB/632xESB**, bits 31:26.

1. **631xESB/632xESB** only.

13.3.31 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 13.3.30), and the status of the cause bit is reflected in the Interrupt Cause Read Register (see Section 13.3.27).

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.

Table 13-101. IMC Register Bit Description

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Clears the mask for the Transmit Descriptor Written Back.
TXQE	1	0b	Clears the mask for the Transmit Queue Empty.
LSC	2	0b	Clears the mask for the Link Status Change.
RXSEQ	3	0b	Clears the mask for the Receive Sequence Error. This is reserved bit for the **82573E/82573V/82573L**.
RXDMT0	4	0b	Sets mask for Receive Descriptor Minimum Threshold hit.
Reserved	5	0b	Reserved. Reads as 0b. Should be written with 1b to ensure future compatibility.
RXO	6	0b	Clears the mask for Receiver Overrun. Sets on Receive Data FIFO Overrun.
RXT0	7	0b	Clears the mask for the Receiver Timer Interrupt.

Field	Bit(s)	Initial Value	Description
Reserved	8	0b	Reserved: Should be written with 1b to ensure future compatibility.
MDAC	9	0b	Clears the mask for the MDI/O Access Complete Interrupt.
RXCFG	10	0b	Clears the mask for the Receiving /C/ ordered sets. This is reserved bit for the **82573E/82573V/82573L**.
Reserved	12:11	0b	Reserved. Reads as 0b. Should be written with 1b to ensure future compatibility.
GPI	14:13	0b	Clears the General Purpose Interrupts. This is reserved bit for the **82573E/82573V/82573L**.
TXD_LOW	15	0b	Clears the mask for the Transmit Descriptor Low Threshold hit.
SRPD	16	0b	Clears the mask for the Small Receive Packet Detect Interrupt.
ACK	17	0b	Clears the mask for the Receive ACK Frame Detect Interrupt.
Reserved	19:18	X	Reserved.
RxDesFifoPar0[1]	20	0b	Set Rx Descriptor Parity error mask, queue 0.
TxDesFifoPar0[1]	21	0b	Set Tx Descriptor Parity error mask, queue 0.
HostarbPar[1]	22	0b	Set Host arb Parity error mask.
PBPar[1]	23	0b	Set Packet Buffer Parity error mask.
RxDesFifoPar1[1]	24	0b	Set Rx Descriptor Parity error mask, queue 1.
TxDesFifoPar1[1]	25	0b	Set Tx Descriptor Parity error mask, queue 1.
Reserved	31:20	X	Reserved for the **82571EB/82572EI** and the **82573E/82573V/82573L**) Should be written with 0b to ensure future compatibility. For the **631xESB/632xESB**, bits 31:26.

1. **631xESB/632xESB** only.

13.3.32 Interrupt Acknowledge Auto Mask Register

IAM (000E0h; R/W)

Table 13-102. IAM Register Bit Description

Field	Bit(s)	Initial Value	Description
IAM_VALUE	31:0	0b	Each time the *CTRL_EXT.IAME* bit is set and the *ICR.INT_ASSERT* = 1b, an ICR read or write has the side effect of writing the contents of this register to the IMC register.

13.3.33 Receive Control Register

RCTL (00100h; R/W)

This register controls all Ethernet controller receiver functions.

Table 13-103. RCTL Register Bit Description

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 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. **Note:** If RX_ERR occurs during the preamble of the packet, the packet is not stored even if this bit is set.
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.
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.

Field	Bit(s)	Initial Value	Description
LBM	7:6	0b	Loopback mode. Controls the loopback mode of the Ethernet controller. 00b = Normal operation. 01b = MAC loopback. 10b = Undefined. 11b = Reserved. All loopback modes are only allowed when the Ethernet controller is configured for full-duplex operation. Receive data from transmit data is looped back internally to the SerDes or internal/external PHY.
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 13.3.40 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.
DTYP	11:10	0b	Descriptor Type 00b = Legacy description type. 01b = Packet split description type. 10b = Reserved. 11b = Reserved.
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
BAM	15	0b	Broadcast Accept Mode. 0b = Ignore broadcast. 1b = 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.

Field	Bit(s)	Initial Value	Description
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. 11b = 256 Bytes. RCTL.BSEX = 1b: 00b = Reserved; software should not program this value. 01b = 16384 Bytes. 10b = 8192 Bytes. 11b = 4096 Bytes. **Note:** BSIZE is only used when DTYP = 00b. When DTYP = 01b, the buffer sizes for the descriptor are controlled by fields in the Packet Split Receive Control (PSRCTL) register. BSIZE is not relevant when FLXBUF is something other than 0b. in that case, FLXBUF determines the buffer size.
VFE	18	0b	VLAN Filter Enable 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.

Field	Bit(s)	Initial Value	Description
DPF	22	0b	Discard Pause Frames 0b = incoming pause frames subject to filter comparison. 1b = 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.
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 1b, 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.
FLXBUF	30:27	0b	Determines a flexible buffer size. When this field = 0000b, the buffer size is determined by BSIZE. If this field is something other than 0000b, the receive buffer size is the number represented in KB: For example, 0001b = 1 KB (1024 Bytes).
Reserved	31	0b	Reserved Should be written with 0b to ensure future compatibility.

13.3.34 Early Receive Threshold (82573E/82573V/82573L)

ERT (02008h; R/W)

This register contains the RxThreshold value. This threshold determines how many bytes of a given packet should be in the Ethernet controller's on-chip receive packet buffer before it attempts to begin transmission of the frame on the PCIe* bus. This register enables software to configure the Ethernet controller to use early receives. This means configuring the Ethernet controller to start transferring the packet across the PCIe* bus before it has received the entire frame into its packet buffer. Note that this register has an effect only when the receive packet buffer is nearly empty (the only data in the packet buffer is from the packet that is currently on the wire).

Note: When Early Receive is used in parallel to the Packet Split Receive feature, the minimum value of the ERT register should be larger than the header size to enable the actual packet split.

Table 13-104. ERT Register Bit Description

Field	Bit(s)	Initial Value	Description
RxThreshold	12:0	0000h	Receive Threshold Value This threshold is in units of 8 bytes.
Reserved	13	0b	Reserved for the **82573V** only.
Reserved	31:13	0b	Reads as 0b. Should be written to 0b for future compatibility.

13.3.35 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 (1b), 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 13-105. FCRTL Register Bit Description

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	30:16	0b	Reserved Should be written with 0b for future compatibility. Reads as 0b.
XONE	31	0b	XON Enable 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.

13.3.36 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. This value must be at least 8 bytes less than the maximum number of bytes allocated to the Receive Packet Buffer (PBA, RXA), and the lower 3 bits must be programmed to 0b (8 byte granularity). Each time the receive FIFO reaches the fullness indicated by RTH, hardware transmits a PAUSE frame if the transmission of flow control frames is enabled.

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.

Note: The value of FCRTH.RTH should be at least 128 bytes more than the value in FCTRL.RTL.

Table 13-106. FCRTH Register Bit Description

Field	Bit(s)	Initial Value	Description
Reserved	2:0	0b	Reserved Must be written with 0b. Reads as 0b.
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	30:16	0b	Reserved Should be written with 0b for future compatibility. Reads as 0b.
FlowCntDis	31	0b	Flow Control Disable 1b = Flow control disabled. 0b = Flow control enabled. This is a reserved bit for the **82573E/82573V/82573L**.

13.3.37 Packet Split Receive Control Register

PSRCTL (02170h; R/W)

This register determines the receive buffer size.

Table 13-107. PSRCTL Register Bit Description

Field	Bit(s)	Initial Value	Description
BSIZE0	6:0	256 bytes	Receive Buffer size for Buffer 0 The value is in 128-byte resolution. The value can be from 128 bytes to 16256 bytes (15.875 KB). The default buffer size is 256 bytes. Software should not program this field to a zero value.
Reserved	7	0b	Reserved. Should be written with 0b to ensure future compatibility.
BSIZE1	13:8	4 KB	Receive Buffer Size for Buffer 1 The value is in 1 KB resolution. The value can be from 1 KB to 63 KB. The default buffer size is 4 KB. Software should not program this field to a zero value.
Reserved	15:14	0b	Reserved. Should be written with 0b to ensure future compatibility.
BSIZE2	21:16	4 KB	Receive Buffer Size for Buffer 2 The value is in 1 KB resolution. The value can be from 1 KB to 63 KB. The default buffer size is 4 KB. Software should not program this field to a zero value.
Reserved	23:22	0b	Reserved. Should be written with 0b to ensure future compatibility.
BSIZE3	29:24	0 KB	Receive Buffer Size for Buffer 3 The value is in 1 KB resolution. The value can be from 1 KB to 63 KB. The default buffer size is 0 KB. Software should not program this field to a zero value.
Reserved	31:30	0b	Reserved. Should be written with 0b to ensure future compatibility.

13.3.38 Receive Descriptor Base Address Low Queue 0

RDBAL0 (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 13-108. RDBAL0 Register Bit Description

Field	Bit(s)	Initial Value	Description
0	3:0	0b	Ignored on writes. Returns 0b on reads.
RDBAL0	31:4	X	Receive Descriptor Base Address Low Queue 0.

13.3.39 Receive Descriptor Base Address High Queue 0

RDBAH0 (02804h; R/W)

This register contains the upper 32 bits of the 64-bit Descriptor Base Address.

Table 13-109. RDBAH0 Register Bit Description

Field	Bit(s)	Initial Value	Description
RDBAH0	31:0	X	Receive Descriptor Base Address Queue 0[63:32]

13.3.40 Receive Descriptor Length Queue 0

RDLEN0 (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 13-110. RDLEN0 Register Bit Description

Field	Bit(s)	Initial Value	Description
Zero	6:0	0b	Zero value Ignore on write. Reads back as 0b.
LEN0	19:7	0b	Receive Descriptor Length Queue 0 Provides the number of receive descriptors (in multiples of eight).
Reserved	31:20	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

13.3.41 Receive Descriptor Head Queue 0

RDH0 (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 13-111. RDH0 Register Bit Description

Field	Bit(s)	Initial Value	Description
RDH0	15:0	0b	Receive Descriptor Head Queue 0.
Reserved	31:16	0b	Reserved. Should be written with 0b for future compatibility. Reads as 0b.

13.3.42 Receive Descriptor Tail Queue 0

RDT0 (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. Software should also write an even number to the tail register if the Ethernet controller uses the Packet Split feature.

Table 13-112. RDT0 Register Bit Description

Field	Bit(s)	Initial Value	Description
RDT0	15:0	0b	Receive Descriptor Tail Queue 0.
Reserved	31:16	0b	Reserved Reads as 0b. Should be written with 0b for future compatibility.

13.3.43 Receive Interrupt Delay Timer (Packet Timer) Register

RDTR (02820h; R/W)

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.

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.

Table 13-113. RDTR Register Bit Description

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. For the **631xESB/632xESB** and the **82571EB/82572EI**, the flush operation stops at the 4 KB boundary. A second flush command is recommended to make sure all descriptors were flushed.

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.

13.3.44 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 PCIe* header CLS field) must not represent greater than 31 descriptors.

Note: When GRAN = 1b, all descriptors are written back (even if not requested).

Table 13-114. RXDCTL Register Bit Description

Field	Bit(s)	Initial Value	Description
PTHRESH	5:0	00h	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	00h	Reserved Reads as 0b. Should be written as 0b for future compatibility.
HTHRESH	13:8	00h	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.

Field	Bit(s)	Initial Value	Description
RSV	15:14	00h	Reserved Reads as 0b. Should be written as 0b for future compatibility.
WTHRESH	21:16	01h	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 one or more of the 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. For the **631xESB/632xESB** and the **82571EB/82572EI**, WTHRESH should be set to the same value in both queues 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.

13.3.45 Receive Interrupt Absolute Delay Timer

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.

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.

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.

13.3.46 Receive Descriptor Base Address Low Queue 1

RDBAL1 (02900h;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 13-115. RDBAL1 Register Bit Description

Field	Bit(s)	Initial Value	Description
0	3:0	0b	Ignored on writes. Returns 0b on reads.
RDBAL1	31:4	X	Receive Descriptor Base Address Low Queue 1.

13.3.47 Receive Descriptor Base Address High Queue 1

RDBAH1 (02904h; R/W)

This register contains the upper 32 bits of the 64-bit Descriptor Base Address.

Table 13-116. RDBAH1 Register Bit Description

Field	Bit(s)	Initial Value	Description
RDBAH1	31:0	X	Receive Descriptor Base Address Queue 1 [63:32]

13.3.48 Receive Descriptor Length Queue 1

RDLEN1 (02908h; 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 13-117. RDLEN1 Register Bit Description

Field	Bit(s)	Initial Value	Description
Zero	6:0	0b	Zero value Ignore on write. Reads back as 0b.
LEN1	19:7	0b	Receive Descriptor Length Queue 1 Provides the number of receive descriptors (in multiples of eight).
Reserved	31:20	0b	Reserved Should be written with 0b to ensure future compatibility. Reads as 0b.

13.3.49 Receive Descriptor Head Queue 1

RDH1 (02910h; 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 13-118. RDH1 Register Bit Description

Field	Bit(s)	Initial Value	Description
RDH1	15:0	0b	Receive Descriptor Head Queue 1.
Reserved	31:16	0b	Reserved. Should be written with 0b for future compatibility. Reads as 0b.

13.3.50 Receive Descriptor Tail Queue 1

RDT1 (02918h; 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. Software should also write an even number to the tail register if the Ethernet controller uses the Packet Split feature.

Table 13-119. RDT1 Register Bit Description

Field	Bit(s)	Initial Value	Description
RDT1	15:0	0b	Receive Descriptor Tail Queue 1.
Reserved	31:16	0b	Reserved Reads as 0b. Should be written with 0b for future compatibility.

13.3.51 Receive Descriptor Control 1

RXDCTL1 (02928h; 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 PCIe* header CLS field) must not represent greater than 31 descriptors.

Note: When GRAN = 1b, all descriptors are written back (even if not requested).

Table 13-120. RXDCTL1 Register Bit Description

Field	Bit(s)	Initial Value	Description
PTHRESH	5:0	00h	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	00h	Reserved Reads as 0b. Should be written as 0b for future compatibility.
HTHRESH	13:8	00h	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	00h	Reserved Reads as 0b. Should be written as 0b for future compatibility.
WTHRESH	21:16	01h	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.

Field	Bit(s)	Initial Value	Description
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.

13.3.52 Receive Small Packet Detect Interrupt RSRPD (02C00h; R/W)

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.

13.3.53 Receive ACK Interrupt Delay Register RAID (02C08h; R/W)

Field	Bit(s)	Initial Value	Description
ACK_DELAY	15:0	0b	The ACK delay timer measured in increments of 1.024 μs. When the Receive ACK Frame Detect interrupt is enabled in the IMS register, ACK packets being received use a unique delay timer to generate an interrupt. When an ACK is received, an absolute timer loads to the value of ACK_DELAY. The interrupt signal is set only when the timer expires. If another ACK packet is received while the timer is counting down, the timer is not reloaded to ACK_DELAY.
Reserved	31:16	0b	Reserved. Reads as 0b.

Note: If an immediate (non-scheduled) interrupt is desired for any received ACK frame, ACK_DELAY should be set to 0b.

13.3.54 CPU Vector Register

CPUVEC (02C10h; R/W)

Field	Bit(s)	Initial Value	Description
CPU Vector	31:0	0h	Each bit is associated with one of 32 possible host processors/ queues. It denotes that the processor has work to do. A received packet sets the appropriate bit based on the output of the redirection table. A register read clears all bits along with a few bits in the Global Cause register (RXT0, ACK and SPRD). Writing a 1b to a bit also clears that bit. Writing 0b to a bit leaves it unchanged. If, as a result of a write operation, the register becomes 0b, the same bits, previously described in the Global Cause register, (RXT0, ACK and SPRD) are also cleared.

13.3.55 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 13-121. RXCSUM Register Bit Description

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 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.

Field	Bit(s)	Initial Value	Description
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.
Field	**Bit(s)**	**Initial Value**	**Description**
Reserved	31:10	0b	Reserved Reads as 0b. Should be written with 0b for future compatibility.

13.3.56 Receive Filter Control Register

RFCTL (05008h; R/W)

Field	Bit(s)	Initial Value	Description
ISCSI_DIS	0	0b	iSCSI Disable Disables the iSCSI filtering.
ISCSI_DWC	5:1	0b	iSCSI Dword Count This field indicates the Dword count of the iSCSI header that is used for the packet split mechanism.
NFSW_DIS	6	0b	NFS Write Disable Disables filtering of NFS write request headers.
NFSR_DIS	7	0b	NFS Read Disable Disables filtering of NFS read reply headers.
NFS_VER	9:8	00b	NFS Version 00b = NFS version 2. 01b = NFS version 3. 10b = NFS version 4. 11b = Reserved for future use.

Field	Bit(s)	Initial Value	Description
IPv6_DIS	10	0b	IPv6 Disable Disables IPv6 packet filtering.
IPv6XSUM_DIS	11	0b	IPv6 Xsum Disable Disables XSUM on IPv6 packets.
ACKDIS	12	0b	ACK Accelerate Disable When set, the Ethernet controller does not accelerate interrupt on TCP packets.
ACKD_DIS	13	0b	ACK Data Disable 0b = Ethernet controller recognizes ACK packets according to the ACK bit in the TCP header plus no CP data. 1b = Ethernet controller recognizes ACK packets according to the ACK bit only.
IPFRSP_DIS	14	0b	IP Fragment Split Disable When this bit is set the header of IP fragmented packets are not set.
EXSTEN	15	0b	Extended Status Enable When the EXSTEN bit is set or when the Packet Split receive descriptor is used, the Ethernet controller writes the extended status to the Rx descriptor.
Reserved	31:16	0b	Reserved Should be written with 0b to ensure future compatibility.

Note: When NFS parsing is disabled in register RFCTL, an NFS packet is recognized as an iSCSI packet.

13.3.57 Transmit Control Register

TCTL (00400h; R/W)

This register controls all transmit functions for the Ethernet controller.

Table 13-122. TCTL Register Bit Description

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.

Field	Bit(s)	Initial Value	Description
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 (3Fh) **Note:** A value of 3Fh results in a collision distance of 64 byte times.
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.
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. For the **631xESB/632xESB**, set this bit to 1b.
UNORTX	25		Underrun No Re-Transmit

Field	Bit(s)	Initial Value	Description
TXCSCMT	27:26		Tx Descriptor Minimum Threshold
MULR	28	1b	Multiple Request Support This bit defines the number of read requests the Ethernet controller issues for transmit data. When set to 0b the Ethernet controller submits only one request at a time, When set to 1b the Ethernet controller can submit up to 4 concurrent requests. The device driver must not modify this register when the Tx head register is not equal to the tail register. This bit is loaded from the EEPROM word 24h/14h (bit 14).
Reserved	31:29	0b	Reserved Read as 0. Should be written with 0b for future compatibility.

Two fields deserve special mention: CT and COLD. Software can choose to abort packet transmission in less than the Ethernet mandated 16 collisions. For this reason, hardware provides CT.

Wire speeds of 1000 Mb/s result in a very short collision radius with traditional minimum packet sizes. COLD specifies the minimum number of bytes in the packet to satisfy the desired collision distance. It is important to note that the resulting packet has special characters appended to the end. These are NOT regular data characters. Hardware strips special characters for packets that go from 1000 Mb/s environments to 100 Mb/s environments. Note that the hardware evaluates this field against the packet size in full duplex as well.

Note: While 802.3x flow control is only defined during full duplex operation, the sending of PAUSE frames via the SWXOFF bit is not gated by the duplex settings within the Ethernet controller. Software should not write a 1b to this bit while the Ethernet controller is configured for half-duplex operation.

RTLC configures the Ethernet controller to perform retransmission of packets when a late collision is detected. Note that the collision window is speed dependent: 64 bytes for 10/100 Mb/s and 512 bytes for 1000 Mb/s operation. If a late collision is detected when this bit is disabled, the transmit function assumes the packet has successfully transmitted. This bit is ignored in full-duplex mode.

Note: For the **631xESB/632xESB**, in order to work in multiple queues it must be also be configured to multiple requests (MULR = 1b) and assert both TXDCTL (3828h) bit 22 and TXDCTL1 (3928h) bit 22.

13.3.58 Transmit Control Extended (631xESB/632xESB)

TCTL_EXT (00404;R/W)

This register controls late collision detection.

Field	Bit(s)	Initial Value	Description
Reserved	9:0	0h	Reserved. Should be set to 0h.
COLD	19:10	0h	Collision Distance Specifies the minimum number of byte times that must elapse for proper CSMA/CD operation. For the **631xESB/632xESB**, the recommended value is 40h.
Reserved	31:20	0b	Reserved.

13.3.59 Transmit IPG Register

TIPG (00410;R/W)

This register controls the Inter Packet Gap (IPG) timer for the Ethernet controller. Note that the recommended TIPG value to achieve 802.3 compliant minimum transmit IPG values in full and half duplex is 00702008h.

Note: For the **631xESB/632xESB**, when operating over a GLCI link at 10 Mb/s or 100 Mb/s Ethernet link speed, the IPGT value should be set to 9 to for correct operation.

Table 13-123. TIPG Register Bit Description

Field	Bit(s)	Initial Value	Description
IPGT	9:0	08h	IPG Back to Back Specifies the IPG length for back to back transmissions in both full and half duplex. Measured in increments of the MAC clock: • 8 ns MAC clock when operating @ 1 Gb/s. • 80 ns MAC clock when operating @ 100 Mb/s. • 800 ns MAC clock when operating @ 10 Mb/s. An offset of 4-byte times must be added to the programmed value to determine the total IPG. A value of 8 is recommended to achieve a 12-byte effective IPG. **Note:** For the **631xESB/632xESB**, when the *Optimize_Nibble_creation_for_HD* bit is set, the IPG is shorter by 1 byte, thus a value of 9 is recommended for the IPGT.
IPGR1	19:10	08h	IPG Part 1 Specifies the portion of the IPG in which the transmitter defers to receive events. IPGR1 should be set to 2/3 of the total effective IPG (8). Measured in increments of the MAC clock: • 8 ns MAC clock when operating @ 1 Gb/s. • 80 ns MAC clock when operating @ 100 Mb/s • 800 ns MAC clock when operating @ 10 Mb/s.

Field	Bit(s)	Initial Value	Description
IPGR	29:20	07h	IPG After Deferral Specifies the total IPG time for non back-to-back transmissions (transmission following deferral) in half duplex. Measured in increments of the MAC clock: • 8 ns MAC clock when operating @ 1 Gb/s. • 80 ns MAC clock when operating @ 100 Mb/s • 800 ns MAC clock when operating @ 10 Mb/s. An offset of 5-byte times must be added to the programmed value to determine the total IPG after a defer event. A value of 7 is recommended to achieve a 12-byte effective IPG. Note that the IPGR must never be set to a value greater than IPGT. If IPGR is set to a value equal to or larger that IPGT, it overrides the IPGT IPG setting in half duplex resulting in inter-packet gaps that are larger then intended by IPGT. In this case, full duplex is unaffected and always relies on IPGT.
Reserved	31:30	0	Reserved Read as 0b. Should be written with 0b for future compatibility.

13.3.60 Adaptive IFS Throttle Register[1]

AIT (00458h;R/W)

Table 13-124. AIT Register Bit Description

Field	Bit(s)	Initial Value	Description
AIFS	15:0	0b	Adaptive IFS Value This value is in units of 8 ns.
Reserved	31:16	0b	This field should be written with 0.

Adaptive IFS 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. 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.

The *AdaptiveIFS* field provides a similar function to the IPGT field in the TIPG register. However, it only affects the initial transmission timing, not re-transmission timing.

Note: If the value of the *AdaptiveIFS* field is less than the *IPGTransmitTime* field in the Transmit IPG register, then it has no effect because the Ethernet controller selects the maximum of the two values.

1. Not applicable to the **631xESB/632xESB**.

13.3.61 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 13-125. TDBAL Register Bit Description

Field	Bit(s)	Initial Value	Description
ZERO	3:0	0b	Zero Value This field is ignored on writes and reads as 0b.
TBDAL	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.

13.3.62 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 13-126. TDBAH Register Bit Description

Field	Bit(s)	Initial Value	Description
TBDAH	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.

13.3.63 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 13-127. TDLEN Register Bit Description

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.

13.3.64 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 13-128. TDH Register Bit Description

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.

13.3.65 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 13-129. TDT Register Bit Description

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.

13.3.66 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) 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 13-130. TIDV Register Bit Description

Field	Bit(s)	Initial Value	Description
IDV	15:0	X	Interrupt Delay Value Counts in units of 1.024 µs. A value of 0b is not allowed.
Reserved	30:16	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.
FDP	31	0b	Flush Partial Description Block when set to 1b, ignored otherwise. Reads as 0b and is self clearing.

13.3.67 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) based on the GRAN flag.

Note: When GRAN = 1b, all descriptors are written back (even if not requested).

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 is set. Furthermore, with transmit descriptor bursting enabled, some descriptors are written back that did not have RS set in their respective descriptors.

LWTHRESH 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.

Table 13-131. TXDCTL Register Bit Description

Field	Bit(s)	Initial Value	Description
PTHRESH	5:0	00h	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	00b	Reserved Reads as 0b. Should be written as 0b for future compatibility.
HTHRESH	15:8	00h	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	00b	Reserved Reads as 0b. Should be written as 0b for future compatibility.
WTHRESH	21:16	00h	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.
Reserved	22	0b 1b[1]	Reserved Reads as 0b. Should be written as 0b for future compatibility. For the **82571EB/82572EI** and the **631xESB/632xESB**, always write to 1b.

Field	Bit(s)	Initial Value	Description
Reserved	23	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). 0b = Cache line granularity. 1b = Descriptor granularity.
LWTHRESH	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. **82571EB/82572EI** and the **631xESB/632xESB**.

13.3.68 Transmit Absolute Interrupt Delay Value TADV (0382Ch; RW)

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.

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.

13.3.69 Transmit Arbitration Counter Queue 0

TARC0 (03840h; RW)

TARC0 and TARC1 registers allow tuning the arbitration parameters, which control the transmission of both transmission queues.

Arbitration between the two queues is provided by:

- Normal Mode – In this mode, the arbitration is done per DMA packets. For non LSO packets, a DMA packet is equivalent to a network packet. For LSO packet, a DMA packet is the full IP packet provided by the host. In this mode, an internal packet counter is decreased each time a DMA packet is sent. When this counter reaches zero the arbitration moves to the second queue. The initial value of this counter when the arbitration is given to the queue is fixed by the COUNT field.

While in Normal mode, network allocation to the queues can be controlled by giving different count values to the two queues.

Note: In order to operate in multiple queues, Ethernet controllers must also be configured to multiple requests (MULR = 1b) and assert both TXDCTL (03828h) bit 22 and TXDCTL1 (03928h) bit 22.

- In Normal mode, COUNT must not be equal to any value except 1. The decrease of the counter can be more than 1 per packet for LSO packets. Currently validated only with COUNT = 1b.

13.3.69.1 Multiple Queues Limitations

Limitation	Description
LSO Limitations with MULR set to 1b	The software driver must add a few bytes of data to the header buffer or set TARC1[22] to 1b. The software driver must set TARC0[25] and TARC1[25] to 1b (0b for the **631xESB/632xESB**) when enabling LSO with MULR set to 1b. **Note:** An MSS smaller than 10 bytes is not supported.

Field	Bit(s)	Initial Value	Description
COUNT	6:0	3	Transmit Arbitration Count The number of packets that can be sent from queue 0 to make the N over M arbitration between the queues. Writing 0h to this register is not allowed.
Reserved	7	0b	Reserved.
RATIO	9:8	00b	Compensation Ratio This value determines the ratio between the number of packets transmitted on queue 1 in a TCP segmentation offload to the number of compensated packets transmitted from queue 0. 00b = 1/1 compensation 01b = 1/2 compensation 10b = 1/4 compensation 11b = 1/8 compensation
ENABLE	10	1b	Descriptor Enable Always set to 1b.
Reserved	20:11	00h	Reserved Reads as 00h. Should be written to 00h for future compatibility.
Reserved	21	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI**, must be set to 0b in 10/100 modes. Can be set to 1b only when the Ethernet controller is configured in GbE speed for small packet performance increase.
Reserved	22	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.
Reserved	23	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI**, must be set when enabling multiple transmit queues. Recommend to always set to 1b.
Reserved	24	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI**, must be set when enabling multiple transmit queues. Recommend to always set to 1b.
Reserved	25	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI**, must be set when using TCP segmentation offload and MULR enabled. Recommend to always set to 1b.

Field	Bit(s)	Initial Value	Description
Reserved	26	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI**, must be set when enabling multiple transmit queues. Recommend to always set to 1b.
Reserved	30:27	0000b	Reserved. Always write to 0000b.
Reserved	31	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

13.3.70 Transmit Descriptor Base Address Low Queue 1

TDBAL1 (03900h; 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 13-132. TDBAL1 Register Bit Description

Field	Bit(s)	Initial Value	Description
ZERO	3:0	0b	Zero Value This field is ignored on writes and reads as 0b.
TDBAL1	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.

13.3.71 Transmit Descriptor Base Address High Queue 1

TDBAH1 (03904h; R/W)

This register contains the upper 32 bits of the 64-bit transmit Descriptor base address.

Table 13-133. TDBAH1 Register Bit Description

Field	Bit(s)	Initial Value	Description
TBDAH	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.

13.3.72 Transmit Descriptor Length Queue 1

TDLEN1 (03908h; 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 13-134. TDLEN1 Register Bit Description

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.

13.3.73 Transmit Descriptor Head Queue 1

TDH1 (03910h; 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 13-135. TDH1 Register Bit Description

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.

13.3.74 Transmit Descriptor Tail Queue 1

TDT1 (03918h; 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 13-136. TDT1 Register Bit Description

Field	Bit(s)	Initial Value	Description
TDT1	15:0	0b	Transmit Descriptor Tail
Reserved	31:16	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

13.3.75 Transmit Descriptor Control 1

TXDCTL1 (03928h; R/W)

This register controls the fetching and write back of transmit descriptors for queue 1. 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) based on the GRAN flag.

Table 13-137. TXDCTL1 Register Bit Description

Field	Bit(s)	Initial Value	Description
PTHRESH	5:0	00h	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 (queue 1). 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	00h	Reserved Reads as 0b. Should be written as 0b for future compatibility.
HTHRESH	15:8	00h	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	00h	Reserved Reads as 0b. Should be written as 0b for future compatibility.
WTHRESH	21:16	00h	Write Back Threshold WTHRESH controls the write back of processed transmit descriptors in queue 1. This threshold refers to the number of transmit descriptors in the Ethernet controller's on-chip buffer (queue 1) 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.
Reserved	22	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility. For the **82571EB/82572EI** and the **631xESB/632xESB**, always write to 1b.

Field	Bit(s)	Initial Value	Description
Reserved	23	0b	Reserved Reads as 0b. Should be written as 0b for future compatibility.
GRAN	24	0h	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.
LWTHRESH	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.

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 is set. Furthermore, with transmit descriptor bursting enabled, some descriptors are written back that did not have RS set in their respective descriptors.

LWTHRESH 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.

13.3.76 Transmit Arbitration Counter Queue 1

TARC1 (03940h; RW)

Field	Bit(s)	Initial Value	Description
COUNT	6:0	3	Transmit Arbitration Count The number of packets that can be sent from queue 1 to make the N over M arbitration between the queues. Writing 0h to this register is not allowed.
Reserved	7	0b	Reserved.
RATIO	9:8	00b	Compensation Ratio This value determines the ratio between the number of packets transmitted on queue 1 in a TCP segmentation offload to the number of compensated packets transmitted from queue 0. 00b = 1/1 compensation 01b = 1/2 compensation 10b = 1/4 compensation 11b = 1/8 compensation
ENABLE	10	1b	Descriptor Enable Always set to 1b.
Reserved	21:11	00h	Reserved Reads as 00h. Should be written to 00h for future compatibility.
Reserved	22	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI,** set to 1b when enabling TSO and MULR to workaround the issue described in Section 13.3.69.1.
Reserved	23	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI,** always write to 0b.
Reserved	24	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI,** must be set when enabling multiple transmit queues. Recommend to always set to 1b.
Reserved	25	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI,** must be set when enabling TCP segmentation offload and MULR enabled. Recommend to always set to 1b.
Reserved	26	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility. For the **82571EB/82572EI,** must be set when enabling multiple transmit queues. Recommend to always set to 1b.
Reserved	27	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

Reserved	28	0b	Must be set to 1b when TCTL.MULR is disabled and set to 0b when TCTL.MULR is enabled.
Reserved	30:29	00b	Reserved Always write to 0b.
Reserved	31	0b	Reserved Reads as 0b. Should be written to 0b for future compatibility.

13.4 Filter Registers

This section contains detailed descriptions for those registers associated with the Ethernet controller's address filter capabilities.

13.4.1 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 13-138. 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.

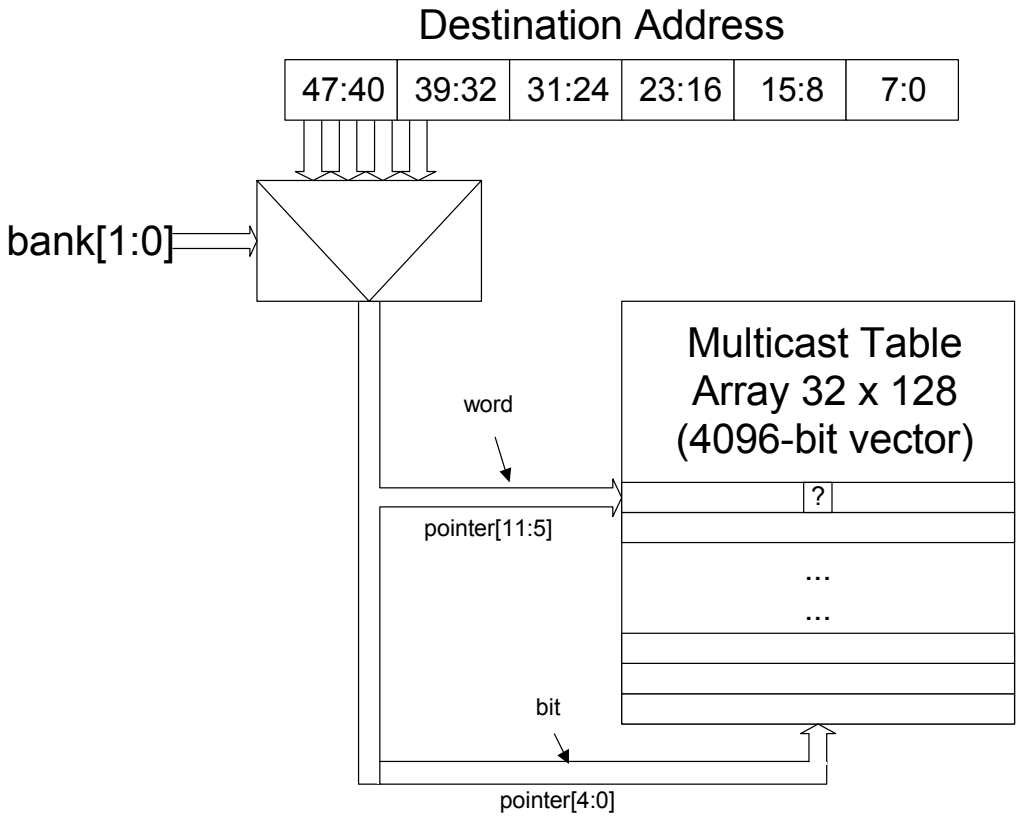


Figure 13-1. Multicast Table Array

13.4.2 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. The addresses stored in these registers are used for unicast/multicast address filtering.

Note: "n" is the exact unicast/multicast address entry and is equal to 0, 1, . . .15.

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 13-139. RAL Register Bit Description

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.

13.4.3 Receive Address High

RAH (05404h + 8∗n; R/W)

These registers contain the upper bits of the 48-bit Ethernet address. The complete address is {RAH, RAL}. AV determines whether this address is compared against the incoming packet. AV is also cleared by a master reset in entries 0-14 and on LAN_POWER_GOOD in entry 15 (reserved for managability). The addresses stored in these registers are used for unicast/multicast address filtering.

Note: "n" is the exact unicast/multicast address entry and is equal to 0, 1, . . .15.

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. When writing to this register, always write low-to-high. When clearing this register, always clear high-to-low.

Note: **L2 Address Filters** – Unicast entry 15 (in the RAH and RAL registers) is dedicated to the manageability block. This entry can be configured to compare either the L2 destination address or neither source address. If there is a match in entry 15 of the unicast table and the *EN_MAC16_FILTER* bit is set in the MANC register, the packet is directed to the manageability block. Unicast entry 15 is configured only by firmwere and should not be accessed by the software device driver (firmware should configure that entry before setting the *EN_MAC16_FILTER* bit).

Refer to the *Intel® 82571/82572/631xESB/632xESB GbE Controller TCO/System Manageability Interface* Application Note for more information.

Table 13-140. RAH Register Bit Description

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.
ASEL	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		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.

13.4.4 VLAN Filter Table Array

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 13.4.1 for a block diagram of the algorithm. If VLANs are not used, there is no need to initialize the VFTA.

Table 13-141. VFTA[127:0] Bit Description

Field	Bit(s)	Initial Value	Description
Bit Vector	31:0	X	Double-word wide bit vector specifying 32 bits in the VLAN Filter table.

13.4.5 Multiple Receive Queues Command Register

MRQC (05818h; R/W)

Table 13-142. MRQC Bit Description

Field	Bit(s)	Initial Value	Description
Multiple Receive Queues Enable	1:0	00h	Multiple Receive Queues Enable Enables support for Multiple Receive Queues and defines the mechanism that controls queue allocation. Note that the *RXCSUM.PCSD* bit must also be set to enable Multiple Receive Queues. 00b = Multiple Receive Queues are disabled. 01b = Multiple Receive Queues as defined by MSFT RSS. The RSS Field Enable bits define the header fields used by the hash function. 10b = Reserved 11b = Reserved
RSS Interrupt Enable	2	0h	RSS Interrupt Enable When set, this bit enables interrupt control by the RSS Interrupt Mask register. When cleared, a receive packet generates an interrupt indication independent of the RSS Interrupt registers.
Reserved	15:3	0h	Reserved.
RSS Field Enable	31:16	0h	Each bit, when set, enables a specific field selection to be used by the hash function. Several bits can be set at the same time. Bit[16] = Enable TcpIPv4 hash function Bit[17] = Enable IPv4 hash function Bit[18] = Enable TcpIPv6Ex hash function Bit[19] = Enable IPv6Ex hash function Bit[20] = Enable IPv6 hash function Bits[31:21] = Reserved

NOTES:
1. MRQC_EN is used for enable/disable RSS hashing and also for enabling multiple receive queues. Disabling this feature is not recommended. Model usage is to reset the Ethernet controller after disable the RSS.
2. In order to enable the IPv6Ex hash function, the IPv6 hash function must also be enabled.
3. Packet would be tagged as IPv6 if it is without any of the Home-Address-Option field and Routing-Header-Type-2 field. As a result, if a packet is tagged with IPv6 (type 5) in this case, the device driver would have to convert it to IPv6Ex (type 4).

13.4.6 RSS Interrupt Mask Register

RSSIM (05864h; R/W)

Table 13-143. RSSIM Register Bit Description

Field	Bit(s)	Initial Value	Description
RSS Interrupt Mask	31:0	0 . . 0h	Each bit in the RSS Interrupt Mask Register masks a corresponding bit in the CPU Vector register (a value of 0b masks the CPU Vector, while a value of 1b allows the CPU vector bit to be set). A bit in the RSS Interrupt Mask Register is cleared when the CPU Vector is read and the corresponding bit in the CPU Vector was set. It is also cleared when the corresponding bit in the CPU Vector is written with a 1b. A bit in the RSS Interrupt Mask Register is set by writing a 1b into it. Writing a 0b has not effect.

13.4.7 RSS Interrupt Request Register

RSSIR (05868h; R/W)

Table 13-144. RSSIR Register Bit Description

Field	Bit(s)	Initial Value	Description
RSS Interrupt Request	31:0	0 . . 0h	Each bit in the RSS Interrupt Request Register indicates that packets are pending processing for the respective CPU. A bit is set when a packet and its descriptor have been posted to memory and the appropriate timer expired. Writing a 1b to a bit clears it. Writing a 0b has no effect.

13.4.8 Redirection Table

RETA (05C00h - 05C7Fh; R/W)

The redirection table is a 128-entry table with each entry being 8 bits wide. Only the 5 LSBs of each entry are used to store the tag value. The table is configured through the following R/W registers.

Field	Bit(s)	Initial Value	Description
Entry 0	0/7:0	0 . . 0h	Determines the tag value and physical queue for index 0.
. . .			
Entry 127	31/31:24	0 . . 0h	Determines the tag value and physical queue for index 127.

13.4.9 RSS Random Key Register

RSSRK (05C80h - 05CA7h; R/W)

The RSS Random Key Register stores a 40 byte key used by the RSS hash function.

Field	Bit(s)	Initial Value	Description
Byte 0	0/7:0	0 . . 0h	Byte 0 of the RSS random key.
Byte 39	9/31:24	0 . . 0h	Byte 39 of the RSS random key.

13.5 Wakeup Registers

13.5.1 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#.

Field	Bit(s)	Initial Value	Description
APME	0	0b	Advance Power Management Enable If set to 1b, APM Wakeup is enabled. This bit is loaded from the EEPROM word 24h/14h (bit 10).
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 PCIe* configuration space.
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.
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.
Reserved	31:4	0b	Reserved Reads as 0b.

13.5.2 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#.

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	6	0b	Directed IPv4 Packet Wakeup Enable.
IPv6	7	0b	Directed IPv6 Packet Wakeup Enable.
Reserved	14:8	0b	Reserved. Set these bits to 0b.
NoTCO	15	0	Ignore TCO/management packets for wakeup.
FLX0	16	0b	Flexible Filter 0 Enable.

Field	Bit(s)	Initial Value	Description
FLX1	17	0b	Flexible Filter 1 Enable.
FLX2	18	0b	Flexible Filter 2 Enable.
FLX3	19	0b	Flexible Filter 3 Enable.

13.5.3 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.

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	6	0b	Directed IPv4 Packet Received.
IPv6	7	0b	Directed IPv6 Packet Received.
Reserved	15:8	0b	Reserved.
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.
Reserved	31:20	0b	Reserved.

13.5.4 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#.

Field	Bit(s)	Initial Value	Description
V40	0	0b	IPv4 Address 0 Valid.
V41	1	0b	IPv4 Address 1 Valid.
V42	2	0b	IPv4 Address 2 Valid.
V43	3	0b	IPv4 Address 3 Valid.
V60	16	0b	IPv6 Address 0 Valid.

13.5.5 IPv4 Address Table

IP4AT (05840h:05858h; R/W)

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
2	5848h	IPV4ADDR1
3	5850h	IPV4ADDR2
4	5858h	IPV4ADDR3

Field	Dword #	Address	Bit(s)	Initial Value	Description
IPV4ADDR0	0	5840h	31:0	X	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

13.5.6 IPv6 Address Table

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	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

13.5.7 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.

Field	Bit(s)	Initial Value	Description
LEN	11:0	X	Length of wakeup packet. (If jumbo frames[1] is enabled and the packet is longer than 2047 bytes this field is 2047.)

1. Not applicable to the **82573E/82573V**.

13.5.8 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.

Field	Bit(s)	Initial Value	Description
WUPD	31:0	X	Wakeup Packet Data

13.5.9 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.

Note: 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).

For the **631xESB/632xESB**, during initialization or after a system reset, the following bit must be set:

- Address 05F04h:
  - — Bit 20 = 0b

31:0
Reserved
Reserved
Reserved
Reserved
Reserved
Reserved

31:11	10:0
Reserved	Length 0
Reserved	Length 1
Reserved	Length 2
Reserved	Length 3
Reserved	Length TCO 0
Reserved	Length TCO 1

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
LEN TCO 0	8	5F20h[1]	10:0	0b (EEPROM)	Minimum Length for Flexible TCO0 Filter
LEN TCO 1	10	5F28h[1]	10:0	0b (EEPROM)	Minimum Length for Flexible TCO1 Filter

1. The default value of 5F20h and 5F28h can be loaded from the EEPROM at power up.

Note: The minimum length supported for the flexible filters and the flexible TCO filters is 2 bytes.

13.5.10 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
Reserved
Reserved
Reserved
Reserved
Reserved

31 4	3 0
Reserved	Byte 0 Mask
Reserved	Byte 1 Mask
Reserved	Byte 2 Mask
Reserved	Byte 126 Mask
Reserved	Byte 127 Mask

Field	Dword #	Address	Bit(s)	Initial Value	Description
MASK0	0	9000h	7:0	X	Mask for Filter [3:0] for Byte 0
MASK1	2	9008h	7:0	X	Mask for Filter [3:0] for Byte 2
MASK2	4	9010h	7:0	X	Mask for Filter [3:0] for Byte 3
...					
MASK127	254	93F8h	7:0	X	Mask for Filter [3:0] for Byte 127

13.5.11 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

13.6 MNG Register (82571EB)

13.6.1 Management Control Register

MANC (05820; R/W)

Field	Bit	Attr	Default	Description
Reserved	2:0	RW	00b	Reserved
EN_FLEXPORT0	3	RW	0b	Enable Flex UDP/TCP Port 0 Filtering When set, all Rx packets that have a UDP/TCP port that matches flex filter 0 are passed to the firmware block.
EN_FLEXPORT1	4	RW	0b	Enable Flex UDP/TCP Port 1 Filtering When set, all Rx packets that have a UDP/TCP port that matches flex filter 1 are passed to the firmware block.

Field	Bit	Attr	Default	Description
EN_FLEXPORT2	5	RW	0b	Enable Flex UDP/TCP port 2 filtering When this bit is set than all Rx packet that has a UDP/TCP port that matches flex filter 2 will passed to the MNG block.
FLEX_TCO1_FILTER_EN	6	RW	0b	Flex TCO Filter 1 Filtering Enable When set, the flex TCO filtering mechanism is enabled for filter 1.
FLEX_TCO0_FILTER_EN	7	RW	0b	Flex TCO Filter 0 Filtering Enable When set, the flex TCO filtering mechanism is enabled for filter 0.
RMCP_EN	8	RW	1b	Enable RMCP 026Fh Filtering Sends RMCP packets with a destination port of 026Fh that pass RMCP filtering to firmware.
0298_EN	9	RW	0b	Enable RMCP 0298h Filtering Sends RMCP packets with a destination port of 0298h that pass RMCP filtering to firmware.
Reserved	12:10	RW	0h	Reserved.
ARP_REQ_EN	13	RW	0b	Enable ARP Request Filtering Sends ARP request packets that pass ARP filtering to firmware.
NEIGHBOR_EN	14	RW	0b	Enable Neighbor Discovery Filtering Sends packets that pass Neighbor Discovery filtering to firmware.
ARP_RES_EN	15	RW	0b	Enable ARP Response Filtering Sends ARP response packets that pass ARP filtering to firmware.
TCO_RESET	16	RW	0b	TCO Reset Occurred Set to 1b on a TCO reset. This bit is only reset by LAN_PWR_GOOD.
RCV_TCO_EN	17	RW	0b	Receive TCO Packets Enabled When set, enables the receive flow from the wire to the firmware block.
BLK_Phy_Rst_On_IDE	18	RW	0b	Block PHY Reset and Power State Changes When this bit is set in one of the ports (LAN0 or LAN1), the PCI reset and power up changes are not reflected in the PHY. When this bit is set to 0b in both ports, the PCI reset and power changes trigger a PHY reset. This bit cannot be written unless the No_Phy_Rst_for_IDE EEPROM bit is set. This bit is reset upon LAN_PWR_GOOD reset and main power down.
RCV_ALL	19	RW	0b	Receive All Enable When set, all packets received from the wire are passed to the firmware block.

Field	Bit	Attr	Default	Description
EN_MAC16_FILTER	20	RW	0b	Enable MAC Address16 Filtering When set and the Rx packet matches MAC address 16 filtering, the packet passes to the firmware block.
EN_MNG2HOST	21	RW	0b	Enable Firmware Packets to Host Memory This bit enables the functionality of the MANC2H register. When set, the packets that are specified in the MANC2H registers are also sent to the host memory if they pass manageability filters.
EN_IP_ ADDRFILTER	22	RW	0b	Enable IP Address Filtering When set, only packets that matches IPv4 or IPV6 address filtering passes to the firmware block. **Note:** For IPv4 this refers only to IPV4ADDR0.
EN_XSUM_FILTER	23	RW	1b	Enable Xsum Filtering to Firmware When set, only packets that pass L3 and L4 checksum are sent to the firmware block.
BR_EN	24	RW	0b	Enable Broadcast Filtering When set, the filter sends broadcast packets to the firmware block.
Reserved	31:25	RW	0h	Reserved.

13.7 PCIe* Registers

13.7.1 PCIe* Control

GCR (05B00h; R/W)

Table 13-145. GCR Register Bit Description

Field	Bit(s)	Initial Value	Description
Disable_Timeout_ Mechanism	31	0b	If set, PCIe* timeout mechanism is disabled.
Self_Test_Result	30	0b	If set, self test result finished successfully.
GIO_Good_10s	29	0b	Force good PCIe* L0s training.
GIO_Dis_Rd_Err	28	0b	Disable running disparity error of PCIe* 108b decoders.
L1_Act_Without_ L0s_Rx	27	0b	If set, enables the Ethernet controller to enter ASPM L1 active without any correlation of L0s_rx. For the **631xESB/632xESB**, this bit is reserved and should be set to 0b.
L1_Entry_Latency (Read Only)	26:25	11b	Determines the idle time of the PCe* link in L0s state before initiating a transition to L1 state. Initial value is loaded from the EEPROM/NVM. 00b - 64 µs 01b - 256 µs 10b - 1 ms 11b - 4 ms For the **631xESB/632xESB**, these bits are reserved and must be set to 11b. See bit 23 for the **82573V**.
L0s_Entry_Lat	24	0b	L0s Entry Latency Set to 0b to indicate L0s entry latency is the same as L0s exit latency. Set to 1b to indicate L0s entry latency is (L0s exit latency/4). For the **631xESB/632xESB**, this is a reserved bit and should be set to 0b.
L1_Entry_Latency (MSB) (Read Only)	23	1b	For the **82573V**: [2:0] Latency (bits 26:25 and 23) 000b = 2 µs 001b = 8 µs 010b = 16 µs 011b = 32 µs 100b = 64 µs 101b = 256 µs 110b = 1 ms 111b = 4 ms (default) This is a reserved bit for all remaining Ethernet controllers. Always set to 0b.

Field	Bit(s)	Initial Value	Description
Reserved	22	1b	Reserved Must be set to 1b.
LogHeader	21	0b	For the **631xESB/632xESB**, when this bit is set to 1b, the header of the PCIe* packet that caused an error is logged into configuration space 11Ch-128h. When set to 0b, there is no header logged. When this bit is set, changing the header log order in the error reporting registers is required (**82573E/82573V/82573L**). This is a reserved bit for the **82571EB/82572EI**.
Reserved	20:9	0b	Reserved Always set to 0b.
Self_Test_Enable	8	0b	When set, firmware should perform a self test. This is a reserved bit for the **82573E/82573V/82573L**.
Elec_Idle_Ind1	7	0b	SerDes1 internal electrical idle indication - RO bit - Cleared (0b) when there is link, Set (1b) when there is no link (electrical idle). This is a reserved bit for the **82573E/82573V/82573L**.
Elec_Idle_Ind0	6	0b	SerDes0 internal electrical idle indication - RO bit - Cleared (0b) when there is link, Set (1b) when there is no link (electrical idle). This is a reserved bit for the **82573E/82573V/82573L**.
Txdscr_Nosnoop	5	0b	Transmit Descriptor Read No snoop indication. Read directly by transaction layer.
Txdscw_Nosnoop	4	0b	Transmit Descriptor Write No snoop indication. Read directly by transaction layer.
Txd_Nosnoop	3	0b	Transmit Data Read No snoop indication. Read directly by transaction layer.
Rxdscr_Nosnoop	2	0b	Receive Descriptor Read No snoop indication. Read directly by transaction layer.
Rxdscw_Nosnoop	1	0b	Receive Descriptor Write No snoop indication. Read directly by transaction layer.
Rxd_Nosnoop	0	0b	receive Data Write No snoop indication. Read directly by transaction layer.

Table 13-146. Function-Tag Register Bit Description

Field	Bit(s)	Initial Value	Description
CNT_3_FUNC	31:29	0b	Function number for event 6/1D, if located in counter 3.
CNT_3_TAG	28:24	0b	Tag number for event 6/1D, if located in counter 3.
CNT_2_FUNC	23:21	0b	Function number for event 6/1D, if located in counter 2.
CNT_2_TAG	20:16	0b	Tag number for event 6/1D, if located in counter 2.
CNT_1_FUNC	15:13	0b	Function number for event 6/1D, if located in counter 1.
CNT_1_TAG	12:8	0b	Tag number for event 6/1D, if located in counter 1.
CNT_0_FUNC	7:5	0b	Function number for event 6/1D, if located in counter 0.
CNT_0_TAG	4:0	0b	Tag number for event 6/1D, if located in counter 0.

13.7.2 PCIe* Statistics Control #1 GSCL_1 (05B10h; R/W)

Table 13-147. GSCL_1 Register Bit Description

Field	Bit(s)	Initial Value	Description
GIO_COUNT_START	31	0b	Start indication of PCIe* statistic counters.
GIO_COUNT_STOP	30	0b	Stop indication of PCIe* statistic counters.
GIO_COUNT_RESET	29	0b	Reset indication of PCIe* statistic counters.
GIO_64_BIT_EN	28	0b	Enable two 64-bit counters instead of four 32-bit counters.
GIO_COUNT_TEST	27	0b	Test Bit Firmware counters for testability.
Reserved	26:4	0b	Reserved.
GIO_COUNT_EN_3	3	0b	Enable PCIe* Statistic Counter Number 3.
GIO_COUNT_EN_2	2	0b	Enable PCIe* Statistic Counter Number 2.
GIO_COUNT_EN_1	1	0b	Enable PCIe* Statistic Counter Number 1.
GIO_COUNT_EN_0	0	b0	Enable PCIe* Statistic Counter Number 0.

13.7.3 PCIe* Statistics Control #2

GSCL_2 (05B14h; R/W)

This counter contains the mapping of an event (which counter counts what event).

Table 13-148. GSCL_2 Register Bit Description

Field	Bit(s)	Initial Value	Description
GIO_EVENT_NUM_3	31:24	0b	Event number that counter 3 counts.
GIO_EVENT_NUM_2	23:16	0b	Event number that counter 2 counts.
GIO_EVENT_NUM_1	15:8	0b	Event number that counter 1 counts.
GIO_EVENT_NUM_0	7:0	0b	Event number that counter 0 counts.

Table 13-149 lists the encoding of the events.

Table 13-149. Event Encodings

Transaction Layer Events	Event Mapping (Hex)	Description
Dwords of transaction layer packet transmitted (transferred to the physical layer), include payload and header.	0	Each 125 MHz cycle, the counter increases by 1 (1 dw) or 2 (2 dw). Counted: completion, memory, message (not replied)
All types of transmitted packets.	1	Only TLP packets. Each cycle, the counter increases by 1 if TLP packet was transmitted to the link. Counted: completion, memory, message (not replied)
Transmit TLP Packets of function #0	2	Each cycle, the counter increases by 1, if the packet was transmitted. Counted: memory, message of function 0 (not replied)
Transmit TLP Packets of function #1	3	Each cycle, the counter increases by 1, if the packet was transmitted Counted: memory, message of function 1 (not replied)
Non posted Transmit TLP packets of function #0	4	Each cycle, the counter increases by 1, if the packet was transmitted Counted: memory (np) of function 0 (not replied)
Non posted Transmit TLP packets of function #1	5	Each cycle, the counter increases by 1, if the packet was transmitted Counted: memory (np) of function 1 (not replied)
Transmit TLP Packets of function X and tag Y, according to FUNC_TAG register	6	Each cycle, the counter increases by 1, if the packet was transmitted Counted: memory, message for a given func# and tag# (not replied)
All types of received packets (TLP only)	1A	Each cycle, the counter increases by 1, if the packet was received Counted: completion (only good), memory, I/O, config
Receive TLP Packets of function #0	1B	Each cycle, the counter increases by 1, if the packet was transmitted. Counted: good completions of func#0
Receive TLP Packets of function #1	1C	Each cycle, the counter increases by 1, if the packet was transmitted Counted: good completions of func#1
Receive Completion Packets	1D	Each cycle, the counter increases by 1, if the packet was received. Counted: good completions for a given func# and tag#
Clock counter	20	Counts GIO cycles.
Bad TLP from LL	21	Each cycle, the counter increases by 1, if bad TLP is received (bad CRC, error reported by AL, misplaced special char, reset in THL of received TLP).

Transaction Layer Events	Event Mapping (Hex)	Description
Header dwords of Transaction layer packet transmitted.	25	Only TLP, each 125 MHz cycle the counter increases by 1 (1 dw of header) or 2 (2 dw of header). Counted: completion, memory, message (not replied)
Header dwords of Transaction layer packet received.	26	Only TLP, each 125 MHz cycle the counter increases by 1 (1 dw of header) or 2 (2 dw of header). Counted: completion, memory, message
Transaction layer stalls transmitting due to lack of flow control credits of the next part.	27	The counter counts the number of times Transaction layer Stop transmitting because of this (per packet). Counted: completion, memory, message
Retransmitted packets.	28	The Counter increases for each retransmitted packet. Counted: completion, memory, message
Stall due to retry buffer full	29	The counter counts the number of times Transaction layer stop transmitting because Retry buffer is full (per packet). Counted: completion, memory, message
Retry buffer is under threshold	2A	Threshold specified by software, Retry buffer is under threshold per packet. Counted: completion, memory, message
PRH (posted request header) flow control credits (of the next part) below threshold	2B	Threshold specified by software. The counter increases each time number of the specific flow control credits is lower than threshold. Counted: According to credit type
PRD (posted request data) flow control credits (of the next part) below threshold	2C	
NPRH (non posted request header) flow control credits (of the next part) below threshold	2D	
CPLH (completion header) flow control credits (of the next part) below threshold	2E	
CPLD (completion data) flow control credits (of the next part) below threshold	2F	
PRH (posted request header) flow control credits (of local part) get to 0.	30	Threshold specified by software. The counter increases each time number of the specific flow control credits is get the value 0 (The period that the credit is 0 not counted). Counted: According to credit type
NPRH (non posted request header) flow control credits (of local part) get to 0.	31	
PRD (posted request data) flow control credits (of local part) get to 0.	32	

Transaction Layer Events	Event Mapping (Hex)	Description
NPRD (non posted request data) flow control credits (of local part) get to 0.	33	
Dwords of Transaction layer packet received, include payload and header.	34	Each 125 MHz cycle the counter increases by 1 (1 dw) or 2 (2 dw). Counted: completion, memory, message, I/O, config
Messages packets received	35	Each 125 MHz cycle the counter increases by 1 Counted: messages (only good)
Received packets to func_logic.	36	Each 125 MHz cycle the counter increases by 1 Counted: memory, I/O, config (only good)
Host Arbiter Events	**Event Mapping (Hex)**	**Description**
Average latency of read request – from initialization until end of completions. Estimated latency is ~5 μs.	40 + 41	The software will select the client need to be tested. The statistic counter will count the number of read request of the required client. In addition, the accumulated time of all requests will be saved in a time accumulator. The average time for read request will be [Accumulated time / Number of read requests] (Event 41 is for the counter)
Average latency of read request RTT– from initialization until the first completion is arrived (Round Trip Time). Estimated latency is 1uSec	42 + 43	The software will select the client need to be tested. The statistic counter will count the number of read request of the required client. In addition, the accumulated time of all RTT will be saved in a time accumulator. The average time for read request will be [Accumulated time / Number of read requests] (Event 43 is for the counter)
Requests that reached Time Out.	44	Number of requests that reached Time Out.
Completion Latency above Threshold	45 + 46	The software will select the client need to be tested. The software will program the required threshold (in GSCL_4 – units of 96 ns). One statistic counter will count the time from the beginning of the request until end of completions. The other counter will count the number of events. If the time is above threshold – add 1 to the event counter. (Event 46 is for the counter)

Host Arbiter Events	Event Mapping (Hex)	Description
Completion Latency above Threshold – for RTT	47 + 48	The software will select the client need to be tested. The software will program the required threshold (in GSCL_4 – units of 96 ns). One statistic counter will count the time from the beginning of the request until first completion arrival. The other counter will count the number of events. If the time is above threshold – add 1 to the event counter. (Event 48 is for the counter)
Link Layer Events	**Event Mapping (Hex)**	**Description**
Dwords of packet transmitted (transferred to the physical layer), include payload and header.	50	Include DLLP (Link layer packets) and TLP (transaction layer packets transmitted. Each 125 MHz cycle the counter increases by 1 (1 dw) or 2 (2 dw).
Dwords of packet received (transferred to the physical layer), include payload and header.	51	Include DLLP (Link layer packets) and TLP (transaction layer packets transmitted. Each 125 MHz cycle the counter increases by 1 (1 dw) or 2 (2 dw).
All types of DLLP packets transmitted from link layer.	52	Each cycle, the counter increases by 1, if DLLP packet was transmitted.
Flow control DLLP transmitted from link layer.	53	Each cycle, the counter increases by 1, if message was transmitted
Ack DLLP transmitted.	54	Each cycle, the counter increases by 1, if message was transmitted.
All types of DLLP packets received.	55	Each cycle, the counter increases by 1, if DLLP was received.
Flow control DLLP received in Link layer.	56	Each cycle, the counter increases by 1, if message was received.
Ack DLLP received.	57	Each cycle, the counter increases by 1, if message was received.
Nack DLLP received.	58	Each cycle, the counter increases by 1, if message was transmitted

13.7.4 PCIe* Statistics Control #3

GSCL_3 (05B18h; R/W)

This counter holds the threshold values needed for some of the event counting. The event increases only after the value passes the threshold boundary.

Table 13-150. GSCL_3 Register Bit Description

Field	Bit(s)	Initial Value	Description
Reserved	31:28	0b	Reserved.
GIO_FC_TH_1	27:16	0b	Threshold of flow control credits. Optional values: 0 - (256-1).
Reserved	15:12	0b	Reserved.
GIO_FC_TH_0	11:0	0b	Threshold of flow control credits. Optional values: 0 - (256-1).

13.7.5 PCIe* Statistics Control #4

GSCL_4 (05B1Ch; R/W)

This counter holds the threshold values needed for some of the event counting. The event increases only after the value passes the threshold boundary.

Table 13-151. GSCL_4 Register Bit Description

Field	Bit(s)	Initial Value	Description
Reserved	31:16	0b	Reserved.
GIO_RB_TH	15:10	0b	Retry buffer threshold.
GIO_COML_TH	9:0	0b	Completions latency threshold.

13.7.6 PCIe* Counter #0

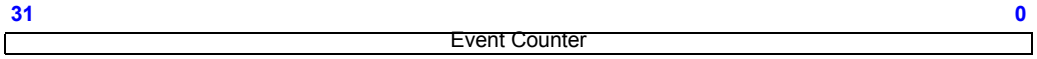
GSCN_0 (05B20h; R/W)

Table 13-152. GSCN_0 Register Bit Description

13.7.7 PCIe* Counter #1

GSCN_1 (05B24h; R/W)

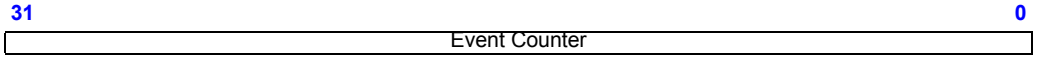
Table 13-153. GSCN_1 Register Bit Description



13.7.8 PCIe* Counter #2

GSCN_2 (05B28h; R/W)

Table 13-154. GSCN_2 Register Bit Description



13.7.9 PCIe* Counter #3

GSCN_3 (05B2Ch; R/W)

Table 13-155. GSCN_3 Register Bit Description

13.7.10 Function Active and Power State to MNG

FACTPS (05B30h; RO)

Firmware uses this register for configuration.

Table 13-156. Register Bit Description

Field	Bit(s)	Initial Value	Description
PM State Changed	31	0b	Indication that one or more of the functions power states had changed. This bit is also a signal to the MNG unit to create an interrupt. This bit is cleared on read, and is not set for at least 8 cycles after it was cleared.
LAN Function Sel	30	0b	When both LAN ports are enabled and the LAN Function Sel equals 0b, LAN 0 is routed to PCIe* Function 0 and LAN 1 is routed to PCIe* Function 1. If the LAN Function Sel equals 1b, LAN 0 is routed to PCIe* Function 1 and LAN 1 is routed to PCIe* Function 0. If any of the LAN functions are disabled, the other one is routed to PCIe* Function 0 regardless of the LAN Function Sel. This bit is initiated by EEPROM word 21h. This is a reserved bit for the **82573E/82573V/82573L**.
MNGCG	29	0b	MNG Clock Gated When set, indicates that the manageability clock is gated.
Reserved	28	0b	Reserved.
Func4 Aux_En	27	0b	Function 4 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
IPMI Enable KCS Enable (**82573E/ 82573V/82573L**)	26	0b	IPMI/KCS Enable When set to 0b, indicates that the IDE at function 2 is disabled. When the function is enabled, the bit is set to 1b. This bit is initiated by EEPROM word 21h. For the **82573E/82573V/82573L**, When set to 0b, indicates that the KCS at function 4 is disabled. When the function is enabled, the bit is set to 1b. This bit is initiated by NVM word 49h.
Func4 Power State	25:24	0b	Power state indication of Function 4.
Reserved	23:22	0b	Reserved.
Func3 Aux_En	21	0b	Function 3 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
SP Enable	20	0b	SP Enable When set to 0b, it indicates that the IDE at function 2 (function 3 for the **82573E/82573V/82573L**) is disabled. When the function is enabled, the bit is set to 1b. This bit is initiated by EEPROM word 21h (NVM word 49h for the **82573E/82573V/82573L**).
Func3 Power State	19:18	0b	Power state indication of Function 3.
Reserved	17:16	0b	Reserved.
Func2 Aux_En	15	0b	Function 2 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
IDE Enable	14	0b	IDE Enable When set to 0b, it indicates that the IDE at function 2 is disabled. When the function is enabled, the bit is set to 1b. This bit is initiated by EEPROM word 21h (NVM word 49h for the **82573E/ 82573V/82573L**).
Func2 Power State	13:12	0b	Power state indication of Function 2.
Reserved	11:10	0b	Reserved.

Field	Bit(s)	Initial Value	Description
LAN_1_EN	10	0b	For the **631xESB/632xESB** only: 1b = Indicates that LAN1 is enabled. Set by EEPROM bit (LAN_PCI_DIS being 0b - bit 10 in word 10h) ANDed with LAN1 Valid bit.
Func1 Aux_En	9	0b	Function 1 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space. This is a reserved bit for the **82573E/82573V/82573L**.
LAN1 Valid	8	0b	LAN 1 Enable When set to 0b, it indicates that the LAN 0 function is disabled. When the function is enabled, the bit is set to 1b. The LAN 0 enable is set by the LAN 1 Enable / TEST_POINT[3] strapping pin. For the **82573E/82573V/82573L**, hardwired to 0b.
Func1 Power State	7:6	00b	Power state indication of Function 1 00b -> DR 01b -> D0u 10b -> D0a 11b -> D3 These are reserved bits for the **82573E/82573V/82573L**.
Reserved	5:4	0b	Reserved.
LAN_0_EN	4	0b	For the **631xESB/632xESB** only: 1b = Indicates that LAN0 is enabled. Set by EEPROM bit (LAN_PCI_DIS being 0b - bit 10 in word 20h) ANDed with LAN0 Valid bit.
Func0 Aux_En	3	0b	Function 0 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
LAN0 Valid	2	0b	LAN 0 Enable When set to 0b, it indicates that the LAN 0 function is disabled. When the function is enabled, the bit is set to 1b. For the **82571EB/82572EI**, the LAN 0 enable is set by the LAN 0 Enable / TEST_POINT[2] strapping pin. For the **631xESB/632xESB**, this bit is set by the LAN_DIS being 0b (EEPROM, word 20h, bit 11). For the **82573E/82573V/82573L**, hardwired to 1b.
Func0 Power State	1:0	00b	Power state indication of Function 0 00b -> DR 01b -> D0u 10b -> D0a 11b -> D3

Note: For the **631xESB/632xESB** and the **82571EB/82572EI**, all the function enable bits and the power state fields are not valid when the system is in S5. If firmware wants to read the function enable fields when the system is in S5, it should read them from the EEPROM.

13.7.11 SerDes/CCM/PCIe* CSR (82571EB/82572EI) GIOANACTL0 (05B34h; R/W)

Firmware uses this register for analog circuit configuration.

Table 13-157. GIOANACTL0 Register Bit Description

Field	Bit(s)	Initial Value	Description
Done Indication	31	1b	When a write operation completes this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.
Reserved	30:16	0b	Reserved.
Address	15:8	0b	Address to SerDes.
Data	7:0	0b	Data to SerDes.

13.7.12 SerDes/CCM/PCIe* CSR (82571EB/82572EI) GIOANACTL1 (05B38h; R/W)

Firmware uses this register for analog circuit configuration.

Table 13-158. GIOANACTL1 Register Bit Description

Field	Bit(s)	Initial Value	Description
Done Indication	31	1b	When a write operation completes this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.
Reserved	30:16	0b	Reserved.
Address	15:8	0b	Address to SerDes.
Data	7:0	0b	Data to SerDes.

GIOANACTL2 (05B3Ch; R/W)

Firmware uses this register for analog circuit configuration.

Table 13-159. GIOANACTL2 Register Bit Description

Field	Bit(s)	Initial Value	Description
Done Indication	31	1b	When a write operation completes this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.
Reserved	30:16	0b	Reserved.
Address	15:8	0b	Address to SerDes.
Data	7:0	0b	Data to SerDes.

GIOANACTL3 (05B40h; R/W)

Firmware uses this register for analog circuit configuration.

Table 13-160. GIOANACTL3 Register Bit Description

Field	Bit(s)	Initial Value	Description
Done Indication	31	1b	When a write operation completes this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.
Reserved	30:16	0b	Reserved.
Address	15:8	0b	Address to SerDes.
Data	7:0	0b	Data to SerDes.

13.7.13 SerDes/CCM/PCIe* CSR (82571EB/82572EI) GIOANACTLALL (05B44h; R/W)

Firmware uses this register for analog circuit configuration.

Table 13-161. GIOANACTLALL Register Bit Description

Field	Bit(s)	Initial Value	Description
Done Indication	31	1b	When a write operation completes this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.
Reserved	30:16	0b	Reserved.
Address	15:8	0b	Address to SerDes.
Data	7:0	0b	Data to SerDes.

13.7.14 SerDes/CCM/PCIe* CSR (82571EB/82572EI) CCMCTL (05B48h; R/W)

Firmware uses this register for analog circuit configuration.

Table 13-162. CCMCTL Register Bit Description

Field	Bit(s)	Initial Value	Description
Done Indication	31	1b	When a write operation completes this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.
Reserved	30:16	0b	Reserved.
Address	15:8	0b	Address to SerDes.
Data	7:0	0b	Data to SerDes.

13.7.15 Analog Control Register (631xESB/632xESB) ANACTL (05B48h; R/W)

Software or firmware uses this register for GLCI/SerDes analog circuit configuration. To write to one of these registers, software performs the following steps:

1. Wait until the *Done* bit is set to 1b.
2. Write 1b in *Write* bit (bit 30), the address of the register in the address field (bits 15-8) and the data to be written in the lower bits of the register (bits 7-0).

To read from one of the analog registers, software performs the following steps:

1. Write 0b in *Write* bit (bit 30), the address of the register in the address field (bits 15-8). This causes the address to be latched in an internal register.
2. Check the *Done* bit to make sure it is set to 1b.
3. Read the same address; the requested data resides on bits 7-0.

Note: These registers can be accessed by both software and firmware. As a result, developers need to implement a careful arbitration in order to prevent collisions between accesses from both sides.

Table 13-163. ANACTL Register Bit Description

Field	Bit(s)	Initial Value	Description
Done Indication	31	1b	During a write, this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by a new data. During a read, this bit is set to 1b indicating that the read data is valid.
Write Command	30	0b	1b = Write. 0b = Read.
Reserved	29:16	0b	Reserved.
Address	15:8	0b	Address of Analog register.
Data	7:0	0b	Read and write data from/to Analog register.

13.7.16 SerDes/CCM/PCIe* CSR (82571EB/82572EI)

SCCTL (05B4Ch; R/W)

Firmware uses this register for analog circuit configuration.

Table 13-164. SCCTL Register Bit Description

Field	Bit(s)	Initial Value	Description
Done Indication	31	1b	When a write operation completes this bit is set to 1b indicating that new data can be written. This bit is over written to 0b by new data.
Reserved	30:16	0b	Reserved.
Address	15:8	0b	Address to SerDes.
Data	7:0	0b	Data to SerDes.

13.7.17 Software Semaphore Register

SWSM (5B50h; R/W)

Firmware does not use this register directly. Instead, hardware uses it for synchronizing software/ firmware flows.

Field	Bit(s)	Initial Value	Description
Reserved	31:3	0h	Reserved
WMNG (SC)	2	0h	Wake MNG clock When this bit is set, hardware wakes the MNG clock (if gated). Asserting this bit does not clear the CFG_DONE bit in the EEMNGCTL register. This bit is self cleared on writes.
SWESMBI	1	0b	Software EEPROM Semaphore Bit This bit should be set only by the device driver (read only to firmware). The bit is not set if bit 0 in the FWSM register is set. The device driver should set this bit and then read it to see if it was set. If it was set, it means that the device driver can read/ write from/to the EEPROM. The device driver should clear this bit after completing EEPROM access. Hardware clears this bit on GIO soft reset.
SMBI (RS)	0	0b	Semaphore Bit This bit is set by hardware when this register is read by the device driver and cleared when the host driver writes a 0b to it. The first time this register is read, the value is 0b. In the next read the value is 1b (hardware mechanism). The value remains 1b until the software device driver clears it. This bit can be used as a semaphore between the two device's drivers in the GbE controller. This bit is cleared on GIO soft reset.

13.7.18 Firmware Semaphore Register

FWSM (B54h)

Firmware uses this register mainly to provide external indications to the host about its state. The only exception to this is bit 0, which is part of a hardware mechanism that synchronizes between software and firmware access to the EEPROM.

Field	Bit(s)	Initial Value	Description
Reserved	31:25	0h	Reserved
Ext_Err_Ind	24:19	0h	External error indication Firmware writes here the reason that the firmware has reset / clock gated. For example, EEPROM, flash, patch corruption, etc. Possible values: 00h: No Error 01h: Bad checksum in manageability sections 02h: EEPROM protection disabled 03h: Clock off command executed 04h: Bad checksum in Flash header 05h: Bad checksum in patch section 06h: Bad checksum in loader section 07h: Bad checksum in diagnostic section 08h: Invalid firmware mode 09h: TLB table exceeded 0Ah: EEPROM read failed 0Bh: Bad hardware version in patch load 0Ch: SFlash read failed 0Dh: Unspecified error 0Eh: Flash authentication failed 0Fh: PFlash read failed 10h: Invalid Flash entry point 11h:3Fh: Reserved.
Reset_Cnt	18:16	0h	Reset Counter Firmware increments the count at every reset.
FW_Val_Bit	15	0h	Firmware Valid Bit Hardware clears this bit in reset de-assertion so software can know firmware mode (bits 1-5) is invalid. Firmware should set this bit to 1b when it is ready (end of boot sequence).
Reserved	14:7	0h	Reserved
EEP_Reload_ Ind	6	0h	EEPROM reloaded indication Set to 1b after firmware reloads the EEPROM.
SOL On	5	0b	SOL On Set to 1b by firmware when SOL is on.
IDE On	4	0b	IDE On Set to 1b by firmware when IDE redirection is on.

Field	Bit(s)	Initial Value	Description
FW_Mode	3:1	0h	Firmware Mode Indicates the firmware mode as follows: 000b = No MNG. 010b = PT mode. 100b = Host Interface enable only.
EEP_FW_ Semaphore	0	0h	EEPROM Firmware Semaphore Firmware should set this bit to 1b before accessing the EEPROM. If software using the SWSM does not lock the EEPROM, firmware is able to set this bit to 1b. Firmware should set this bit to 0b after completing EEPROM access.

13.8 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.”

13.8.1 CRC Error Count

CRCERRS (04000h; RC)

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 13-165. CRCERRS Register Bit Description

Field	Bit(s)	Initial Value	Description
CEC	31:0	0b	CRC error count

13.8.2 Alignment Error Count

ALGNERRC (04004h; RC)

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 13-166. ALGNERRC Register Bit Description

Field	Bit(s)	Initial Value	Description
AEC	31:0	0b	Alignment error count

13.8.3 Symbol Error Count[1]

SYMERRS (04008h; RC)

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.

Table 13-167. SYMERRS Register Bit Description

Field	Bit(s)	Initial Value	Description
SYMERRS	31:0	0b	Symbol Error Count

13.8.4 RX Error Count

RXERRC (0400Ch; RC)

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, this register increments on the reception of /V/ codes.

1. Not applicable to the **631xESB/632xESB**.

Table 13-168. RXERRC Register Bit Description

Field	Bit(s)	Initial Value	Description
RXEC	31:0	0b	RX error count

13.8.5 Missed Packets Count

MPC (04010h; RC)

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 13-169. MPC Register Bit Description

Field	Bit(s)	Initial Value	Description
MPC	31:0	0b	Missed Packets Count

13.8.6 Single Collision Count

SCC (04014h; RC)

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 13-170. SCC Register Bit Description

Field	Bit(s)	Initial Value	Description
SCC	31:0	0b	Number of times a transmit encountered a single collision.

13.8.7 Excessive Collisions Count

ECOL (04018h; RC)

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 13-171. ECOL Register Bit Description

Field	Bit(s)	Initial Value	Description
ECC	31:0	0b	Number of packets with more than 16 collisions.

13.8.8 Multiple Collision Count

MCC (0401Ch; RC)

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 13-172. MCC Register Bit Description

Field	Bit(s)	Initial Value	Description
MCC	31:0	0b	Number of times a successful transmit encountered multiple collisions.

13.8.9 Late Collisions Count

LATECOL (04020h; RC)

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 13-173. LATECOL Register Bit Description

Field	Bit(s)	Initial Value	Description
LATECOL	31:0	0b	Number of packets with late collisions.

13.8.10 Collision Count

COLC (04028h; RC)

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 13-174. COLC Register Bit Description

Field	Bit(s)	Initial Value	Description
COLC	31:0	0b	Total number of collisions experienced by the transmitter.

13.8.11 Defer Count

DC (04030h; RC)

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 13-175. DC Register Bit Description

Field	Bit(s)	Initial Value	Description
DC	31:0	0b	Number of defer events.

13.8.12 Transmit with No CRS

TNCRS (04034h; RC)

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 mode, and is only valid when the Ethernet controller is operating at full duplex.

Table 13-176. TNCRS Register Bit Description

Field	Bit(s)	Initial Value	Description
TNCRS	31:0	0b	Number of transmissions without a CRS assertion from the PHY.

13.8.13 Sequence Error Count

SEC (04038h; RC)

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.

Table 13-177. SEC Register Bit Description

Field	Bit(s)	Initial Value	Description
SEC	31:0	0b	Number of sequence error events.

13.8.14 Carrier Extension Error Count (631xESB/632xESB)

CEXTERR (0403Ch; RC)

This register counts the number of packets received in which the carrier extension error was signaled across the 1000BASE-T 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 while RX_ER is asserted to the MAC. This register only increments if receives are enabled and the **631xESB/632xESB** is operating at 1000 Mb/s.

Table 13-178. CEXTERR Register Bit Description

Field	Bit(s)	Initial Value	Description
CEXTERR	31:0	0b	Number of packets with a carrier extension error.

13.8.15 Receive Length Error Count

RLEC (04040h; RC)

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 13-179. RLEC Register Bit Description

Field	Bit(s)	Initial Value	Description
RLEC	31:0	0b	Number of packets with receive length errors.

13.8.16 XON Received Count

XONRXC (04048h; RC)

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 13-180. XONRXC Register Bit Description

Field	Bit(s)	Initial Value	Description
XONRXC	31:0	0b	Number of XON packets received.

13.8.17 XON Transmitted Count

XONTXC (0404Ch; RC)

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 13-181. XONTXC Register Bit Description

Field	Bit(s)	Initial Value	Description
XONTXC	31:0	0b	Number of XON packets transmitted.

13.8.18 XOFF Received Count

XOFFRXC (04050h; RC)

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 13-182. XOFFRXC Register Bit Description

Field	Bit(s)	Initial Value	Description
XOFFRXC	31:0	0b	Number of XOFF packets received.

13.8.19 XOFF Transmitted Count

XOFFTXC (04054h; RC)

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 13-183. XOFFTXC Register Bit Description

Field	Bit(s)	Initial Value	Description
XOFFTXC	31:0	0b	Number of XOFF packets transmitted.

13.8.20 FC Received Unsupported Count

FCRUC (04058h; RC)

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 13-184. FCRUC Register Bit Description

Field	Bit(s)	Initial Value	Description
FCRUC	31:0	0b	Number of unsupported flow control frames received.

13.8.21 Packets Received (64 Bytes) Count

PRC64 (0405Ch; RC)

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 13-185. PRC64 Register Bit Description

Field	Bit(s)	Initial Value	Description
PRC64	31:0	0b	Number of packets received that are 64 bytes in length.

13.8.22 Packets Received (65-127 Bytes) Count

PRC127 (04060h; RC)

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 13-186. PRC127 Register Bit Description

Field	Bit(s)	Initial Value	Description
PRC127	31:0	0b	Number of packets received that are 65-127 bytes in length.

13.8.23 Packets Received (128-255 Bytes) Count

PRC255 (04064h; RC)

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 13-187. PRC225 Register Bit Description

Field	Bit(s)	Initial Value	Description
PRC255	31:0	0b	Number of packets received that are 128-255 bytes in length.

13.8.24 Packets Received (256-511 Bytes) Count

PRC511 (04068h; RC)

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 13-188. PRC551 Register Bit Description

Field	Bit(s)	Initial Value	Description
PRC511	31:0	0b	Number of packets received that are 256-511 bytes in length.

13.8.25 Packets Received (512-1023 Bytes) Count

PRC1023 (0406Ch; RC)

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 13-189. PRC1023 Register Bit Description

Field	Bit(s)	Initial Value	Description
PRC1023	31:0	0b	Number of packets received that are 512-1023 bytes in length.

13.8.26 Packets Received (1024 to Max Bytes) Count

PRC1522 (04070h; RC)

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 13.8.36). 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 13-190. PRC1522 Register Bit Description

Field	Bit(s)	Initial Value	Description
PRC1522	31:0	0b	Number of packets received that are 1024-Max bytes in length.

13.8.27 Good Packets Received Count

GPRC (04074h; RC)

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 13.8.36). 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 13-191. GPRC Register Bit Description

Field	Bit(s)	Initial Value	Description
GPRC	31:0	0b	Number of good packets received (of any length).

13.8.28 Broadcast Packets Received Count

BPRC (04078h; RC)

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. This register does not count packets counted by the Missed Packet Count (MPC) register.

Table 13-192. BPRC Register Bit Description

Field	Bit(s)	Initial Value	Description
BPRC	31:0	0b	Number of broadcast packets received.

13.8.29 Multicast Packets Received Count

MPRC (0407Ch; RC)

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 13-193. MPRC Register Bit Description

Field	Bit(s)	Initial Value	Description
MPRC	31:0	0b	Number of multicast packets received.

13.8.30 Good Packets Transmitted Count

GPTC (04080h; RC)

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 13-194. GPTC Register Bit Description

Field	Bit(s)	Initial Value	Description
GPTC	31:0	0b	Number of good packets transmitted.

13.8.31 Good Octets Received Count

GORCL (04088h; RC)/GORCH (0408Ch; RC)

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 13-195. GORCL and GORCH Register Bit Description

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.

13.8.32 Good Octets Transmitted Count

GOTCL (04090h; RC)/ GOTCH (04094; RC)

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 13-196. GOTCL and GOTCH Register Bit Description

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.

13.8.33 Receive No Buffers Count

RNBC (040A0h; RC)

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 13-197. RNBC Register Bit Description

Field	Bit(s)	Initial Value	Description
RNBC	31:0	0b	Number of receive no buffer conditions.

13.8.34 Receive Undersize Count

RUC (040A4h; RC)

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 13-198. RUC Register Bit Description

Field	Bit(s)	Initial Value	Description
RUC	31:0	0b	Number of receive undersize errors.

13.8.35 Receive Fragment Count

RFC (040A8h; RC)

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 13-199. RFC Register Bit Description

Field	Bit(s)	Initial Value	Description
RFC	31:0	0b	Number of receive fragment errors.

13.8.36 Receive Oversize Count

ROC (040ACh; RC)

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 13-200. ROC Register Bit Description

Field	Bit(s)	Initial Value	Description
ROC	31:0	0b	Number of receive oversize errors.

13.8.37 Receive Jabber Count

RJC (040B0h; RC)

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 13-201. RJC Register Bit Description

Field	Bit(s)	Initial Value	Description
RJC	31:0	0b	Number of receive jabber errors.

13.8.38 Management Packets Received Count

MPRC (040B4h; RC)

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.

Field	Bit(s)	Initial Value	Description
MPRC	31:0	0b	Number of management packets received.

13.8.39 Management Packets Dropped Count

MPDC (040B8h; RC)

This register counts the total number of packets received that pass the management filters (as described in the appropriate System Management Bus Interface Application Notes) and then are dropped because the management receive FIFO is full. Management packets include any packet directed to the BMC, RMCP and ARP packets.

Field	Bit(s)	Initial Value	Description
MPDC	31:0	0b	Number of management packets dropped.

13.8.40 Management Packets Transmitted Count

MPTC (040BCh; RC)

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.

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0b	Number of management packets transmitted.

13.8.41 Total Octets Received

TORL (040C0h; RC)/TORH (040C4h; RC)

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.

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 13-202. TORL and TORH Register Bit Descriptions

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.

13.8.42 Total Octets Transmitted

TOTL (040C8h; R/W / TOTH (040CCh; RC)

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 13-203. TOTL and TOTH Register Bit Descriptions

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.

13.8.43 Total Packets Received

TPR (040D0h; RC)

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 13-204. TPR Register Bit Description

Field	Bit(s)	Initial Value	Description
TPR	31:0	0b	Number of all packets received.

13.8.44 Total Packets Transmitted

TPT (040D4h; RC)

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, and packets generated by system manageability.

Table 13-205. TPT Register Bit Description

Field	Bit(s)	Initial Value	Description
TPT	31:0	0b	Number of all packets transmitted.

13.8.45 Packets Transmitted (64 Bytes) Count

PTC64 (040D8h; RC)

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 system manageability.

Table 13-206. PTC64 Register Bit Description

Field	Bit(s)	Initial Value	Description
PTC64	31:0	0b	Number of packets transmitted that are 64 bytes in length.

13.8.46 Packets Transmitted (65-127 Bytes) Count

PTC127 (040DCh; RC)

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, and packets generated by system manageability.

Table 13-207. PTC127 Register Bit Description

Field	Bit(s)	Initial Value	Description
PTC127	31:0	0b	Number of packets transmitted that are 65-127 bytes in length.

13.8.47 Packets Transmitted (128-255 Bytes) Count

PTC255 (040E0h; RC)

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 system manageability.

Table 13-208. PTC255 Register Bit Description

Field	Bit(s)	Initial Value	Description
PTC255	31:0	0b	Number of packets transmitted that are 128-255 bytes in length.

13.8.48 Packets Transmitted (256-511 Bytes) Count

PTC511 (040E4h; RC)

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, and packets generated by system manageability.

Table 13-209. PTC511 Register Bit Description

Field	Bit(s)	Initial Value	Description
PTC511	31:0	0b	Number of packets transmitted that are 256-511 bytes in length.

13.8.49 Packets Transmitted (512-1023 Bytes) Count

PTC1023 (040E8h; RC)

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 system manageability.

Table 13-210. PTC1023 Register Bit Description)

Field	Bit(s)	Initial Value	Description
PTC1023	31:0	0b	Number of packets transmitted that are 512-1023 bytes in length.

13.8.50 Packets Transmitted (1024 Bytes or Greater) Count

PTC1522 (040ECh; RC)

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, and packets generated by system manageability.

Table 13-211. PTC1522 Register Bit Description

Field	Bit(s)	Initial Value	Description
PTC1522	31:0	0b	Number of packets transmitted that are 1024 or more bytes in length.

13.8.51 Multicast Packets Transmitted Count

MPTC (040F0h; RC)

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 13-212. MPTC Register Bit Description

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0b	Number of multicast packets transmitted.

13.8.52 Broadcast Packets Transmitted Count

BPTC (040F4h; RC)

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 13-213. BPTC Register Bit Description

Field	Bit(s)	Initial Value	Description
BPTC	31:0	0b	Number of broadcast packets transmitted count.

13.8.53 TCP Segmentation Context Transmitted Count

TSCTC (040F8h; RC)

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.

Field	Bit(s)	Initial Value	Description
TSCTC	31:0	0b	Number of TCP Segmentation contexts transmitted count.

13.8.54 TCP Segmentation Context Tx Fail Count

TSCTFC (040FCh; RC)

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 could be due to PAYLEN errors. This register only increments if transmits are enabled.

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.

13.8.55 Interrupt Assertion Count[1]

IAC (04100h; RC)

This counter counts the total number of LAN interrupts generated in the system.

Field	Bit(s)	Initial Value	Description
IAC	31:0	0b	This is a count of all the LAN interrupt assertions that have occurred.

13.8.56 Interrupt Cause Rx Packet Timer Expire Count

ICRXPTC (04104h; RC)

Note: The statistics generated by this register might be skewed if the Interrupt Throttling Rate (ITR) register is not cleared.

Field	Bit(s)	Initial Value	Description
ICRXPTC	31:0	0b	This is a count of the receive packet timer expirations interrupt events that have generated interrupts.

13.8.57 Interrupt Cause Rx Absolute Timer Expire Count

ICRXATC (04108h; RC)

Note: The statistics generated by this register might be skewed if the Interrupt Throttling Rate (ITR) register is not cleared.

Field	Bit(s)	Initial Value	Description
ICRXATC	31:0	0b	This is a count of the receive absolute timer expirations interrupt events that have generated interrupts.

13.8.58 Interrupt Cause Tx Packet Timer Expire Count

ICTXPTC (0410Ch; RC)

Note: The statistics generated by this register might be skewed if the Interrupt Throttling Rate (ITR) register is not cleared.

1. Not applicable to the **631xESB/632xESB**.

Field	Bit(s)	Initial Value	Description
ICTXPTC	31:0	0b	This is a count of the transmit packet timer expirations interrupt events that have generated interrupts.

13.8.59 Interrupt Cause Tx Absolute Timer Expire Count ICTXATC (04110h; RC)

Note: The statistics generated by this register might be skewed if the Interrupt Throttling Rate (ITR) register is not cleared.

Field	Bit(s)	Initial Value	Description
ICTXATC	31:0	0b	This is a count of the transmit absolute timer expirations interrupt events that have generated interrupts.

13.8.60 Interrupt Cause Transmit Queue Empty Count ICTXQEC (04118h; RC)

Note: The statistics generated by this register might be skewed if the Interrupt Throttling Rate (ITR) register is not cleared.

Field	Bit(s)	Initial Value	Description
ICTXQEC	31:0	0b	This is a count of the transmit queue empty interrupt events that have generated interrupts.

13.8.61 Interrupt Cause Transmit Descriptor Low Threshold Count ICTXDLTC (0411Ch; RC)

Note: The statistics generated by this register might be skewed if the Interrupt Throttling Rate (ITR) register is not cleared.

Field	Bit(s)	Initial Value	Description
ICTXDLTC	31:0	0b	This is a count of the transmit descriptor low threshold interrupt events that have generated interrupts.

13.8.62 Interrupt Cause Receive Descriptor Minimum Threshold Count ICRXDMTC (04120h; RC)

Note: The statistics generated by this register might be skewed if the Interrupt Throttling Rate (ITR) register is not cleared.

Field	Bit(s)	Initial Value	Description
ICRXDMTC	31:0	0b	This is a count of the receive descriptor minimum threshold interrupt events that have generated interrupts.

13.8.63 Interrupt Cause Receive Overrun Count ICRXOC (04124h; RC)

Note: The statistics generated by this register might be skewed if the Interrupt Throttling Rate (ITR) register is not cleared.

Field	Bit(s)	Initial Value	Description
ICRXOC	31:0	0b	This is a count of the receive overrun interrupt events that have generated interrupts.

General Initialization and Reset Operation 14

14.1 Introduction

This section lists all necessary initializations and describes the reset commands for the PCIe* Family of Gigabit Ethernet Controllers.

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 device 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/NVM.

14.3 Initialization Sequence

The following sequence of commands is typically issued to the Ethernet controller by the software device driver in order to initialize the Ethernet controller to normal operation. The major initialization steps are:

- Disable Interrupts - see Interrupts during initialization.
- Issue Global Reset and perform General Configuration - see Global Reset and General Configuration.
- Setup the PHY and the link - see Link Setup Mechanisms and Control/Status Bit Summary.
- Initialize all statistical counters - see Initialization of Statistics.
- Initialize Receive - see Receive Initialization.
- Initialize Transmit - see Transmit Initialization.
- Enable Interrupts - see Interrupts During Initialization.

14.4 Interrupts During Initialization

Most drivers disable interrupts during initialization to prevent use of resources before they are allocated. Interrupts are disabled by writing to the IMC register. Note that the interrupts also need to be disabled after issuing a global reset, so a typical driver initialization flow might be:

- Disable interrupts
- Issue a Global Reset
- Disable interrupts (again)
- …

After the initialization completes, a typical driver enables the desired interrupts by writing to the IMS register.

14.5 Global Reset and General Configuration

Ethernet controller initialization typically starts with a global reset that puts it into a known state and enables the software device driver to continue the initialization sequence.

Several values in the Device Control Register (CTRL) need to be set at power up or after an Ethernet controller reset for normal operation.

- MAC mode (MAC TBI mode **82571EB/82572EI**) is determined by the LINK_MODE setting and can be read by software from the Device Status register. However, upon reading by software of this field, the software should set the MAC/PHY interface accordingly.
- FD should be set per interface negotiation (if done in software), or is set by the hardware if the interface is Auto-Negotiating. This is reflected in the Device Status Register in the Auto-Negotiating case.
- Speed is determined via Auto-Negotiation by the PHY, Auto-Negotiation by the MAC in TBI mode (SerDes mode the **631xESB/632xESB**), or forced by software if the link is forced. Status information for speed is also readable in STATUS.
- BEM is a system property and should be set accordingly.
- In TBI mode (**82571EB/82572EI** only), 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.
- CTRL.ILOS should be set to 0b (**82571EB/82572EI** only).

The default value of the Device Control Register is: 00100800h for the **631xESB/632xESB** (except for settings specifically called out in the EEPROM).

If using XOFF flow control, program the FCAH, FCAL, and FCT registers. If not, they should be written with 0b.

Table 14-1 lists several values that need to be set at power up, after an Ethernet controller reset, or after link up for normal operation.

Table 14-1. Power Up, Reset Bit, or Link Up Settings

631xESB/632xESB (MAC) Register	Bit/Offset/ Page	631xESB/632xESB Setting	82571EB/82572EI Setting
TARC0	26:23 21	N/A N/A	1111b[a,b] 0b[c]
TARC0	20 6:0	0b 1h	N/A
TARC1	28 26:24 22 6:0	N/A N/A N/A 1h	0b[d] 111b[a,b] 1b[e] N/A
TXDCTL	22	1b	1b
TXDCTL1	22	1b	1b
FFLT	20	0b	N/A
KUMCTRLSTA	Offset 0 Offset 2 Offset 10	808h 510h 0b (1000 Mb/s link) 100b (10/100 Mb/s link)	N/A
TCTL	11:4 21:12 24	0Fh 3Fh 1b	N/A
TCTL_EXT	19:10	40h	N/A
TIPG	29:20 9:0	7h 8h (1000 Mb/s link) 9h (10/100 Mb/s link)	N/A
FCTTV	15:0	FFFFh	N/A
82563EB/82564EB (PHY) Register	**Bit/Offset/ Page**	**631xESB/632xESB Setting**	**82571EB/82572EI Setting**
Power Management Control	Offset 14, Page C1	9h or 11h	N/A
	Offset 10, Page C1h	180h (1000 Mb/s link)[f] 980h (10/100 Mb/s link)	N/A

a. Set TARC0[26, 24, 23] and TARC1[26, 24] if enabling multiple transmit queues. Note that bits 26 and 24 must be set as a group (no other combination is allowed).
b. TARC0[25] and TARC1[25] must be set if enabling both TCP segmentation offload and multiple requests (MULR = 1b). Recommend to always set to 1b.
c. Must be set to 0b in 10/100 modes. Can be set to 1b only when device is configured in GbE speed for small packet performance increase.
d. TARC1[28] must be set to 1b when TCTL.MULR is disabled and set to 0b when TCTL.MULR is enabled.
e. Set to 1b when enabling TSO and MULR to workaround the issue described in Section 13.3.69.1.
f. Used to disable pass false carrier while in full-duplex mode.

Note: N/A = Not Applicable

14.6 Receive Initialization

Program the Receive address register(s) per the station address. This can come from the EEPROM/NVM or from any other means (for example, on some systems, this comes from the system PROM not the EEPROM/NVM on the adapter card)

Initialize the MTA (Multicast Table Array) to 0b. Entries can be added to this table as directed by software.

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.

Program RXDCTL with appropriate values. If initializing it at this stage, it is best to leave the receive logic disabled (EN = 0b) until after the receive descriptor ring has been initialized. If VLANs are not used, software should clear VFE. Then there is no need to initialize the VFTA. Select the receive descriptor type. Note that if using the header split RX descriptors, tail and head registers should be incremented by 2 per descriptor.

Initialize the Receive Control Register, by setting the *EN* bit to 1b.

14.6.1 Initialize the Receive Control Register

To properly receive packets requires simply that the receiver is enabled. This should be done only after all other setup is accomplished. If software uses the Receive Descriptor Minimum Threshold Interrupt, that value should be set.

The following should be done once per receive queue:

- Allocate a region of memory for the receive descriptor list.
- Receive buffers of appropriate size should be allocated and pointers to these buffers should be stored in the descriptor ring.
- Program the descriptor base address with the address of the region (RDBALn, RDBAHn, n = 0b or 1b).
- Set the length register to the size of the descriptor ring (RDLENn, n = 0b or 1b).
- If needed, program the head and tail registers (RDHn, RDTn, n = 0b or 1b). Note that the head and tail pointers are initialized (by hardware) to 0b after a power-on or a software-initiated device reset.
- The tail pointer should be set to point one descriptor beyond the end.

14.7 Transmit Initialization

Program the TXDCTL register with the desired TX descriptor write back policy. Suggested values are:

- GRAN = 1b (descriptors)
- WTHRESH = 1b
- All other fields 0b

Program the TXDCTL register. Suggested configuration:

- CT = 0Fh (16d collision)
- COLD = 03Fh (64d byte times)
- PSP = 1b
- EN=1b
- All other fields 0b

The following should be done once per transmit queue:

- Allocate a region of memory for the transmit descriptor list.
- Program the descriptor base address with the address of the region.
- Set the length register (TDLEN) to the size of the descriptor ring.
- If needed, program the head and tail registers (TDH, TDT). **Note:** the head and tail pointers are initialized (by hardware) to 0b after a power-on or a software-initiated device reset.

If working with multiple transmit queues, set up the queues priority by programming TARC0 and TARC1 registers.

Program the TIPG register (see the register description for the required value in Section 13.3.59).

Note: IPGR1 and IPGR2 are not needed in full-duplex, but it is easier to always program them to the values shown.

Initialize the transmit descriptor registers (TDBAL0/1, TDBAH0/1, TDL0/1, TDH0/1, and TDT0/1).

14.8 Link Setup Mechanisms and Control/Status Bit Summary

14.8.1 PHY Initialization

Refer to the PHY documentation for the initialization and link setup steps. The software device driver uses the MDIC register to initialize the PHY and setup the link.

14.8.2 MAC/PHY Link Setup

This section summarizes the various means of establishing proper MAC/PHY link setups, differences in MAC CTRL register settings for each mechanism, and the relevant MAC status bits. The methods are ordered in terms of preference (the first mechanism being the most preferred).

- MAC settings automatically based on duplex and speed resolved by PHY (CTRL.FRCDPLX = 0b, CTRL.FRCSPD = 0b

  CTRL.FD .....Don't care; duplex setting is established from PHY's internal indication to the MAC (FDX) after PHY has auto-negotiated 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 software after reading flow control resolution from PHY registers
  CTRL.TFCE .....Must be set by software after reading flow control resolution from PHY registers
  CTRL.SPEED.....Don't care; speed setting is established from PHY's internal indication to the MAC (SPD_IND) after PHY has auto-negotiated a successful link-up
  STATUS.FD.....Reflects the actual duplex setting (FDX) negotiated by the PHY and indicated to MAC
  STATUS.LU.....Reflects link indication (LINK) from PHY qualified with CTRL.SLU (set to 1b)
  STATUS.SPEED.....Reflects actual speed setting negotiated by the PHY and indicated to the MAC (SPD_IND)

- MAC duplex setting automatically based on resolution of PHY, software-forced MAC/PHY speed
  (CTRL.FRCDPLX = 0b, CTRL.FRCSPD = 1b

  CTRL.FD.....Don't care; duplex setting is established from PHY's internal indication to the MAC (FDX) after PHY has auto-negotiated 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 software after reading flow control resolution from PHY registers
  CTRL.TFCE .....Must be set by software after reading flow control resolution from PHY registers
  CTRL.SPEED.....Set by software to desired link speed (must match speed setting of PHY)
  STATUS.FD.....Reflects the actual duplex setting (FDX) negotiated by the PHY and indicated to MAC
  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 and speed settings forced by software based on resolution of PHY (CTRL.FRCDPLX = 1b, CTRL.FRCSPD = 1b

  CTRL.FD .....Set by software based on reading PHY status register after PHY has auto-negotiated 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 software after reading flow control resolution from PHY registers
  CTRL.TFCE .....Must be set by software after reading flow control resolution from PHY registers
  CTRL.SPEED.....Set by software based on reading PHY status register after PHY has auto-negotiated 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/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 software to desired flow-control operation (must match flow-control settings of PHY)
  CTRL.TFCE .....Must be set by software 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

14.8.3 MAC/SerDes (TBI-Mode) Link Setup

Link setup procedures using an external SerDes/TBI interface mode:

- Hardware Auto-Negotiation Enabled
  (TXCW.ANE = 1b)

  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 hardware priority resolution (read only)
  CTRL.TFCE .....Set by hardware priority resolution (read only)
  CTRL.SPEED .....Ignored; speed always 1000 Mb/s when using TBI mode communications
  STATUS.FD .....Reflects hardware-negotiated priority resolution
  STATUS.LU.....Reflects RXCW.ANC (Auto-Negotiation complete)
  STATUS.SPEED .....Reflects 1000 Mb/s speed, reporting fixed value of 10b
- Software-Executed Auto-Negotiation Enabled
  (TXCW.ANE = 0b)

  CTRL.FD .....Should be set by software to the duplex value established via software priority resolution
  CTRL.SLU.....Should be set by software to 1b when software Auto-Negotiation is completed
  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 .....Ignored; speed always 1000 Mb/s when using TBI mode communications
  STATUS.FD .....Reflects the value written by software to CTRL.FD
  STATUS.LU.....Reflects whether loss-of-signal (LOS) from SerDes is indicated, qualified with CTRL.SLU (set to 1b)
  STATUS.SPEED .....Reflects 1000 Mb/s speed, reporting fixed value of 10b

- Forced-Link (Auto-Negotiation Skipped)
  (TXCW.ANE = 0b, and no software auto-negotiation performed)

  CTRL.FD.....Duplex is set by software for the desired duplex mode of operation
  CTRL.SLU .....Must be set to 1 by software to enable communications to the SerDes
  CTRL.RFCE.....Set by software for the desired mode of operation
  CTRL.TFCE .....Set by software for the desired mode of operation
  CTRL.SPEED.....Ignored; speed always 1000 Mb/s in TBI mode
  STATUS.FD.....Reflects the value written by software to CTRL.FD
  STATUS.LU.....Reflects whether loss-of-signal (LOS) from SerDes is indicated, qualified with CTRL.SLU (set to 1b)
  STATUS.SPEED.....Reflects 1000 Mb/s speed, reporting fixed value of 10b

14.9 Reset Operation

The Ethernet controller's reset sources are as follows:

LAN_PWR_GOOD[1]:

The Ethernet controller has an internal mechanism for sensing the power pins. Once the power is up and stable the Ethernet controller creates an internal reset, this reset acts as a master reset of the entire chip. It is level sensitive, and while it is 0b holds all of the registers in reset. LAN_PWR_GOOD is interpreted to be an indication that the Ethernet controller's power supplies are all stable. LAN_PWR_GOOD changes state during system power-up.

Note: The **631xESB/632xESB** only uses this signal as a reset. It does not have the internal mechanism for sensing power pins.

GIO_PWR_GOOD:

Asserting GIO_POWER_GOOD indicates that both the power and the PCIe* clock sources are stable. This pin asserts an internal reset also after a D3cold exit. Most units of the Ethernet controllers are reset on the rising edge of GIO_POWER_GOOD. The only exception is the GIO unit, which is kept in reset while GIO_POWER_GOOD is de-asserted (level).

Note: MMS is not reset by GIO_PWR_GOOD.

PERST# (82573E/82573V/82573L):

De-asserting PERST# indicates that both the power and the PCIe* clock sources are stable. This pin also asserts an internal reset after a D3cold exit. Most units are reset on the rising edge of PERST#. The only exception is the GIO unit, which is kept in reset while PERST# is active.

1. Not applicable to the **631xESB/632xESB**.

Device Disable/Dr Disable (82573E/82573V/82573L):

The Ethernet controller enters a Device Disable mode when the DEVICE_OFF_N pin is asserted without shutdown. The Ethernet controller enters Dr Disable mode when some conditions are met in Dr state.

Inband PCIe* Reset:

The Ethernet controller generates an internal reset in response to a Physical layer message from the PCIe* or when the PCIe* link goes down (entry to Polling or Detect state). This reset is equivalent to PCI reset in previous (PCI) gigabit LAN controllers.

D3hot to D0 Transition:

This is also known as ACPI Reset. The Ethernet controller generates an internal reset on the transition from D3hot power state to D0 (caused after configuration writes from D3 to D0 power state). Note that this reset is per function and resets only the function that transitioned from D3hot to D0.

Software Reset:

Software can reset the Ethernet controller by writing the *Device Reset* bit of the Device Control Register (CTRL.RST). The Ethernet controller re-reads the per-function EEPROM/NVM fields after a software reset. Bits that are normally read from the EEPROM/NVM are reset to their default hardware values. Note that this reset is per function and resets only the function that received the software reset. PCI Configuration space (configuration and mapping) of the Ethernet controller is unaffected.

Force TCO:

This reset is generated when manageability logic is enabled. It is only generated if the Reset on Force TCO bit of the EEPROM/NVM's Management Control word is 1b. In pass through mode it is generated when receiving a ForceTCO SMB command with bit 0 set. In ASF mode it can be set under various conditions, including watchdog timer expiration.

Force TCO (631xESB/632xESB):

Firmware running on the MMS can reset any of the LAN functions. This reset can be generated when either one of the manageability modes is enabled. It only is generated if the *Reset on Force TCO* bit of the relevant function in EEPROM's (word 23h) is 1b. In pass through mode it is generated when receiving a ForceTCO SMB command with bit 0 set. The generation of this reset in BMC mode is vendor dependent. This reset affects the MAC of the targeted function.

The **631xESB/632xESB** re-reads the per-function EEPROM fields after a force TCO reset. The LAN driver or the firmware should initialize this function after the Force TCO reset.

Firmware Reset:

This reset is activated by writing a 1b to the FWR bit in the HOST Interface Control Register (HICR) in CSR address 8F00h.

TBI[1] Link Reset:

When the Link Reset bit of the Device Control Register (CTRL.LRST) is written as a logic 1b, the Ethernet controller is forced into a link reset state. In this state TBI auto-negotiation is disabled. The transmitter sends /C/ ordered sets when Link Reset is 1b. Auto-negotiation is initiated/restarted when LRST transitions to 0b. A link reset is only relevant in TBI mode.

EEPROM/NVM Reset:

Writing a 1b to the EEPROM/NVM Reset bit of the Extended Device Control Register (CTRL_EXT.EE_RST) causes the Ethernet controller to re-read the per-function configuration from the EEPROM/NVM, setting the appropriate bits in the registers loaded by the EEPROM/NVM.

PHY Reset (82573E/82573V/82573L):

Software can write a 1b to the PHY Reset bit of the Device Control Register (CTRL.PHY_RST) to reset the internal/external PHY. The MMS unit must configure the PHY following a PHY Reset.

Following is the procedure of a PHY reset by software:

1. Obtain the Software/Firmware semaphore (*SWSM.SWESMBI* - 05B50h; bit 1). Set it to 1b.
2. Drive PHY reset (*CTRL.PHY_RST* at offset 0000h [bit 31], write 1b, wait 100 µs, and then write 0b).
3. Delay 10 ms
4. Start configuring the PHY.

Note: Release the Software/Firmware semaphore after configuring the PHY.

PHY Reset (82571EB/82572EI):

To reset the PHY using software:

1. Obtain the Software/Software semaphore (*SWSM.SMBI* - 05B50h; bit 0). This is needed for multi-threaded environments.
2. Read (*MANC.BLK_Phy_Rst_On_IDE* – 05820h; bit 18) and then wait until it becomes 0b.
3. Obtain the Software/Firmware semaphore (*SWSM.SWESMBI* - 05B50h; bit 1).
4. Drive PHY reset (*CTRL.PHY_RST* at offset 0000h [bit 31], write 1b, wait 100 µs, and then write 0b).
5. Release the Software/Firmware semaphore (*SWSM.SWESMBI* - 05B50h; bit 1).
6. Wait for the *CFG_DONE* (*EEMNGCTL.CFG_DONE* at offset 1010h [bit 18] becomes 1b).

1. **82571EB/82572EI** only.

7. Wait for a 1 ms delay. The PHY should now be ready. If additional access to the PHY is necessary (reads or writes) the Software/Firmware semaphore (*SWSM.SWESMBI* - 05B50h; bit 1) must be re-acquired and then released once done.
8. Release the Software/Software semaphore (*SWSM.SMBI* - 05B50h; bit 0). This is needed for multi-threaded environments.

In-Band PHY Reset (631xESB/632xESB)

Software writes a 1b to the *PHY Reset* bit of the Device Control Register (CTRL.PHY_RST) to send an inband reset to the relevant port in the 82563EB/82564EB (PHY) through the GLCI interface. MMS firmware receives an interrupt and an indication of the reset through the *PHYR0A* and *PHYR1A* bit in ICR4. Firmware should reconfigure the PHY following a PHY Reset.

LPC Reset (631xESB/632xESB):

The behavior of the LPC interface and timings of LRESET# are different for general system reset than for entry or exit from low power state.

The host and peripherals must obey the following rules:

1. When LRESET# de-asserts, the clock is assumed to be running. The exact number of clocks are the same as stated in the PCI specification.
2. When LRESET# is asserted, the host behaves as follows:
   a. LFRAME# is driven high
   b. LAD[3:0] is tri-stated

When LRESET# is asserted, the **631xESB/632xESB** behaves as follows:

- LFRAME# is ignored
- LAD[3:0] is tri-stated

LRESET# assertion for general system reset can occur at any time and might be synchronous to LCLK.

LRESET# is the same as PCI 2.3 reset signal PCIRST#.

14.10 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 D0active power state (when internal registers become accessible, as enabled by setting the Memory Access Enable of the PCIe* Command register), and is guaranteed to be completed within 1 µs of this transition. Access to statistics registers prior to this interval may return indeterminate values.

All of the statistical counters are cleared on read and a typical software device driver reads them (thus making them zero) as a part of the initialization sequence.

Diagnostics and Testability 15

15.1 Diagnostics

To assist in test and debug of device-driver software, a set of software-usable features have been provided in the Ethernet controller. These features include controls for specific test-mode usage, as well as some registers for verifying device internal state against what the device-driver might be expecting.

The Ethernet controller provides software visibility (and controllability) into certain major internal data structures, including all of the transmit & receive FIFO space. However, interlocks are not provided for any operations, so diagnostic accesses may only be performed under very controlled circumstances.

The Ethernet controller also provides software-controllable support for certain loopback modes, to allow a device-driver to test transmit and receive flows to itself. Loopback modes can also be used to diagnose communication problems and attempt to isolate the location of a break in the communications path.

15.1.1 FIFO Pointer Accessibility

The Ethernet controller's internal pointers into its transmit and receive data FIFOs are visible through the head and tail diagnostic data FIFO registers listed in Section 13. Diagnostics software can read these FIFO pointers to confirm expected hardware state following a sequence of operation(s). Diagnostic software can further write to these pointers as a partial-step to verify expected FIFO contents following specific operation, or to subsequently write data directly to the data FIFOs.

15.1.2 FIFO Data Accessibility

The Ethernet controller's internal transmit and receive data FIFOs contents are directly readable and writeable. The specific locations read or written are determined by the values of the FIFO pointers, which may be read and written. When accessing the actual FIFO data structures, locations must be accessed as 32-bit words. See Section 13.

15.1.3 Loopback Operations

Loopback operations are supported by the Ethernet controller to assist with system and Ethernet controller debug. Loopback operations can be used to test transmit and receive aspects of software device drivers, as well as to verify electrical integrity of the connections between the Ethernet controller and the system (PCIe* bus connections, etc.).

Note: Configuration for loopback operations vary depending on the link configuration being used. Contact your Intel Field Service Representative for additional information on loopback modes for each Ethernet controller.

15.2 Testability[1]

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.

Note: Not applicable to the **82571EB/82572EI** controller.

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.

1. Not applicable to the **82573E/82573V/82573L**.